

Y. KERGALL

**Brève communication. Une nouvelle méthode  
de marquage dans la recherche d'une chaîne  
de longueur maximale d'un arbre**

*Revue française d'automatique, d'informatique et de recherche  
opérationnelle. Recherche opérationnelle*, tome 14, n° 2 (1980),  
p. 211-217.

[http://www.numdam.org/item?id=RO\\_1980\\_\\_14\\_2\\_211\\_0](http://www.numdam.org/item?id=RO_1980__14_2_211_0)

© AFCET, 1980, tous droits réservés.

L'accès aux archives de la revue « Revue française d'automatique, d'informatique et de recherche opérationnelle. Recherche opérationnelle » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/legal.php>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme  
Numérisation de documents anciens mathématiques  
<http://www.numdam.org/>

## *Brève communication*

### **UNE NOUVELLE MÉTHODE DE MARQUAGE DANS LA RECHERCHE D'UNE CHAÎNE DE LONGUEUR MAXIMALE D'UN ARBRE (\*)**

par Y. KERGALL (<sup>1</sup>)

---

*Résumé. — Nous présentons ici un nouvel algorithme de marquage pour chercher les chaînes de longueur maximale d'un arbre, qui nécessite seulement deux parcours de l'arbre ce qui le rend plus efficace que les algorithmes classiques.*

*Abstract. — A new marking method for finding the maximal length chains in a tree is given here; requiring only two scanning of the tree, it is more efficient than usual algorithms.*

#### **1. INTRODUCTION**

Il existe de nombreuses méthodes pour chercher les chaînes de longueur maximale d'un arbre, notées CLM, c'est-à-dire comportant le plus grand nombre d'arêtes : méthode matricielle [6], méthode d'effeuillage progressif [3], méthode de marquage [6], utilisée par Flament dans [2] améliorée dans [1]. Bien que les méthodes de marquage classiques soient moins rapides que la méthode d'effeuillage progressif, nous reprenons ici en les simplifiant l'étude des méthodes de marquage dont l'emploi se justifie lorsqu'on désire construire une arborescence associée à l'arbre, c'est-à-dire partitionner les sommets de l'arbre en niveaux par rapport à un sommet  $x$  de l'arbre choisi comme racine de l'arborescence.

Soit  $A$  un arbre ayant  $N$  sommets et  $F$  feuilles. Nous savons que la méthode de marquage revient à parcourir  $N$  fois l'arbre, que la méthode de marquage améliorée ne comporte, elle, que  $F$  parcours de l'arbre. Nous décrivons ici une méthode de marquage dite de « double marquage » qui se réduit à deux explorations de l'arbre initial et qui s'appuie sur un théorème établi dans cet article.

---

(\*) Reçu décembre 1978.

(<sup>1</sup>) Laboratoire d'Informatique appliquée, parc de Grandmont, Tours.

L'expérimentation semble montrer que cette méthode qui est la plus rapide des méthodes de marquage, est aussi plus efficace que l'effeuillage progressif.

## 2. THÉORÈME

**THÉORÈME :** *Soit  $x$  un sommet quelconque de l'arbre  $A$ . L'extrémité d'une plus longue chaîne issue de  $x$  est extrémité d'une CLM de  $A$ .*

Démontrons-le par l'absurde : soit  $f$  l'extrémité d'une plus longue chaîne issue de  $x$  et supposons que  $f$  ne soit pas extrémité d'une CLM. Deux cas sont à étudier; dans chacune des deux figures ci-dessous les lettres désignent les longueurs des chaînes correspondantes.

1° La plus longue chaîne issue de  $x$  coupe une CLM (en gros trait sur la figure) :

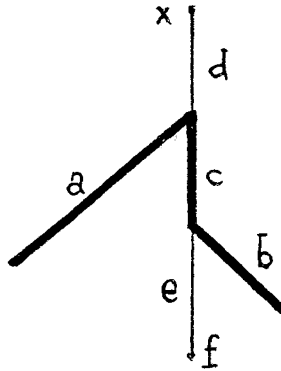
$f$  n'est pas extrémité d'une CLM, d'où

$$a + c + b > a + c + e \Rightarrow b > e.$$

$f$  est extrémité d'une chaîne de longueur maximale issue de  $x$ , d'où

$$d + c + e \geq d + c + b \Rightarrow e \geq b,$$

d'où contradiction.



Ceci est vrai quels que soient  $c$  et  $d$ , en particulier si  $d=0$  ( $x$  appartient à une CLM) ou si  $c=0$  (l'intersection d'une CLM et de la plus longue chaîne issue de  $x$  est réduite à un sommet).

2° La plus longue chaîne issue de  $x$  ne coupe pas une CLM :

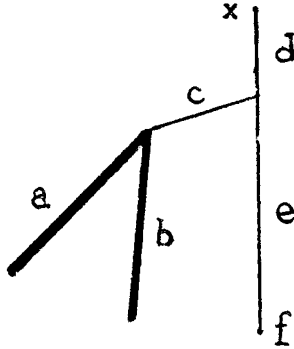
$f$  n'est pas extrémité d'une CLM, d'où :

$$a + b > a + c + e \Rightarrow b > e,$$

$f$  est extrémité d'une chaîne de longueur maximale issue de  $x$ , d'où :

$$d + e \geq d + c + b \Rightarrow e > b \quad \text{car } c > 0,$$

d'où contradiction, quelle que soit la valeur de  $d$ .



*Remarque :* On démontrerait comme précédemment que les plus longues chaînes issues d'un sommet  $x$  quelconque passent toutes par un centre de l'arbre.

### 3. ALGORITHME DE LA MÉTHODE DE DOUBLE MARQUAGE

Partant d'un sommet  $x$  quelconque de l'arbre  $A$ , on cherche l'extrémité d'une plus longue chaîne issue de  $x$ . Ceci se fait aisément par la méthode de marquage qui revient à partitionner les sommets de l'arbre en niveaux définis par

$$N_0 = \{x\}, \quad N_i = \{y \in A / d(x, y) = i\},$$

où  $d(x, y)$  est la longueur de la chaîne joignant  $x$  et  $y$ . Cette partition en niveaux  $N_i, 0 \leq i \leq k$ , se fait d'autant plus rapidement que  $k$  est petit, c'est-à-dire que le sommet  $x$  du départ est proche du centre de l'arbre.

Soit  $N_k$  le niveau d'indice maximal qui ne contient donc que des feuilles de  $A$  et soit  $f$  un élément de  $N_k$ . On réutilise la méthode de marquage, cette fois-ci en posant

$$N_0 = \{f\}, \quad N_i = \{y \in A / d(f, y) = i\}.$$

Soit  $N_l$  le niveau d'indice maximal. On a  $l \geq k$ , l'égalité étant vérifiée si le sommet  $x$  choisi au début est une extrémité d'une CLM. Puisque  $f$  est une extrémité d'une CLM, d'après le théorème précédent, alors la longueur d'une CLM est  $l$  et toute suite

$$(f, x_1, \dots, x_i, \dots, x_l) \quad \text{où } x_i \in N_i$$

et où

$$d(x_i, x_{i+1})=1 \quad \text{pour } 1 \leq i \leq l-1,$$

est une CLM.

Remarquons que dans cette deuxième répartition des sommets en niveaux, le nombre de niveaux est indépendant du sommet  $x$ , initialement choisi.

#### 4. REPRÉSENTATION EN MACHINE. EXPÉRIMENTATION

L'arbre est donné en associant à chaque sommet la liste de ses voisins. Ainsi la décomposition en niveaux à partir d'un sommet  $x$  quelconque choisi comme racine, se fait très simplement au moyen d'une file des descendants de  $x$  dans laquelle après  $x$  on place ses voisins puis les voisins de ses voisins, etc. Ainsi on parcourt une fois l'arbre à partir de  $x$ , en largeur d'abord et non en profondeur d'abord pour éviter les retours en arrière [4]. Cette décomposition est refaite à partir du dernier sommet  $y$  de la file en marquant cette fois-ci pour chaque sommet son ascendant dans l'arborescence de racine  $y$  de façon à avoir immédiatement le chemin de longueur maximale prêt à imprimer. Nous avons deux parcours de l'arbre; le temps d'exécution est donc en  $O(N)$ .

Rappelons que l'idée de l'algorithme d'effeuillage progressif est, en partant des feuilles de l'arbre (parcours de type « bottom up »), de construire toutes les demi-chaînes allant des feuilles au(x) centre(s) de l'arbre, demi-chaînes desquelles on exclut en cours de construction celles ne pouvant appartenir à une CLM. Le(s) centre(s) atteint(s), il ne reste plus qu'à raccorder directement à l'impression les demi-chaînes trouvées. Cet algorithme nécessite un seul parcours de l'arbre, mais le fait de rechercher à chaque étape de l'effeuillage si chacune des demi-chaînes en cours de construction peut ou non appartenir à une CLM, nuit à son rendement. Cet algorithme est néanmoins beaucoup plus rapide que les algorithmes classiques de marquage, et les algorithmes matriciels (étude comparée dans [3]).

Les deux graphiques ci-dessous permettent de comparer les temps (en 1/512 de seconde sur CII 10070) de la méthode d'effeuillage progressif (●) et du double marquage (□), d'abord dans le cas où on a 1 ou 2 CLM (fig. 1) puis dans le cas où on a 6 CLM (fig. 2). Pour mémoire, dans le cas  $N=60$  la méthode de marquage amélioré nécessite un temps  $T=706$  dans le cas de la figure 1 et  $T=740$  dans le cas de la figure 2.

Pour un  $N$  donné, le temps indiqué représente la moyenne des temps obtenus avec des arbres de  $N$  sommets pour lesquels nous avons fait varier le nombre

maximal de voisins, par exemple de 3 à 10 pour  $N = 60$ . Tous ces arbres sont tirés au hasard grâce au sous-programme ARBRE dont nous donnons la liste des instructions en annexe.

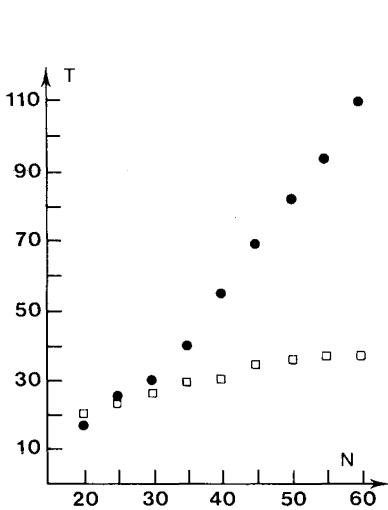


Fig. 1

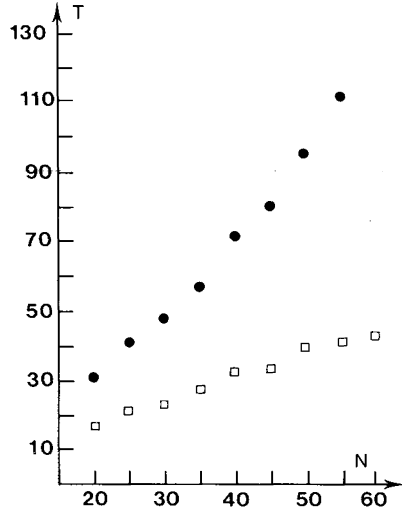


Fig. 2

*Remarques :* La méthode d'effeuillage progressif part des feuilles de l'arbre et va vers le centre  $C$  qui est le sommet qui minimise  $\text{Max}_{y \in A} d(C, y)$ , et à partir duquel on construit l'arborescence de niveaux *minimal*. La méthode du double marquage part d'un sommet quelconque, si possible le plus près du centre et construit une plus longue chaîne, issue de ce sommet, dont l'extrémité est la racine de l'arborescence de niveaux *maximaux*.

Ces méthodes pourront s'appliquer chaque fois qu'une décomposition en niveaux peut se révéler utile, par exemple dans certains problèmes de reconnaissance des formes [5].

#### BIBLIOGRAPHIE

1. J. P. ASSELIN DE BEAUVILLE, *Recherche de la chaîne de longueur maximale d'un arbre*, Informatique et Sciences humaines, n° 36, 1978, Paris.
2. C. FLAMENT, *Théorie des graphes et structures sociales*, Gauthier-Villars, Paris, 1968.
3. Y. KERGALL, *Chaînes de longueur maximale d'un arbre*, Informatique et Sciences humaines, fin 1979 (à paraître).
4. D. E. KNUTH, *Fundamental Algorithms*, 2<sup>e</sup> édition, 1973.
5. J. QUINQUETON, *Recherche d'alignements dans une image de points*, Thèse 3<sup>e</sup> cycle, Université Paris-VI, 1976.
6. B. ROY, *Algèbre moderne et théorie des graphes*, tomes 1 et 2, Dunod, Paris, 1970.

## ANNEXE

Liste des instructions du sous-programme MARK 2 réalisant la méthode du double marquage.

```

SUBROUTINE MARK2(VOISIN,SUC,NIVD,MARK,NSUM,MAXVDI)
C-----VOISIN EST LA MATRICE DANS LAQUELLE EST STOCKEE LA LISTE DES VOISINS-----
C-----CE CHAQUE SOMMET,MAXVDI EST LE DEGRE MAXIMUM DES SOMMETS DE L'ARBRE-----
C-----NSUM EST LE NOMBRE DE SOMMETS,NIVD CONTIENT LA FILE DES DESCENDANTS-----
C-----C LA SOMMET,SUC CONTIENT LE SUCCESSEUR DE CHAQUE SOMMET DANS L'-----
C-----ARBRESCENCE SUSPENDUE PAR UNE EXTREMITÉ D'UNE CLM,MARK EST UN -----
C-----TABLEAU DE MARQUAGE.-----
      INTEGER VOISIN(NSCM,MAXVDI),SUC(NSGM)
      DIMENSION NIVD(NSCM),MARK(NSCM)
      LEC=7
      IMP=8
      DO 7 I=1,NSCM
        NIVC(I)=C
        MARK(I)=C
7      LC 4 I=1,NSCM
        IF(VCISIN(I,2).NE.0) GOTO 5
4      CONTINUE
C-----REPARTITION EN NIVEAUX A PARTIR DU SOMMET I -----
5      MARK(I)=1
        NIVC(I)=1
        L=1
        K=1
11     NIVCK=NIVD(K)
        CC 1C J=1,MAXVDI
        IF(VCISIN(NIVCK,J).EQ.0) GOTO 15
        IF(MARK(VOISIN(NIVCK,J)).EQ.1) GOTO 10
        L=L+1
        NIVC(L)=VOISIN(NIVCK,J)
        MARK(NIVC(L))=1
10     CONTINUE
15     K=K+1
        IF(K-NSCM)11,20,20
20     CC 19 I=1,NSUM
19     MARK(I)=C
C LE SOMMET D'INDICE L EST UNE EXTREMITÉ D'UNE CHAÎNE DE LONGUEUR MAX -----
        MARK(NIVC(L))=1
        NIVD(1)=NIVD(L)
        L=1
        K=1
21     NIVCK=NIVD(K)
        CC 25 J=1,MAXVDI
        IF(VCISIN(NIVCK,J).EQ.0) GOTO 28
        IF(MARK(VOISIN(NIVCK,J)).EQ.1) GOTO 25
        L=L+1
        NIVC(L)=VOISIN(NIVCK,J)
        MARK(NIVC(L))=1
        SUC(NIVD(L))=NIVCK
25     CONTINUE
28     K=K+1
        IF(K-NSCM)21,30,30
30     NIVCL=NIVD(L)
32     WRITE(IMP,31)NIVCL
31     FORMAT(1X,I2)
        IF(NIVCL.EQ.NIVC(1)) RETURN
        NIVCL=SUC(NIVCL)
        GOTO 32
      END

```

## Liste des instructions du sous-programme ARBRE

```

SUBROUTINE ARBRE(NSDM,MAXVCI,VOISIN)
INTEGER VOISIN(NSDM,MAXVCI)
DIMENSION ITEMP(5)
C-----INITIALISATION,LECT.IMPR, DONNEES-----
CC 111 I=1,NSDM
CC 112 J=1,MAXVCI
VCISIN(I,J)=0
112 CONTINUE
111 CONTINUE
IF(NSDM.EQ.2)GOTO 101
WRITE(8,440)
440 FORMAT(1X,'DONNEE INCORRECTE')
RETURN
101 VOISIN(1,1)=2
VCISIN(2,1)=1
K=2
DEG=MAXVCI
C LES 3 INSTRUCTIONS SUIVANTES SERVENT A DETERMINER UN ENTIER IMPAIR-----
C QUELQUE CHOSE PARAMETRE D ENTREE A HASARD. -----
CALL TIME(ITEMP,12)
J=ITEMP(1)*ITEMP(2)
IX=2*J+1
C-----TIRAGE NB. VCISINS DE I-----
CC 102 I=1,NSDM
CALL HASARD(IX,LX,R,AN,0.,DEG)
C AN EST UN REEL ALEATOIRE COMPRIS ENTRE 0 ET DEG , BORNES EXCLUES-----
WRITE(8,441) AN
441 FORMAT(5X,F6.2)
IX=LX
NVCIS=AN
IF(NVCIS.NE.C)GOTO 222
IF(K.NE.1)GOTO 102
NVCIS=1
222 NVCIS=NVCIS+1
C-----CONSTRUCTION DE L ARBRE-----
CC 104 J=2,NVCIS
K=K+1
VCISIN(I,J)=K
VCISIN(K,1)=1
IF(K.EQ.NSDM) GOTO 1000
104 CONTINUE
102 CONTINUE
1000 CC 500 I=1,NSDM
500 WRITE(8,442)I,(VCISIN(I,J),J=1,MAXVCI)
442 FORMAT(1X,14,2X,10I3)
RETURN
END

```