HAL
open science

Algorithme de Calcul du degré de retournement d'un graphe planaire topologique

Jean-Pierre Nzali, Koumpo Tanékou Porgy, Hippolyte Tapamo

To cite this version:

Jean-Pierre Nzali, Koumpo Tanékou Porgy, Hippolyte Tapamo. Algorithme de Calcul du degré de retournement d'un graphe planaire topologique. Revue Africaine de Recherche en Informatique et Mathématiques Appliquées, 2002, Volume 1, 2002, pp.63-94. 10.46298/arima.1831 . hal-01261700

HAL Id: hal-01261700

https://inria.hal.science/hal-01261700v1

Submitted on 25 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algorithme de Calcul du degré de retournement d'un graphe planaire topologique

Jean-Pierre Nzali — Koumpo Tanékou Porguy — Hippolyte Tapamo

Département d'Informatique
Faculté des Sciences
B.P. 812 Yaoundé Cameroun
Email : jpnzali@uycdc.uninet.cm

RÉSUMÉ. Le degré de retournement est une caractéristique des graphes planaires topologiques. Dans cet article nous proposons un algorithme amélioré pour calculer le degré de retournement d'un graphe planaire topologique. Cet algorithme explore les différents cas possibles suivant une méthode descendante. Son implémentation sur machine a donné lieu à des tests sur des cas pratiques, ceci en des temps de calcul tout à fait raisonnables, sur des graphes dont l'un comporte plus d'une cinquantaine de sommets intérieurs de degré impair.

ABSTRACT. One characteristic of planar topological graphs is the reversal degree. In this paper, we propose an improve algorithm for calculating the reversal degree of a planar topological graphs. This algorithm explores various possible cases following the descending method. Practical tests carried out on machine, using graphs with more than fifty internal vertices of odd degree, have been realized within reasonable computing time.

MOTS-CLÉS: graphe planaire topologique, degré de retournement, algorithme, carte, SIG.

KEYWORDS: planar topological graph, reversal degree, algorithm, map, GIS.

1. Introduction

Dans les pays en développement, bien que l'apport des Systèmes d'Informations Géographiques (SIG) soit indéniable comme partout ailleurs [10],[11],[18],[17], surtout dans ses aspects d'aide à la décision et maîtrise de l'environnement, le manque de matériel adapté à la saisie des données spatiales constitue un véritable frein à l'introduction de cette nouvelle branche de l'informatique. Le problème du calcul du degré de retournement d'un graphe dont une solution est présentée ici s'est posé lors de l'acquisition, pour un logiciel de SIG, de cartes par une méthode n'utilisant pas les outils traditionnels comme le scanner ou la table à digitaliser. Cette méthode appelée Digitalisation au Millimètre (DIMI) a été expérimentée sur la carte du Cameroun ([3],[9]) et a donné des résultats satisfaisants, particulièrement dans le cadre des applications en cartographie thématique. La méthode et les résultats obtenus ont été présentés dans [13],[14]. Précisons que le degré de retournement est un minimum sur un ensemble de valeurs et le problème pratique ici est de trouver une orientation des arêtes du graphe planaire topologique qui réalise ce minimum. La connaissance de cette orientation pour une carte (qui est un graphe planaire topologique particulier) facilite son acquisition par la DIMI. La connaissance de cette orientation est aussi intéressante dans les SIG fonctionnant selon le modèle vecteur topologique [10] où les coordonnées des points constituant chaque arc sont rangées une seule fois dans la base, mais doivent être lues dans un sens ou dans un autre selon la face que délimite cet arc.

Nous avons étudié le problème de la recherche du degré de retournement d'un graphe en utilisant la théorie des graphes et nous sommes arrivés à des formules et une méthode algorithmique de calcul de cette caractéristique que nous présentons ci-dessous après quelques définitions. Cet algorithme est mis en œuvre et testé sur des exemples pratiques de difficulté graduée.

2. Définitions

Dans cette partie nous rappelons quelques définitions de la théorie des graphes [1], [8], [21] et nous précisons la notion de profondeur d'un sommet et de degré de retournement d'un graphe.

G est un **graphe planaire** s'il est possible de le représenter sur un plan de sorte que les sommets soient des points distincts, les arêtes des courbes simples et que deux arêtes ne se rencontrent pas en dehors de leurs extrémités. Sa représentation sur un plan est appelée **graphe planaire topologique**. Dans la suite de l'exposé nous nous intéresserons uniquement aux graphes planaires topologiques sans isthme. Un isthme

est une arête dont la suppression augmente le nombre de composantes connexes. Quand nous parlerons d'un graphe G il s'agira toujours d'un graphe planaire topologique G sans isthme.

Une face d'un graphe G est une région du plan limitée par des arêtes et telle que deux points arbitraires dans cette région peuvent toujours être reliés par un trait continu ne rencontrant ni sommets ni arêtes.

le **degré d'un sommet** est le nombre d'arêtes issues de ce sommet. Nous dirons qu'un sommet S de G appartient à une face f si et seulement si S est incident à deux arêtes appartenant à la frontière de f.

Nous dirons qu'un sommet de degré n de G est intérieur s'il appartient à n faces finies. Le terme **sidi** sera utilisé pour désigner un sommet intérieur de degré impair.

Un sommet intérieur sera dit de **profondeur 1** s'il est adjacent à au moins un sommet extérieur. Un sommet intérieur sera dit de **profondeur n** (n>1) s'il est adjacent à au moins un sommet de profondeur n-1 et n'est adjacent à aucun sommet de profondeur inférieur à n-1. Pour un sommet intérieur S de profondeur n il existe donc au moins une chaîne de n arêtes reliant S à un sommet extérieur. Par la suite nous appellerons **chaîne minimale** associée à S toute chaîne constituée de n arêtes et reliant S à un sommet extérieur.

Nous désignerons par **chaîne minimale** reliant deux sommets intérieurs toute chaîne reliant ces deux sommets et ayant le minimum d'arêtes.

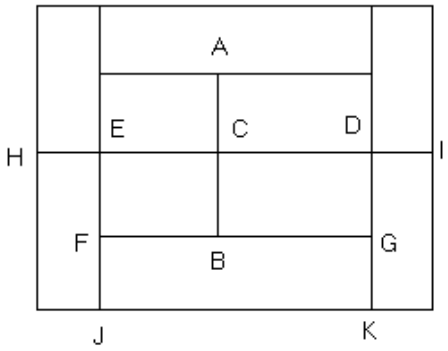

Figure 1. *Exemple de graphe planaire topologique sans isthme*

La figure 1 présente un exemple de graphe. Les sommets H, I, J et K par exemple, sont des sommets extérieurs. Les sommets A, B, C, D, E, F et G sont intérieurs. Les sommets D, E, F et G sont de profondeur 1. Les sommets A, B et C sont de profondeur

2. La chaîne constituée des arêtes AC et CB constitue la chaîne minimale reliant les sommets A et B. La chaîne constituée des arêtes BF et FJ est une chaîne minimale associée au sommet intérieur B. La chaîne constituée des arêtes BG et GK est une autre chaîne minimale associée au même sommet B.

Nous dirons que G est à **zéro retournement** si on peut orienter dans un seul sens les arêtes de G de telle sorte que chaque face finie soit délimitée par un circuit. Nous dirons que G est **à n retournements** (n>0) s'il faut orienter au moins n de ses arêtes dans les deux sens pour que chaque face finie de G soit délimitée par un circuit. Les graphes planaires topologiques à zéro et à un retournement ont été entièrement caractérisés dans [14] et [15]. Il en ressort qu'une condition nécessaire et suffisante pour que G soit à zéro retournement est que G ne possède aucun sommet intérieur de degré impair. De même une condition nécessaire et suffisante pour que G soit à un retournement est qu'il ait un seul sidi de profondeur 1 ou qu'il ait deux sidis adjacents.

Nous dirons que G est entièrement orienté si chacune de ses faces finies est délimitée par un circuit, certaines arêtes étant éventuellement orientées dans les deux sens.

Nous dirons qu'une chaîne est orientée dans les deux sens si chaque arête constituant cette chaîne est orientée dans les deux sens. Un sidi sera dit isolé si sa chaîne minimale est orientée dans les deux sens. Un sidi sera dit apparié à un autre si la chaîne minimale les reliant est orientée dans les deux sens.

3. Calcul du degré de retournement d'un graphe Planaire topologique

3.1. Théorème 1

Une condition suffisante pour que G soit entièrement orientable est que chaque sidi de G soit ou isolé ou apparié à un autre sidi de G.

Remarquons d'abord que pour orienter les arêtes des n faces auxquelles appartient un sommet intérieur de degré pair de telle sorte que chaque face soit délimitée par un circuit, il suffit d'imposer une orientation à l'une des arêtes issues de ce sommet, l'orientation des autres arêtes issues du même sommet s'en déduit automatiquement (figure 2 a). En ce qui concerne un sommet de degré impair, nous allons d'abord orienter une arête issue de ce sommet dans les deux sens avant d'imposer une orientation à l'une des arêtes restantes. L'orientation des autres arêtes s'en déduit aussi automatiquement (figure 2 b). Autrement dit, dès que les arêtes d'une face contenant un

sommet sont orientées de telle sorte que cette face soit délimitée par un circuit, l'orientation des autres faces s'en déduit automatiquement.

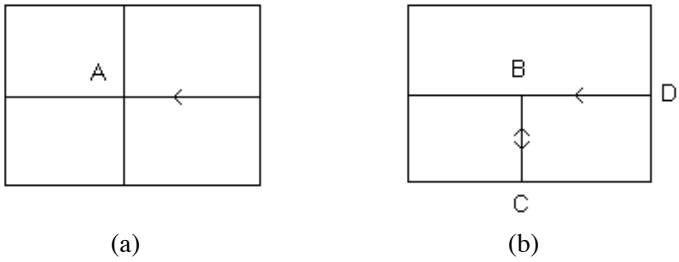

Figure 2. *Exemples d'orientation initiale des arêtes*

Pour démontrer le théorème nous supposons donc que G est un graphe planaire topologique possédant n sidis. Chaque sidi est soit isolé soit apparié à un autre. Précisons qu'un sidi isolé n'est pas apparié à un autre et inversement un sidi apparié à un autre n'est pas isolé. Au départ seules les arêtes appartenant aux différentes chaînes minimales sont orientées dans les deux sens. Les autres arêtes de G n'ont aucune orientation. Nous allons maintenant montrer comment il faut procéder pour orienter les différentes arêtes restantes de G pour que chaque face soit délimitée par un circuit.

Pour cela nous allons partir de l'un des sidis S de G et imposer une orientation à l'une des arêtes non déjà orientées de S. D'après la remarque faite plus haut, ce sens permet d'orienter les arêtes des différentes faces auxquelles appartient S de telle sorte que chacune de ces faces soit un circuit. Ce premier travail permet d'imposer une orientation à des arêtes appartenant à des faces ne contenant pas S. Nous utilisons ce sens pour orienter ces nouvelles faces et de proche en proche pour orienter entièrement G. En fait nous pouvons commencer l'orientation sur n'importe quelle arête non encore orientée puisqu'en définitive chaque arête aura un seul sens d'orientation.

Dans cette orientation nous n'avons pris que les chaînes minimales ce qui veut dire qu'il peut exister d'autres orientations de G avec plus d'arêtes orientées dans les deux sens. C'est pour cela que notre condition n'est pas nécessaire. Comme le degré de retournement que nous recherchons est un minimum, nous ne sommes pas intéressés par les cas d'orientations comportant un plus grand nombre d'arêtes orientées dans les deux sens.

Calculer le degré de retournement de G revient donc à éliminer l'effet de chaque sidi soit en l'isolant, soit en l'appariant à un autre sidi. Cette approche permet non seulement de trouver le degré de retournement mais aussi de trouver les différentes orientations des arêtes pour que chaque face soit un circuit.

3.2. Exemple : Graphe planaire topologique à 3 sidis

Considérons le schéma simplifié d'un graphe planaire topologique à trois sidis S_1, S_2 et S_3 (figure 3). Chaque sidi S_i a une profondeur P_i. Entre deux sidis S_i et S_j il y a une chaîne minimale composée de h_{ij} arêtes. Pour éliminer l'effet de ces trois sidis, nous avons quatre possibilités : isoler chacun des trois sidis (une possibilité), isoler un des trois sidis et apparier les deux autres (trois possibilités). Le degré de retournement de ce graphe est alors donné par la formule suivante :

$$D = \min(p_1 + p_2 + p_3, p_1 + h_{23}, p_2 + h_{13}, p_3 + h_{12})$$

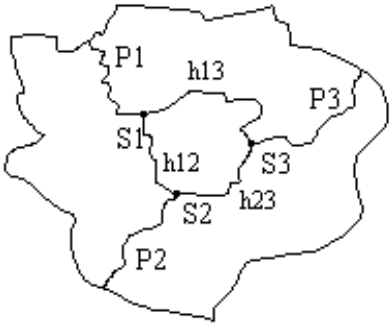

Figure 3. *Schéma simplifié d'un graphe planaire topologique à trois sidis*

La première expression (P_1+P_2+P_3) représente le poids du cas où les arêtes reliant chaque sidi au sommet extérieur le plus proche ont été orientées dans les deux sens. Les trois sidis sont isolés. La deuxième expression (P_1+h_{23}), quant à elle, représente le poids du cas où S_1 est isolé et les deux autres sidis sont appariés. Les deux dernières expressions se déduisent de la deuxième par permutation des rôles joués par les trois sidis.

Si nous avons par exemple P_1=1, P_2=1, P_3=3, h_{12}=1, h_{13}=3 et h_{23}=2, alors le degré de retournement de la carte sera de 3.

Cette analyse montre que le degré de retournement d'un graphe est inférieur ou égal à la somme des profondeurs de ses sidis. Il est aussi supérieur ou égal à la moitié du nombre de sidis. Ce minimum est atteint quand on a un nombre pair de sidis appariés deux à deux de chaîne minimale égale à un. Le degré de retournement D d'un graphe possédant n sidis vérifie donc la relation suivante :

$$\frac{n}{2} \leq D \leq \sum_{i=1}^{n} p_i$$

Le graphe planaire topologique de la figure 4 est à deux degrés de retournement avec trois sidis A, B et C. Ce degré de retournement est obtenu en isolant le sidi A (ce qui revient à orienter dans les deux sens l'arête AD) et en appariant les sidis B et C (ce qui revient à orienter dans les deux sens l'arête BC).

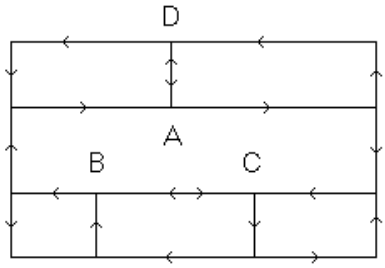

Figure 4. *Exemple de graphe à 3 sidis et 2 degrés de retournement*

3.3. Graphe planaire topologique à n sidis

Considérons un graphe planaire topologique G possédant n sidis. Nous utiliserons le terme «cas possible» pour désigner une façon d'éliminer les n sidis. A chaque cas possible correspond un poids qui est le nombre d'arêtes orientées dans les deux sens pour ce cas. Adoptons la notation suivante :

Ω_n désigne le nombre de cas possibles quand on dispose d'un graphe ayant n sidis.

Ω_n^i désigne le nombre de cas dans lesquels on a exactement i paires de sidis appariés.

Ω_i désigne le poids associé au cas i.

C_i désigne le nombre de paires de sidis appariés contenus dans le cas i.

D désigne le degré de retournement du graphe G.

Nous avons les formules suivantes :

$$\Omega_n = \sum_{i=0}^{ent(n/2)} \Omega_n^i \qquad (1)$$

ent(n/2) représentant la partie entière de la division de n par 2.

$$\Delta_i = \sum_{j=1}^{c_i} h_j + \sum_{j=1}^{n-2c_i} p_j \qquad (2)$$

$$D = \min(\Delta_i), i = 1,2,\ldots \Delta_n \qquad (3)$$

La relation (1) traduit simplement le fait que l'ensemble des cas possibles est la somme des cas possibles contenant zéro paire de sidis appariés, une paire de sidis appariés, …, ent(n/2) paires de sidis appariés.

La relation (2) donne le poids de chaque cas possible. Par h_j on désigne le nombre d'arêtes dans une chaîne minimale séparant les sidis de la paire j. p_j désigne quant à lui la profondeur du sidi j. La relation (3) en déduit le degré de retournement par calcul du minimum sur les différents poids. Plusieurs cas possibles peuvent avoir le poids minimum. Ce sont alors des solutions équivalentes.

3.4. Théorème 2

Pour un graphe qui a n sidis, le nombre de cas possibles dans lequel on a i paires de sidis appariés (1<i<=ent(n/2)) est donné par la formule suivante:

$$\Delta_n^i = \Delta_{n-1}^i + (n-1)\Delta_{n-2}^{i-1}$$

Les cas possibles de i paires contenant une paire de la forme (S_1,S_a) avec 1<a<=n sont au nombre de (n-1) Δ^{i-1}_{n-2}; les cas possibles de i paires ne contenant aucune paire de la forme (S_1,S_a) avec 1<a<=n sont au nombre de Δ^i_{n-1}, d'où le théorème [22].

Il existe un seul cas dans lequel tous les sidis sont isolés. Il n'y a pas de paire de sidi. D'où la relation :

$$\Delta_n^0 = 1 \quad \forall n > 0$$

Avec un seul sidi on ne peut constituer de paire, d'où la relation :

$$\Delta_1^i = 0 \quad \forall i > 0$$

Pour constituer i paires de sidis il faut que 2i soit au plus égal à n, d'où la relation:

$$\Delta_n^i = 0 \quad \textit{pour tout i tel que } 2i > n$$

Le nombre de cas possibles contenant une paire avec n sidis est donné par la formule classique des combinaisons suivante :

$$\Delta_n^1 = c_n^2 = \frac{n(n-1)}{2}$$

Le tableau 1 ci-après présente quelques valeurs de $\square_n^i$ et du nombre de cas possibles pour n variant de 1 à 14. La première colonne donne les différentes valeur de n. La première ligne donne le nombre de paires dans chaque cas. Les deux dernières colonnes donnent le nombre de cas possibles $\square_n$ comparé à 2^n. Pour n=5 par exemple on lit sur cette ligne du tableau qu'il y a un cas possible avec 0 paire de sidis, 10 cas avec une paire de sidis et 15 cas avec 2 paires de sidis.

N	Nombre de paires par cas								$\square_n$	2^n
	0	1	2	3	4	5	6	7		
1	1	0	0	0	0	0	0	0	1	2
2	1	1	0	0	0	0	0	0	2	4
3	1	3	0	0	0	0	0	0	4	8
4	1	6	3	0	0	0	0	0	10	16
5	1	10	15	0	0	0	0	0	26	32
6	1	15	45	15	0	0	0	0	76	64
7	1	21	105	105	0	0	0	0	232	128
8	1	28	210	420	105	0	0	0	764	256
9	1	36	378	1 260	945	0	0	0	2 620	512
10	1	45	630	3 150	4 725	945	0	0	9 496	1 024
11	1	55	990	6 930	17 325	10 395	0	0	35 696	2 048
12	1	66	1 485	13 860	51 975	62 370	10 395	0	140 152	4 096
13	1	78	2 145	25 740	135 135	270 270	135 135	0	568 504	8 192
14	1	91	3 003	45 045	315 315	945 945	945 945	135 135	2 255 345	16 384

Tableau 1. *Valeur de $\square_n$ pour n inférieur à 15.*

Comme le montre le tableau 1, la croissance est plus qu'exponentielle. Ce résultat ne peut être utilisé tel quel que pour des graphes avec peu de sidis (une dizaine environ). Nous sommes en présence d'un problème d'explosion combinatoire [12] ou d'optimisation combinatoire [21]. En effet si on désigne par C l'ensemble des cas possibles et par f⊡C → N l'application de C dans l'ensemble des entiers N qui à chaque cas possible associe son poids, il s'agit de trouver _ ⊡C tel que⊡

$$f(_) = \mathrm{Min}_{c \square C} [f(c)]$$

Pour des graphes possédant un nombre élevé de sidis il convient donc d'utiliser une approche algorithmique et des méthodes appropriées afin de pouvoir évaluer le degré de

retournement dans un temps acceptable sans forcément passer par tous les cas possibles. C'est ce que nous nous proposons d'examiner dans la suite de ce travail.

4. Algorithme de calcul du degré de retournement

Dans cette partie nous présentons une méthode algorithmique pour déterminer le degré de retournement d'un graphe dans le cas général. Cet algorithme est basé sur les méthodes par séparation et évaluation [21] et utilise un parcours en profondeur [4], [12]. Nous allons d'abord présenter en général la méthode utilisée pour effectuer la séparation et l'évaluation ensuite nous présenterons l'algorithme qui en découle.

4.1 Méthode de séparation et d'évaluation

Le degré de retournement est un minimum parmi un grand nombre de poids associés aux différents cas possibles. Nous connaissons un majorant et un minorant pour cette quantité. Un graphe G peut avoir plusieurs cas possibles associés au degré de retournement, c'est-à-dire associés au poids minimum. L'algorithme s'arête s'il trouve un cas possible réalisant ce minimum. G n'a qu'un seul degré de retournement qui est un entier. Le nombre de cas possibles est très élevé. C'est un problème d'optimisation combinatoire [21]. Les méthodes de séparation et évaluation vont nous permettre de trouver ce degré sans examiner systématiquement tous les cas possibles. Pour cela nous allons décomposer par étape l'ensemble des cas possibles comme indiqué dans la figure 5. A la première étape nous apparions le sidi 1 à l'un des n-1 sidis restants ou bien nous l'isolons. A la seconde étape nous apparions deux sidis parmi les n-2 restant et ainsi de suite jusqu'à ce qu'il ne soit plus possible de faire des appariements. Les branches de l'extrême droite correspondent aux sidis isolés.

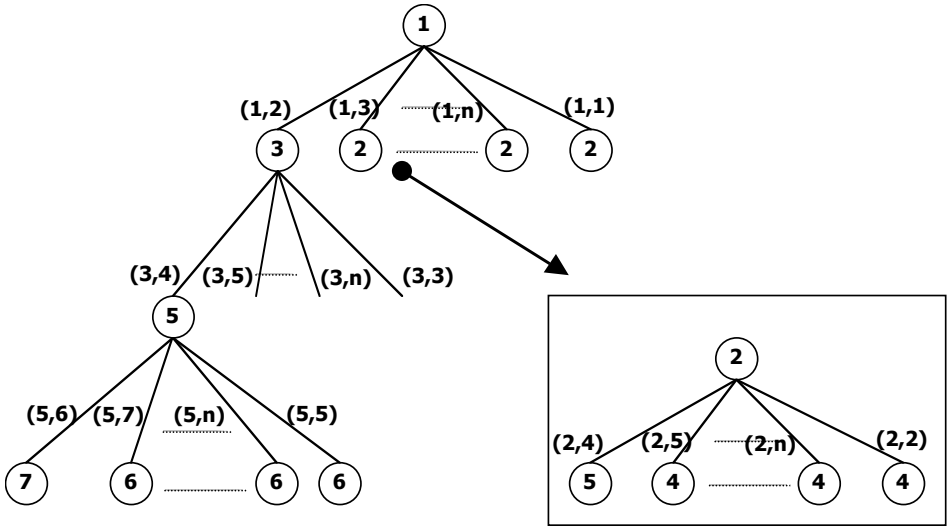

Figure 5. *Exemple de décomposition de l'ensemble des cas possibles*

Sur la figure 5 la notation (n,m), avec n différent de m, indique que le sidi n est apparié au sidi m et la notation (n,n) indique que le sidi n est isolé. Dans le cas où les sidis sont adjacents les uns aux autres il est évident que le degré de retournement sera donné par un cas possible contenant un grand nombre de sommets appariés et très peu de sommets isolés. Quand le sidi 1 est apparié au sidi 3 (deuxième branche issue de la racine à partir de la gauche), le sidi 2 à son tour peut être apparié soit à 4, soit à 5, …, soit à n. Il peut même être isolé. C'est ce que montre le sous-arbre développé au bout de la flèche issue du petit rond contenant 2.

La condition suivante (condition d'éligibilité) nous permet d'éliminer, sans aller jusqu'au bout, les cas possibles ne pouvant conduire à un minimum□

Si à une étape donnée, le sidi n ne peut pas être apparié au sidi m (chaîne minimale trop longue par rapport au maximum imposé par l'utilisateur par exemple), le parcours du sous-arbre issue de cette paire est abandonné.

La méthode de parcours en profondeur de l'arbre [4] est utilisée pour explorer les différents cas possibles. Elle permet de descendre plus profondément dans l'arbre chaque fois que c'est nécessaire et d'examiner systématiquement tous les cas possibles. Pour un cas possible donné, l'évaluation se fait en calculant son poids. Si ce poids est plus petit que le minimum trouvé jusque là, il devient le minimum, sinon on le rejette.

4.2 Présentation de l'algorithme

Cet algorithme reçoit en entrée l'ensemble des n sidis du graphe avec pour chacun sa profondeur et pour chaque paire de sidis la chaîne minimale les reliant. Les profondeurs sont présentées dans un vecteur de n éléments dit vecteur de profondeur. Les chaînes minimales sont présentées dans une matrice triangulaire à diagonale nulle dite matrice de chaîne minimale. Il met en sortie un cas possible de poids minimum qui est le degré de retournement cherché.

Bien que cet algorithme puisse générer tous les cas possibles (pour un petit nombre de sidis par exemple), son paramétrage peut permettre, comme nous le verrons plus loin, de trouver le degré de retournement en examinant très peu de cas possibles.

Cet algorithme a connu d'importantes améliorations par rapport à celui présenté au Colloque Africain sur la Recherche en Informatique (CARI' 2000) [16]. Ces améliorations nous ont permis de calculer en quelques minutes le degré de retournement de cartes qui ont un nombre élevé de sidis comme celles de l'Afrique et des Etats Unis avec leur découpage en états.

Nous présentons cet algorithme à travers cinq procédures :

- Une procédure principale
- Une procédure d'initialisation (Procédure INIT) qui lit les paramètres c'est-à-dire entre autres le nombre de sidis, le vecteur de profondeur et la matrice de chaîne minimale.
- Une procédure de calcul du poids (fonction à minimiser) de chaque cas possible (Procédure POIDS) qui retient aussi le poids minimum. Elle utilise pour ce calcul le vecteur de profondeur et la matrice de chaîne minimale.
- Une procédure (Procédure FIXER) qui fixe les m-1 paires pour une valeur de m donnée supérieure ou égale à deux. La paire m est ensuite calculée dans la procédure COMBINE en utilisant les combinaisons.
- Une procédure COMBINE qui calcule la paire m (une fois que les m-1 paires sont fixées) en utilisant les combinaisons.

Nous présentons ci-après ces différentes procédures. Pour faciliter cette présentation, nous supposons que toutes les variables utilisées sont des variables communes à l'ensemble des procédures.

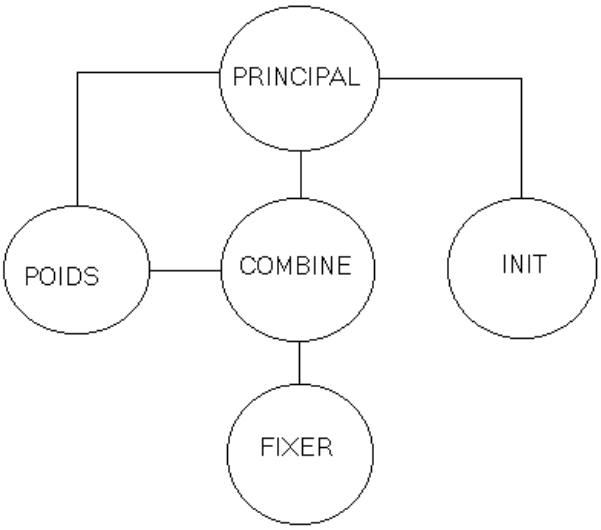

Figure 6. *Interconnexion entre les cinq procédures*

4.2.1. Procédure Principale

Début

Appeler la procédure INIT pour la lecture des paramètres;
in := 0;
Appeler la procédure POIDS pour l'initialisation;
in := 1;
Appeler la procédure COMBINE pour générer les différents cas possibles;
Imprimer le degré de retournement et le cas possible correspondant;

Fin.

La procédure principale appelle la procédure INIT pour la lecture des paramètres et la procédure POIDS pour l'initialisation de C0 et p0 (cas possible de départ et poids associé). Elle appelle ensuite la procédure COMBINE. A la fin de l'algorithme, la procédure principale imprime p0 et C0 qui représentent respectivement le poids minimum (degré de retournement du graphe proposé) et un cas possible ayant ce poids minimum. Le cas possible qui minimise la fonction poids permet d'orienter les arêtes de telle sorte que chaque face soit un circuit avec un nombre minimum d'arêtes orientées dans les deux sens.

4.2.2. Procédure INIT

Début
Lecture du nombre n de sidis;
Lecture du vecteur Vp de profondeur des n sidis;
Lecture de la matrice Mcm de chaîne minimale;
Lecture de Cmax, chaîne maximale autorisée entre 2 sidis à apparier;
Pour i=1 **jusqu'à** n;
C1(i) := i;
C2(i) := i;
Vi(i) := 0;
FinPour;
Fin;

Vp est un vecteur tel que Vp(i) désigne la profondeur du sidi i. Mcm est la matrice symétrique à diagonale nulle de chaîne minimale. Mcm(i,j) désigne donc la chaîne minimale (en terme de nombre d'arêtes) reliant les sidis i et j. Cmax est un nombre qui représente la chaîne maximale autorisée entre deux sidis à apparier. Si la chaîne minimale entre deux sidis est supérieure à Cmax, les deux sidis ne seront pas appariés par l'algorithme. Le sous-arbre correspondant n'est pas exploré. Cette approche permet d'éliminer d'office, comme nous le verrons plus loin, un grand nombre de cas qui dans certaines situations ne peuvent conduire à un minimum. Pour une carte de géographie par exemple, les sommets intérieurs sont souvent des sidis de degré trois. On peut, dans ce cas, donner à Cmax la valeur 1 pour que seuls les sidis adjacents soient effectivement appariés. La détermination des éléments de la matrice Mcm s'en trouve simplifiée. Mcm(i,j) vaut 1 si les deux sidis i et j sont adjacents et par exemple 2 (ou toute autre valeur supérieure à 1) dans les autres cas.

Les variables C1 et C2 utilisées ici sont des vecteurs contenant au départ les entiers de 1 à n. Pendant le déroulement de l'algorithme, C2 restera inchangé. Il sert uniquement à reconstituer C1 quand on passe d'un cas possible contenant m paires de sidis appariés à un autre contenant m+1 paires de sidis appariés. La variable C1, par contre contiendra le dernier cas possible généré. Le vecteur Vi quant à lui, indique les sidis isolés au milieu de C1, ceci par opposition, comme nous le verrons plus loin, aux sidis qui peuvent être isolés soit au début soit à la fin de C1. Supposons que C1 et Vi contiennent par exemple les valeurs suivantes pour n=10 sidis et m=3 paires de sidis appariés⊡

C1⊡	-1	2	3	4	-5	6	-7	8	9	10
Vi ⊡	0	0	1	1	0	0	0	0	0	0

Cette configuration traduit le fait que les sidis 3 et 4 sont isolés au milieu de C1. Les sidis 1 et 2 sont appariés. Il en est de même des sidis 5 et 6 ainsi que des sidis 7 et 8. Les sidis 9 et 10 sont isolés à la fin de C1 car le nombre de paires (3 ici) est déjà atteint. Pour un souci de clarté dans l'exposé, nous avons fait précéder le premier sidi d'une paire par le signe moins. Ainsi –1 signifie que le sidi 1 est apparié au sidi 2, de même, -5 signifie que le sidi 5 est apparié au sidi 6. il n'est donc pas nécessaire de faire placer ce signe par l'algorithme. Cette configuration est représentée dans l'arbre de la figure 7.

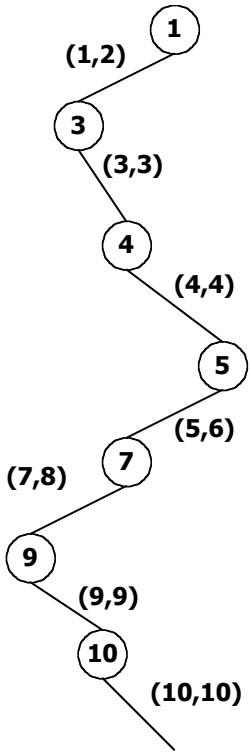

Figure 7. *Représentation d'un cas possible dans l'arbre.*

4.2.3.Procédure POIDS

La procédure POIDS calcule le poids d'un cas possible. Ce poids est la somme des profondeurs des sidis isolés de ce cas possible ajoutée à la somme des chaînes

minimales des paires composant ce même cas possible. Pour un cas possible contenu dans le vecteur C1, elle calcule d'abord la somme des profondeurs d'éventuels sidis isolés au début de C1, sidis de numéro inférieur à la valeur de la variable dpp (début première paire en partant de la gauche). Elle ajoute à cette somme les chaînes minimales des différentes paires ainsi que les profondeurs des sidis isolés tant au milieu de C1 qu'à la fin.

```
Début
    Si in=0  Alors   /* Initialisation du poids et des éléments pour l'impression du
                     résultat */
                     p0 := sommes des profondeurs des n sidis;
                     C0 := C2;
                     Vi0 := Vi;   /* Pas d'élément isolé au milieu du vecteur C0 */
                     dpp0 := 0;   /* Pas d'élément isolé au début de C0 */
                     m0 := 0;     /* Aucune paire de sidis */
             Sinon
                     /* Calcul du poids du cas possible qui est dans C1 */
                     d :=1;
                     p :=0;
                      /* Calcul du poids des sidis isolés au début de C1 */
             Tant que d<dpp Faire
                     p := p+Vp(d);
                     d := d+1;
             Fin Tant que
                     np := 1;  /* compteur du nombre de paires */
            /* Ajout des poids des paires et des sidis isolés au milieu de C1 */
             Tant que  np<=m Faire
                     p1 := C1(d);
                     Si Vi(d)=1 Alors p := p + Vp(p1);d := d + 1;
                     Sinon p2 := C1(d+1); p :=p + Mcm(p1,p2);
                                   d := d + 2; np := np+1;
                     FinSi
             Fin Tant que
             /*  Ajout des poids des sidis  isolés à la fin */
             Tant que d<=n Faire
                     p1 := C1(d);
                     p := p + Vp(p1);
                     d := d + 1;
             Fin Tant que
              /* Si le nouveau poids est plus petit il remplace l'ancien */
             Si p<p0 Alors
```

```
                    p0 := p;
                    C0 := C1;
                    Vi0 := Vi;
                    m0 := m;
                    dpp0 := dpp;
            FinSi;
        FinSi;
    Fin;
```

Les variables p0, C0, Vi0, dpp0 et m0 servent de variables de sauvegarde respectivement pour les variables p (poids d'un cas possible), C1, Vi, dpp et m. Elles permettent à la fin de l'algorithme d'imprimer le degré de retournement et le cas possible associé.

La variable C0 qui est un vecteur comme C1 et C2, contient le cas possible de poids minimum. La variable p0 contient le poids minimum associé à ce cas possible. La variable C0 est initialisée avec C2 c'est-à-dire le cas possible où tous les sidis sont isolés. Le poids p0 correspondant est tout simplement la somme des profondeurs de l'ensemble des n sidis. Cette initialisation est faite lors du premier appel avec in=0. Quand la variable in est non nulle, un nouveau cas possible vient remplacer C0 si son poids est inférieur à l'ancien p0.

4.2.4. Procédure COMBINE

```
Début
        /* Génération des cas possibles ne contenant qu'une paire */
        Pour k=1 jusqu'à  n-1  Faire
                Pour r=k+1 jusqu'à  n  Faire
                        Si Mcm(k,r)<=Cmax Alors
                        Appeler POIDS avec la paire (S_k, S_r),
                        les autres sidis étant isolés;
                        FinSi;
                FinPour;
        FinPour;
        /* Génération des cas possibles contenant plus d'une paire */
        nei := 0                        /* Nombre d'éléments isolés au milieu */
        dpp := 1;                       /* Début première paire  */
        fini := faux;
        pr_appel := vrai;
        m :=  2;                        /* Nombre de paire par cas */
        Tant que (m<= Ent(n/2) et fini=faux) Faire
```

```
                Appeler la procédure FIXER;
                Si fini=faux Alors
                        /* Génération de la dernière paire par combinaison */
                        Pour k=2(m-1)+dpp+nei  jusqu'à  n-1 Faire
                                Pour r=k+1 jusqu'à  n  Faire
                                        p1:=C1(k); p2:=C1(r);
                                        Si Mcm(p1,p2) <= Cmax Alors
                                        Appeler la procédure POIDS;
                                        FinSi;
                                FinPour;
                        FinPour;
                FinSi;
        FinTant que;
    Fin.
```

La variable **pr_appel** a la valeur **vrai** lors du premier appel de la procédure FIXER et a la valeur **faux** lors des autres appels. La variable **m** donne le nombre de paires par cas possible. La variable **dpp** qui veut dire début de la première paire, donne l'indice du début de la première paire en partant de la gauche. La variable **nei** donne le nombre total de sidis isolés au milieu du vecteur C1.

Pour générer les cas possibles contenant m paires de sidis (m>1), la procédure COMBINE fixe à l'aide de la procédure FIXER les m-1 premières paires et génère la dernière paire en utilisant les combinaisons. Comme dans le cas d'une paire, elle appelle chaque fois la procédure POIDS pour mettre à jour C0 et p0.

4.2.5. Procédure FIXER

```
Début
        cas_bon := faux;
        Tant que (ca_bon=faux et fini=faux) Faire
                Si pr_appel Alors
                        C1 := C2;
                        pr_appel :=faux ;
                Sinon
                        j := 0;
                        i : = m - 1;
                        Tant que (i > 0 et j = 0) Faire
                                r := pgep(i) +1;
                                reste := n-dpp-nei;
                                /* nombre d'éléments restant pour l'appariement */
                                Si r>n Alors Si reste >= 2*m Alors
```

```
                    /* on isole au milieu de C1 */
                    nei:=nei+1;
                    Isoler le 1° élément de la paire i et trier les
                    autres à droite;
                    j:=1;
                    Sinon   Libérer l'éventuelle plage
                            d'isolement avant la paire i;
                            i=i-1;
                    FinSi;
            Sinon Si r se trouve à droite Alors
                    Permuter r et pgep(i);
                    Trier à droite de la paire i;
                    j := 1;
                    Sinon r := r + 1;
                    FinSi;
            FinSi;
        Fin Tant que;
        Si i = 0 Alors
            /* On décale d'un élément vers la droite si possible */
            C1 : = C2;
            dpp := dpp + 1;
            Si n-dpp+1<2m Alors
                    /* On augmente le nombre de paire par cas */
                    m := m + 1;
                    dpp := 1;
                    Si  2m>n  Alors fini := vrai;
                    FinSi;
            FinSi;
        FinSi;
    FinSi;
    /* On vérifie ici si les m-1 paires générées sont toutes éligibles */
    bonne_paire :=vrai; cp :=0;
    Tant que (cp<=m-1 et bonne_paire=vrai et fini=faux) Faire
        cp :=cp+1;
        p1 := premier élément de la paire cp;
        p2 := deuxième élément de la paire cp;
        Si Mcm(p1,p2) > Cmax Alors bonne_paire=faux FinSI;
    Fin tant que;
    i :=cp;
    cas_bon=bonne_paire;
  Fin Tant que;
Fin.
```

Cette procédure fixe les m-1 premières paires de sidis (m>1). Elle vérifie ensuite que chaque paire est telle que la chaîne minimale séparant ses deux sidis est inférieure à Cmax avant de renvoyer ces m-1 paires à la procédure COMBINE. Dans le cas contraire, elle continue la recherche. Nous allons d'abord décrire cette procédure sans tenir compte de cet aspect que nous appelons contrôle de validité. Nous y reviendrons un peu plus loin.

Pour illustrer la démarche décrite par cette procédure, nous allons prendre le cas de n=10 sidis et m=3 paires. La première fois que cette procédure est appelée avec m=3 elle renvoie dans les vecteurs C1 et Vi les éléments suivants:

C1	-1	2	-3	4	5	6	7	8	9	10
Vi	0	0	0	0	0	0	0	0	0	0

Ceci traduit le fait que C1 contient deux paires de sidis (S_1,S_2) et (S_3,S_4). Aucun sidi n'étant isolé au milieu, Vi ne contient que des zéros. La procédure COMBINE va, au retour, générer la dernière paire par combinaison des sidis 5 à 10. Quand elle aura généré tous les cas possibles contenant les paires (S_1,S_2) et (S_3,S_4), elle va à nouveau appeler la procédure FIXER. Cette dernière va renvoyer dans C1 et Vi les éléments suivants:

C1	-1	2	-3	5	4	6	7	8	9	10
Vi	0	0	0	0	0	0	0	0	0	0

Dans cette nouvelle configuration de C1, 5 est venu remplacer 4 dans la deuxième paire et 4 est allé à l'ancienne place de 5. Dans l'algorithme **pgep(i)** désigne le plus grand élément de la paire i. Dans ce cas il s'agit du plus grand élément de la paire 2 qui est 4. En ajoutant 1 à cet élément nous obtenons 5 qui se trouve effectivement à droite de la paire i dans C1. On permute pgep(i) et r, c'est-à-dire 4 et 5. Ici le tri ne change pas grand chose. En positionnant la variable j à 1, cela nous permet de sortir de la boucle **Tant que** et de revenir à la procédure COMBINE pour la génération de la dernière paire.

Au huitième appel, en ajoutant 1 au plus grand élément de la deuxième paire, c'est-à-dire à 10 on trouve 11 qui est plus grand que n. On isole 3 et les deux premières paires seront constituées de (S_1,S_2) et de (S_4,S_5). L'isolement du sidi 3 se traduit par un «□□ à la position correspondante du vecteur d'isolement Vi comme indiqué ci-après.

C1	-1	2	3	-4	5	6	7	8	9	10
Vi	0	0	1	0	0	0	0	0	0	0

On recommence le processus d'isolement quand la deuxième paire sera (S_4,S_{10}). Le sidi 4 sera à son tour isolé, puis le sidi 5 et le sidi 6. La variable **nei** donne le nombre total d'éléments ainsi isolés. La configuration suivante donne la situation des vecteurs C1 et Vi après l'isolement du sidi 6.

C1	-1	2	3	5	4	6	-7	10	8	9
Vi	0	0	1	1	1	1	0	0	0	0

La première paire est toujours formée des sidis 1 et 2. la deuxième paire est formée des sidi 7 et 10 et la dernière paire est formée des sidis 8 et 9. Le deuxième élément de la deuxième paire est au maximum. On devrait isoler le sidi 7, mais si on l'isole le nombre de sidis non isolés est de 5 (les sidis 1, 2, 8, 9 et 10), ce qui ne permet pas de former trois paires. On ne peut donc continuer à isoler.

On procède à la libération des éléments isolés pour permettre leur appariement avec les autres. On diminue alors i de 1 ce qui permet de passer à la paire immédiatement à gauche (la première paire dans notre cas). On applique l'algorithme sur la première paire et la procédure FIXER renvoie dans C1 les éléments suivants :

-1	3	-2	4	5	6	7	8	9	10

Les appels suivants vont permettre de générer les cas possibles contenant les paires (S_1,S_3) et (S_2,S_i) pour i >3. On passera ensuite aux cas contenant les paires (S_1,S_4) et (S_2,S_i) avec i>2 et i≠4 et ainsi de suite.

Au bout d'un certain nombre d'appels, la procédure FIXER va recevoir en entrée la configuration suivante :

C1	-1	10	2	3	4	5	-6	9	7	8
Vi	0	0	1	1	1	1	0	0	0	0

Dans cette configuration, nous avons les paires (S_1,S_{10}) et (S_6,S_9), les sidis 2 à 5 étant isolés au milieu de C1. En ajoutant 1 au plus grand élément de la deuxième paire on trouve 10 qui n'est pas à droite. On ajoute à nouveau 1 et le résultat est supérieur à n. On libère les sidis isolés, on passe à la paire immédiatement à gauche (la première paire). Ici, il faut à nouveau passer à la paire immédiatement à gauche. Il n'y a pas de paire à gauche. La variable i a pris la valeur zéro. On sort de la boucle **Tant que**. Dans la partie qui suit de la procédure FIXER, quand cette situation arrive, on décale d'un élément vers la droite. La variable **dpp** qui donne l'indice de la première paire en partant de la

gauche est incrémentée de 1. Ce qui permet d'isoler un sidi à gauche (le sidi 1 dans ce cas). Cette situation marque dans notre exemple la fin des cas possibles contenant une paire de la forme (S_1,S_i) avec $i \neq 1$. La procédure FIXER renvoie dans C1 les éléments suivants :

C1	1	-2	3	-4	5	6	7	8	9	10
Vi	0	0	0	0	0	0	0	0	0	0

La variable dpp a pris la valeur 2 pour dire que la première paire de sidis commence au deuxième élément du vecteur C1. Comme nous l'avons mentionné plus haut, les sidis à gauche de cette première paire (comme le sidi 1 dans cet exemple) sont des sidis isolés au début de C1. Les éléments du vecteur d'isolement Vi ne sont pas positionnés à 1 pour ces sidis isolés au début du vecteur C1. Ils ne sont pas non plus positionnés pour les sidis isolés à la fin de ce même vecteur. Plus loin, la procédure FIXER recevra en entrée après plusieurs décalages à droite les éléments suivants avec dpp = 5 :

C1	1	2	3	4	-5	10	-6	9	7	8
Vi	0	0	0	0	0	0	0	0	0	0

La première paire de sidi est donc constituée des sidis 5 et 10 et la deuxième des sidi 6 et 9. Les sidis 1 à 4 sont isolés au début de C1. On va sortir de la boucle **Tant que** avec i nulle. En voulant décaler à nouveau vers la droite on va s'apercevoir que les éléments qui resteront à droite ne permettront pas de constituer les trois paires qu'il faut. C'est la fin des cas possibles contenant trois paires de sidis. En augmentant m de 1 on passe à la génération des cas possibles contenant 4 paires de sidis. En augmentant m on repart sans sommet isolé à gauche. Si en augmentant m il n'est plus possible de générer m paires avec les n sidis, la variable **fini** prend la valeur **vrai**, c'est la fin de l'algorithme. Le retour à la procédure principale provoque l'impression du cas possible de poids minimum.

4.2.6. Remarque

A un moment donné, le vecteur d'isolement peut avoir plusieurs plages de sidis isolés au milieu. Si nous prenons par exemple n=16 et m=4, on aura à un moment donné la configuration suivante avec dpp =2 :

C1	1	-2	3	4	5	-6	7	8	9	10	-11	12	13	14	15	16
Vi	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0

Le sidi 1 est isolé puisque la première paire commence à 2. Cette première paire est constituée des sidis 2 et 3. les sidis 4 et 5 sont isolés au milieu de C1 et constituent la première plage d'isolement. Les sidis 6 et 7 sont appariés. Les sidis 8, 9 et 10 sont aussi isolés au milieu de C1 et constituent la deuxième plage d'isolement. Les sidis 11 et 12 sont appariés. La quatrième paire est constituée des sidis 13 et 14. Les sidis 15 et 16 sont isolés à la fin de C1. Plus le nombre de sidis est élevé, plus il peut y avoir de plages d'isolement distinctes.

4.2.7. Contrôle de validité des cas générés

La description, faite ci-dessus de la procédure FIXER s'est faite en supposant que les m-1 paires chaque fois générées par cette procédure remplissaient la condition d'éligibilité. C'est-à-dire que pour chaque paire, la chaîne minimale séparant les deux sidis qui la composent était inférieure ou égale à Cmax. Dans la pratique, et tel que le montre la dernière partie de cette procédure, les m-1 premières paires ne sont renvoyées à la procédure COMBINE que si cette condition est remplie. La procédure COMBINE génère alors la dernière paire et vérifie que la même condition est remplie pour cette paire.

Si par exemple Mcm(1,2) est supérieur à Cmax, la procédure FIXER ne renverra à la procédure FIXER aucun cas possible contenant la paire (S_1,S_2).

Elle va directement passer aux cas possibles contenant la paire (S_1,S_3). Evaluons pour n=10 par exemple le nombre de cas possibles ainsi éliminés. Nous avons dit plus haut au niveau du théorème 1 que les cas possibles de i paires contenant une paire de type (S_1,S_a) avec 1<a<=n sont au nombre de (n-1) $\square^{i-1}_{n-2}$. Notons $A_{1,n}$ le nombre de cas possibles contenant une paire de type (S_1,S_a). Nous aurons:

$$A_{11,n} = \sum_{i=1}^{ent(n/2)} (n-1)\square^{i-1}_{n-2}$$

$$A_{1,10} = 9\left(\square^{0}_{8} + \square^{1}_{8} + \square^{2}_{8} + \square^{3}_{8} + \square^{4}_{8}\right) = 9 \text{ x } 764 = 6876$$

Les différentes valeurs sont données par le tableau 1 présenté plus haut. Pour n=10 un tel passage élimine donc l'examen de 6 876 cas possibles. Ce qui représente plus de 72 % de l'ensemble des cas. Ce pourcentage est assez élevé pour qu'il soit intéressant de recenser et de noter dans la matrice de chaîne minimale les sidis qu'il ne faut pas apparier. Le tableau 2 donne en dernière ligne les différents pourcentages pour n variant de 2 à 14. La deuxième ligne de ce tableau donne le nombre de cas éliminés, la troisième ligne donne le nombre de cas au total ($\square_n$) comme précisé dans le tableau 1. Ce

tableau montre que le pourcentage croît en fonction de n et se trouve au dessus de cinquante pour les valeurs examinées.

	$A_{1,2}$	$A_{1,3}$	$A_{1,4}$	$A_{1,5}$	$A_{1,6}$	$A_{1,7}$	$A_{1,8}$	$A_{1,9}$	$A_{1,10}$	$A_{1,11}$	$A_{1,12}$	$A_{1,13}$	$A_{1,14}$
Nb cas	1	2	6	16	50	156	532	1 856	6 876	26 200	104 456	428 352	1 821 846
Total	2	4	10	26	76	232	764	2 620	9 496	35 696	140 142	568 504	2 255 345
%	50	50	60,0	61,5	65,8	67,2	69,6	70,8	72,4	73,4	74,5	75,3	80,8

Tableau 2. *Pourcentage des cas éliminés par la condition d'éligibilité.*

4.2.8. Autre paramétrage

Dans la démarche que nous venons de présenter, nous avons proposé une recherche du degré de retournement par examen des différents cas possibles en commençant par ceux contenant une paire de sidis et en allant vers ceux contenant le plus grand nombre de paires. Un paramétrage approprié permet de faire une recherche locale d'un poids minimum, c'est-à-dire un poids minimum pour des cas possibles contenant un nombre donné de paires de sidi. Il suffit pour cela de préciser comme paramètre le nombre de paires minimum par cas possible, le nombre maximum par cas possible et le sens de la recherche (croissant ou décroissant).

5. Exemples d'application

L'algorithme présenté plus haut a été implémenté en langage Pascal et nous l'avons testé sur des exemples. Nous en présentons deux dans cette partie.

5.1 Exemple 1

Le graphe de la figure 8 présente le plan simplifié d'une ville avec ses rues et carrefours. Ce graphe planaire possède 16 sidis numérotés de 1 à 16 sur la figure. Combien de rues au minimum (degré de retournement de ce plan) faut-il mettre dans les deux sens et les autres en sens unique pour que chaque habitant puisse faire le tour de son quartier en respectant les sens imposés? L'algorithme proposé trouve neuf rues (neuf degrés de retournement) avec les sept paires de sidis suivantes :

(1,2); (3,4), (5,10), (6,7), (8,9), (11,16), (12,13) et deux sidis isolés à savoir 14 et 15.

Sur le plan les flèches indiquent les rues à double sens. Une solution équivalente consiste à apparié aussi les sidis 14 et 15. On obtient ainsi huit paires de sidis appariés. La chaîne minimale de la dernière paire de sidis étant égale à deux cela ne change pas le degré de retournement.

Le vecteur de profondeur associé au graphe de la figure 8 est présenté dans le tableau 3.

Numéro du Sidi	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
Profondeur	1	1	2	1	1	1	2	3	2	2	1	1	2	1	1	1

Tableau 3. *Profondeur des sidis du graphe de la figure 8*

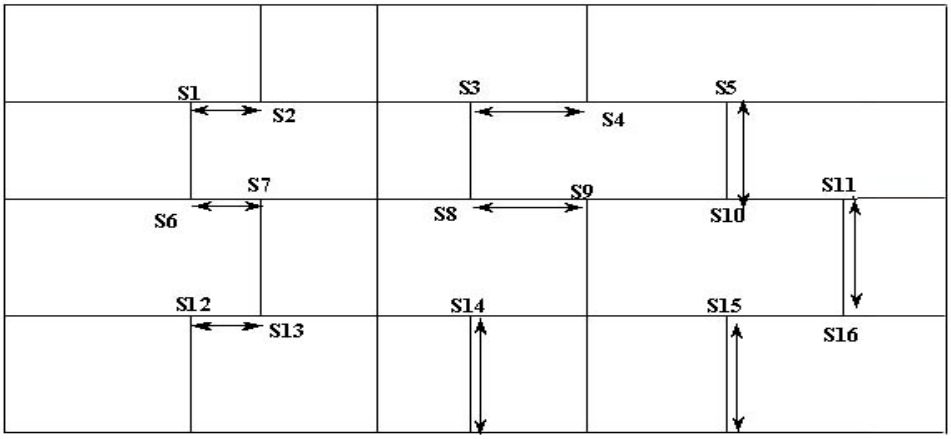

Figure 8. *Plan d'une ville*

La matrice de chaîne minimale associée au plan de la figure 8 est présentée dans le tableau 4.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
2	1														
3	3	2													
4	4	3	1												
5	5	4	2	1											
6	1	2	4	5	6										
7	2	3	3	4	5	1									
8	4	3	1	2	3	3	2								
9	5	4	2	3	2	4	3	1							
10	6	5	3	2	1	5	4	2	1						
11	7	6	4	3	2	6	5	3	2	1					
12	4	5	5	6	7	3	2	4	5	6	7				
13	3	4	4	5	6	2	1	3	4	5	6	1			
14	5	4	4	5	4	4	3	3	2	3	4	3	2		
15	7	6	4	5	4	6	5	3	2	3	2	5	4	2	
16	8	7	5	4	3	7	6	4	3	2	1	6	5	3	1

Tableau 4. *Matrice de chaîne minimale des sidis du graphe de la figure 8*

5.2 Exemple 2

Le deuxième exemple concerne la carte de la France avec son découpage en régions. Cette carte possède 23 sidis. L'algorithme a trouvé comme degré de retournement douze ainsi que le montrent les flèches placées sur la carte. Il suffit d'imposer une orientation à n'importe quelle arête non orientée dans les deux sens pour en déduire l'orientation des autres arêtes et aboutir au fait que chaque face est délimitée par un circuit.

Les temps de calcul sont élevés. Le cas de la carte de la France a nécessité plus de 23 heures de calcul sur un Pentium 333 Mhz avec l'algorithme non optimisé et une recherche systématique qui a examiné tous les cas possibles. Avec l'algorithme optimisé qui ne prend en compte que les cas possibles ne contenant que des paires éligibles, le calcul prend à peine deux minutes.

Dans le cas pratique de la carte de la France, le minimum absolu pour le degré de retournement est atteint (12 degré pour 23 sidis). On peut modifier la procédure POIDS pour que l'algorithme s'arête automatiquement quand le minimum absolu est atteint. Ce qui évite de continuer inutilement les recherches quand un cas possible de ce poids est trouvé.

Figure 9. *Carte de la France avec le découpage en régions.*

5.3 Exemple 3

Avant de présenter cet exemple nous allons donner quelques résultats permettant de mieux l'illustrer.

Théorème 3

Soit E l'ensemble des cas éligibles tel que Cmax = 1 et tout sidi isolé est de profondeur 1. Alors tout élément de E ayant un nombre maximal de paires est de poids minimum dans E.

Soit x un élément de E contenant m paires de sidis. Le poids associé à x est n–m ou n désigne le nombre de sidis. Pour un autre cas y contenant m – 1 paires, le poids

associé est n–(m-1) = n-m+1. Le poids de x est donc inférieur au poids de y donc au poids de tout cas contenant moins de paires de sidis.

La condition d'éligibilité d'un cas possible énoncée dans ce théorème 3 est surtout intéressante pour les graphes qui sont comme les cartes de géographie. En effet les cartes de géographie contiennent des sidis de degré trois et très rarement des sidis de degré pair.

La carte de la figure 10 est celle de l'Afrique avec son découpage en états. Cette carte possède 55 sidis. Pour évaluer son degré de retournement il a fallu en plus de l'algorithme optimisé choisir un sens de recherche à partir du nombre maximum de paires possibles (27 dans ce cas) et en allant vers 1. Nous avons aussi imposé la condition d'éligibilité énoncée dans le théorème 3. L'algorithme n'a trouvé aucun cas possible éligible avec 27 paires de sidis. Le premier cas éligible a 26 paires de sidis et donc trois sidis isolés tous de profondeur 1. Le poids correspondant à ce cas est de 29. Le degré de retournement de cette carte de l'Afrique est donc inférieur ou égal à 29. Mais comme aucun cas possible éligible ne possède 27 paires, 29 est effectivement le degré de retournement de cette carte. Ceci vient du simple fait que tout cas ayant 27 paires de sidis contient au moins une paire de chaîne minimale supérieure à 1 ou contient un sidi isolé de profondeur supérieur à 1.

C0 qui résume le cas possible associé au degré de retournement est le suivant:

-1	2	-3	9	-4	7	-5	10	-6	11	-8	12
-13	27	-14	15	-16	17	18	-19	20	-21	22	23
-24	25	-26	33	-28	29	30	-31	32	-34	37	-35
36	-38	55	-39	41	-40	48	-42	49	-43	44	-45
47	-46	54	-50	51	-52	53					

Les sidis isolés sont les sidis 18, 23 et 30. Ce résultat est obtenu en moins d'une minute. Pour examiner tous les 3 140 cas éligibles il faut cependant plus d'une heure sur un Pentium 133 MHZ. Parmi ces cas éligibles il y a plusieurs cas de poids égal à 29.

On peut se rendre compte que la numérotation initiale des sidis n'influence pas le résultat. Le sidi 3 est apparié au sidi 9, le sidi 26 au sidi 33 et le sidi 38 au sidi 55 par exemple.

Figure 10. *Carte d'Afrique avec le découpage en pays.*

Avec l'examen de ces trois exemples on se rend compte que d'autres améliorations peuvent encore être faites sur l'algorithme. Particulièrement en affinant la notion de cas éligible. On peut aussi modifier la procédure COMBINE pour qu'elle ne fasse les combinaisons que dans le cas où le poids des m-1 paires est au plus égal à p0-2. Si le poids des m-1 premières paires est au moins égal à p0-1, aucun des poids du nouveau cas à obtenir dans la procédure COMBINE ne peut être inférieur au minimum. En effet la dernière paire à constituer aura comme chaîne minimale au moins un.

6. Conclusion

Nos précédents travaux ont montré que la connaissance du degré de retournement d'une carte avec l'orientation des différentes arêtes associée permet de minimiser le nombre de retournement à effectuer dans la confection des contours lors de la digitalisation au millimètre. Le degré de retournement intervient aussi, comme nous l'avons signalé, dans les SIG fonctionnant selon le modèle vecteur topologique.

Dans ce présent document nous avons rappelé des formules permettant de calculer le degré de retournement d'un graphe dans le cas général. De ces formules, il ressort qu'il faut examiner un nombre de cas qui est une fonction croissante du nombre de sommets intérieurs de degré impair du graphe considéré. A partir de 6 sommets intérieurs de degré impair, cette fonction croît plus vite que la fonction exponentielle. C'est un problème d'explosion ou d'optimisation combinatoire. Pour examiner dans un temps acceptable tous ces cas possibles il faut utiliser des méthodes appropriées comme celles de séparation et d'évaluation.

Nous avons proposé un algorithme permettant de générer toutes les situations ou cas possibles et d'en déduire le degré de retournement pour tout graphe planaire topologique donné. Nous avons utilisé pour cela une méthode de séparation et d'évaluation avec un parcours en profondeur de l'arbre associé. Des améliorations de cet algorithme nous ont permis de l'appliquer en des temps d'exécution tout à fait raisonnables sur trois exemples pratiques de difficulté graduée. L'un des cas comporte cinquante cinq sommets intérieurs de degré impair. Les résultats sont présentés ici. Une analyse plus approfondie permettra de mieux optimiser l'approche et de réduire davantage les temps de calcul tout en abordant des graphes ayant un nombre de sommets intérieurs de degré impair toujours plus important.

7. Bibliographie

[1] BERGE C., *Graphes et hypergraphes,* Dunod, 1970.

[2] BERNARD M., *Gestion de projet de SIG Urbain*, Tutorial, 15° Symposium Européen des Systèmes d'Information Urbains (UDMS), Lyon, 1992.

[3] CGN (Centre Géographique National), *Carte Administrative du Cameroun au 1/2 000 000*, 7° édition, 1984.

[4] CORMEN T., LEISERSON C., RIVEST R., *Introduction à l'algorithmique,* Dunod, 1994.

[5] FAIZ S., BOURSIER P., *Geographic Data Quality : From Assessment to Exploitation*, Cartographica, n° 33: 1/Carto, 1997, Mars 1997, pp. 33-40.

[6] FAIZ S., NZALI J.P., BOURSIER P., *Representing Quality Depending on the use context,* Proc. Joint European Conference and Exhibition on Geographical Information (JEC'96) Barcelone, Espagne, p. 73-77, Mars 1996.

[7] FRED S. ROBERTS, *Graph Theory and its applications to problems of society,* CBMS-NSF, Regional Conference Seties in Applied Mathematics, Vol 29, p. 7-13, Philadelphia, Pensylvania, 1978.

[8] GONDRAN M., MINOUX M., *Graphes et Algorithmes*, Eyrolles, 1979.

[9] INC (Institut National de Cartographie), *Carte Administrative du Cameroun au 1/1 500 000*, édition, 1996.

[10] LAURINI R., MILLERET-RAFFORT F., *Les bases de données en géomatique*, Hermès, 1993.

[11] LAURINI R., THOMPSON D., Fundamentals of spatial information systems, The APIC series, Academic Press, 1995.

[12] LEVY G., *Algorithmique combinatoire, méthodes constructives,* Dunod, 1994.

[13] NZALI J.P., TAPAMO H., *Digitalisation au Millimètre*, 3° Colloque Africain sur la Recherche en Informatique (CARI'96), p. 237-246, Libreville, Gabon, Octobre 1996.

[14] NZALI J.P., TAPAMO H., *Analyse de la Digitalisation au Millimètre*, TSI, Vol 17, n° 6, Hermès, juin 1998,

[15] NZALI J.P., *Degré de retournement d'une carte*, Revue Internationale de Géomatique, Vol 9, n° 2, Hermès Science 1999.

[16] NZALI J.P., KOUMPO T.P., TAPAMO H., *Calcul du degré de retournement d'un graphe*, 5° Colloque Africain sur la Recherche en Informatique (CARI'2000), p. 57-64, Antananarivo, Madagascar, Octobre 2000.

[17] PETTANG, KOUAMOU, *Pour un système interactif d'aide à la décision pour la résorption de l'habitat spontané en milieu urbain*, revue des Systèmes de Décision, vol 6-2, Hermès, 1997.

[18] PORNON H., *Utilisation et place des SIG dans les systèmes d'information des organisations*, Revue de géomatique, Vol 3- n° 1-2/1993, Hermès.

[19] OYSTEIN O., *Graphs and their uses*, Random House, The L.W. Singer Compagny, 1963.

[20] ROUET P., *Les données dans les systèmes d'information géographique*. Hermès. 1991.

[21] SAKAROVITCH M., *Optimisation Combinatoire, programmation discrète,* Hermann, 1984.

[22] SLOANE., *Handbook of integer sequences, suite 469.* Academic Press, 1973.