RAIRO. RECHERCHE OPÉRATIONNELLE

JACQUES THÉPOT
GÉRARD LECHENAULT

Note sur une application de la classification hiérarchique à la coloration des graphes

RAIRO. Recherche opérationnelle, tome 15, n° 1 (1981), p. 73-83
<http://www.numdam.org/item?id=RO_1981__15_1_73_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 15, n° 1, février 1981, p. 73 à 83)

NOTE SUR UNE APPLICATION DE LA CLASSIFICATION HIÉRARCHIQUE A LA COLORATION DES GRAPHES (*)

par Jacques THÉPOT ([1]) et Gérard LECHENAULT ([2])

Résumé. — *Cette note présente une méthode itérative de coloration de graphes qui peut s'interpréter de deux manières : soit comme une méthode d'optimisation dérivée de l'algorithme primal du simplexe, soit comme une méthode de classification hiérarchique ascendante.*

A titre indicatif, on présente les résultats obtenus dans la coloration de la carte des départements français.

Mots clés : Classification, Coloration des graphes, Dissimilarité.

Abstract. — *This note deals with an approximate method for graph coloring which can be interpreted in two ways: first, as an optimization method derived from the simplex algorithm in L. P., and secondly as a hierarchical clustering procedure. Computational experience with the coloration of the French departments map is reported on.*

Keywords: Cluster Analysis, Dissimilarity, Graph Coloring.

1. INTRODUCTION

1.1. Les méthodes de coloration de graphes sont utilisées pour résoudre certains problèmes de planning (planning d'examens en particulier). Depuis ces dernières années, la mise au point de telles méthodes a fait l'objet de nombreuses publications.

La littérature propose deux grandes variétés de méthodes :

— *Les méthodes directes* : fondées sur la programmation dynamique (Christofidès [3], Wang [15]) ou sur des procédures d'énumération du type « Branch and Bound » (Randall-Brown [10]) ces méthodes permettent de déterminer la valeur exacte du nombre chromatique; mais elles exigent pour cela un temps de calcul qui est une fonction non polynomiale du nombre de sommets. Le problème de la coloration d'un graphe est en effet un problème dit N.P-complet;

(*) Reçu septembre 1979.

([1]) Institut européen de Recherches et d'Études supérieures en Management, place Stephanie 20, B-1050 Bruxelles et Université Paris-X, Nanterre.

([2]) Université Paris-X, Nanterre.

R.A.I.R.O. Recherche opérationnelle/Operations Research, 0399-0559/1981/ 73 /$ 5.00

– *les méthodes approchées* sont moins gourmandes en temps calcul. Elles s'appuyent sur des considérations heuristiques. La qualité des estimations du nombre chromatique qu'elles donnent dépend de la structure du graphe (Matula [9], Thérani [13]).

1.2. Nous proposons ici une méthode d'un esprit différent en considérant le problème de la coloration d'un graphe comme un problème particulier d'analyse des données et plus spécialement de classification.

Nous cherchons la coloration du graphe qui optimise un certain critère lié à la finesse de la partition. La dépendance de ce critère à l'égard du nombre de classes n'est pas univoque; c'est ainsi le problème lui même qui est un problème *approché* mais la méthode employée est une méthode *exacte*. Elle s'interprète en effet de deux manières :

– comme une *méthode d'optimisation*, analogue à une méthode de gradient pour un problème non convexe;

– comme une méthode de *classification hiérarchique ascendante* (*cf.* Benzecri [1], Jambu [7]) ([3]).

A titre indicatif, nous présentons l'application de la méthode à la coloration de la carte des départements français. Les résultats obtenus sont encourageants. La souplesse d'utilisation et sa simplicité (*cf.* Thépot [11]) semblent de bons atouts pour l'élaboration d'une procédure interactive d'aide à la décision.

2. POSITION DU PROBLÈME

2.1. Notations

Soit I un ensemble fini à n éléments et $\mathscr{J}_p$ une partition de l'ensemble I en p classes :

$$\mathscr{J}_p = \{J_1, J_2, \ldots, J_p\},$$

avec :

$$\bigcup_{k=1}^{p} J_k = I, \qquad J_k \cap J_l = \emptyset, \quad \forall k \neq l.$$

soit $n_k = \text{card } J_k$.

([3]) P. Hansen et M. Delattre (*cf.* [6]) utilisant une méthode de coloration de graphes pour construire une classification ascendante. Il est intéressant de constater que nous faisons ici exactement l'inverse.

A cette partition, nous associons un ensemble de n^2 variables binaires $x_{ij}(\mathscr{J}_p)$ définies par :

$$x_{ij}(\mathscr{J}_p)=\left\{\begin{array}{ll}1 & \text{si } \exists k/i \text{ et } j\in J_k,\\ 0 & \text{sinon.}\end{array}\right\} \tag{1}$$

Les variables x_{ij} vérifient les propriétés suivantes :

$$x_{ij}=x_{ji}, \tag{2}$$

$$x_{ii}=1, \tag{3}$$

$$x_{ij}.x_{jk}\leqq x_{ik}, \qquad i, j, k\in I. \tag{4}$$

L'inégalité (4) exprime que les classes de la partition sont disjointes.

PROPOSITION 1 : *Soient x_{ij} n^2 variables binaires vérifiant* (2), (3) *et* (4), *alors il existe une partition $\mathscr{J}_p$ de l'ensemble I telle que :*

$$x_{ij}=x_{ij}(\mathscr{J}_p).$$

Démonstration évidente ■

2.2. Définition et propriétés de la fonction v

Nous définissons la fonction v de la manière suivante :

$$v(\mathscr{J}_p)=\sum_{i,\,j=1}^{n} x_{ij}(\mathscr{J}_p). \tag{5}$$

Il est clair que la valeur de $v(\mathscr{J}_p)$ est donnée par :

$$v(\mathscr{J}_p)=\sum_{k=1}^{p} n_k^2, \tag{6}$$

d'où l'on déduit les inégalités :

$$n\leqq v(\mathscr{J}_p)\leqq n^2, \tag{7}$$

quelle que soit la partition $\mathscr{J}_p$ de l'ensemble I.

PROPOSITION 2 : *Soient $\mathscr{J}_p$ et $\mathscr{H}_q$ deux partitions de l'ensemble I, alors :*

$$\mathscr{J}_p\subset\mathscr{H}_q \quad\Rightarrow\quad v(\mathscr{J}_p)\leqq v(\mathscr{H}_q).$$

Démonstration évidente. ■

Cette proposition indique que la fonction v croît le long d'une suite de partitions emboîtées.

PROPOSITION 3 : *Soit $\mathscr{J}_p$ une partition de l'ensemble I en p classes, alors la fonction $v(\mathscr{J}_p)$ est bornée selon les inégalités :*

$$\frac{n^2}{p} \leqq v(\mathscr{J}_p) \leqq n^2 - (p-1)(2n-p).$$

Démonstration : *voir* Thépot-Lechenault [12].

2.3. Application à la coloration des graphes

Soit $G=(I, U)$ un graphe défini par un ensemble I de n sommets et un ensemble $U \subset I \times I$ d'arcs (non orientés).

La coloration du graphe G qui maximise la fonction v s'obtient en résolvant le problème d'optimisation suivant :

$$\text{Max} \sum_{i,j=1}^{n} x_{ij}, \tag{8}$$

$$x_{ij}=0 \quad \text{pour} \quad (i,j) \in U, \tag{9}$$

$$x_{ij}=x_{ji}, \tag{10}$$

$$x_{ii}=1, \tag{11}$$

$$x_{ij}.x_{jk} \leqq x_{ik}, \quad \forall i, j, k, \tag{12}$$

$$0 \leqq x_{ij} \leqq 1. \tag{13}$$

La résolution de ce problème d'optimisation ne peut fournir qu'une valeur approchée du nombre chromatique du graphe, pour deux raisons :

(*a*) le critère optimisé n'est pas le nombre de classes de la partition mais la fonction v qui en dépend de manière non univoque. Il s'agit cependant d'une dépendance monotone sur toute suite de partitions emboîtées (proposition 2), d'autant meilleure que ces partitions sont équilibrées (proposition 3);

(*b*) les contraintes du type (12) sont des contraintes non convexes; *a priori*, un tel problème peut admettre plusieurs optimums locaux.

3. MÉTHODE DE RÉSOLUTION

La méthode de résolution proposée s'interprète de deux façons :

(*a*) comme une méthode d'optimisation, dérivée de la méthode primale du simplexe de la programmation linéaire;

(*b*) comme une méthode de classification hiérarchique ascendante.

3.1. Principe de la méthode du point de vue de l'optimisation

On part de la solution admissible définie par :

$$x_{ii}=1, \qquad x_{ij}=0, \qquad i\neq j.$$

On associe à cette solution une base constituée d'une part, des variables x_{ii} au niveau 1 et d'autre part, des variables x_{ij} pour $(i, j)\in U$. Ces dernières sont donc dans la base à *un niveau* 0 [du fait des contraintes (9)].

On cherche donc à faire rentrer dans la base *au niveau* 1 une variable x_{ij} qui ne s'y trouve pas $(i, j)\in I\times I-U$, $i\neq j$, de manière à augmenter la valeur de la fonction objectif. Mais la prise en compte des contraintes (12) oblige à introduire alors dans la base *au niveau* 0 un certain nombre de variables hors base. En effet, des relations (12), on déduit les inégalités :

$$x_{ij}.x_{jk}\leqq x_{ik} \qquad \text{et} \qquad x_{ij}.x_{ik}\leqq x_{jk}.$$

C'est-à-dire les implications :

$$x_{ik}=0 \;\Rightarrow\; x_{jk}=0 \qquad \text{et} \qquad x_{jk}=0 \;\Rightarrow\; x_{ik}=0, \tag{14}$$

puisque $x_{ij}=1$. Autrement dit, si la variable x_{ik} est en base au niveau 0, la variable x_{jk} doit l'y rejoindre (et réciproquement). L'introduction dans la base au niveau 1 d'une variable x_{ij} augmente automatiquement l'ensemble U, c'est-à-dire crée de nouveaux arcs dans le graphe.

On définit ainsi l'ensemble A_i^j :

$$A_i^j=\{k\,|\,(j, k)\notin U \text{ et } (i, k)\in U\}. \tag{15}$$

Maximiser la fonction objectif revient donc à minimiser le nombre total de variables qui rentrent dans la base au niveau 0.

Ainsi, on fera rentrer dans la base au niveau 1, la variable x_{ij} telle que :

$$\text{Card } A_i^j+\text{Card } A_j^i=\text{Min } (\text{Card } A_r^s+\text{Card } A_s^r),$$
$$(r, s)\in U, \quad r\neq s.$$

Et l'on définit alors un nouvel ensemble d'arcs $U^2=U\cup(A_i^j\times\{j\})\cup(A_j^i\times\{i\})$. Et ainsi la procédure se poursuit de manière itérative par adjonction d'arcs supplémentaires au graphe. Elle s'arrête lorsque toutes les variables x_{ij} sont entrées dans la base, soit au niveau 0, soit au niveau 1. De ce point de vue, la méthode est une méthode itérative localement de plus grande croissance.

3.2. Liens avec la classification hiérarchique ascendante

La solution initiale de la procédure se représente par un tableau carré et symétrique S. Les éléments de la diagonale sont égaux à 1, ceux qui correspondent à l'ensemble U sont nuls. Les autres sont provisoirement laissés en blanc. Cette solution correspond à la partition fine $\mathscr{J}_n$.

Soit :

$$d_{ij}=\text{Card } A_i^j+\text{Card } A_j^i \qquad \text{pour} \quad (i,j)\notin U, \quad i\neq j.$$

Le nombre d_{ij} représente le nombre d'arcs supplémentaires générés par la constitution d'une classe $\{i,j\}$. Il se calcule aisément comme étant l'*indice de dissimilarité* des éléments (i,j), c'est-à-dire le nombre de configurations du type (0, *) ou (*, 0) que l'on rencontre en balayant les lignes i et j; l'astérisque * désigne ici soit un blanc, soit un 1.

En posant $d_{ij}=+\infty$ pour tout $(i,j)\in U$, on définit un tableau des indices de distance T. Ce tableau contient toutes les informations nécessaires pour le déroulement de la méthode. Il suffit ainsi de travailler sur les variables d_{ij}.

La méthode est de la sorte une méthode de classification hiérarchique ascendante utilisant l'indice de dissimilarité (défini sur $\overline{R}$*) comme indice de distance.*

3.3. Algorithme de résolution

La méthode se déroule de manière itérative par modification progressive du tableau T.

A l'étape 1 : on définit des ensembles K_1 et O_1 par $K_1=U$, $O_1=\{(i,i)\,|\,i\in I\}$. On détermine le tableau $T_1=T$ des indices de dissimilarité $\{d_{ij}^1\}$. On a ainsi :

$$d_{ij}^1=0 \qquad \text{pour} \quad (i,j)\in O_1, \quad d_{ij}^1=+\infty \qquad \text{pour} \quad (i,j)\in K_1.$$

A l'étape t : soit $T_t=\{d_{ij}^t\}$ le tableau des distances. On définit les ensembles K_t et O_t et P_t par :

$$K_t=\{(i,j)\,|\,d_{ij}^t=+\infty\},$$
$$O_t=\{(i,j)\,|\,d_{ij}^t=0\},$$
$$P_t=I\times I-(K_t\cup O_t).$$

Soit $(i_t j_t)$ un couple d'éléments de P_t vérifiant :

$$d_{i_t j_t}^t=\underset{(i,j)\in P_t}{\text{Min}}\ d_{ij}^t. \tag{16}$$

Soient :

$$-A_{i_t}^{j_t}=\{k\in I \mid d_{j_t k}^t<d_{i_t k}^t=+\infty\},$$

$$-A_{j_t}^{i_t}=\{k\in I \mid d_{i_t k}^t<d_{j_t k}^t=+\infty\}.$$

On pose alors :

$$d_{j_t k}^{t+1}=+\infty \quad \text{pour } k\in A_{i_t}^{j_t} \qquad \text{et} \qquad d_{i_t k}^{t+1}=+\infty \quad \text{pour } k\in A_{j_t}^{i_t}. \tag{17}$$

D'où l'on déduit les valeurs modifiées des autres distances.

Soient :

$$e_{kl}^t=\left\{\begin{array}{ll} +1 & \text{si } d_{kl}^t<+\infty, \\ -1 & \text{si } d_{kl}^t=+\infty, \end{array}\right\} \tag{18}$$

les règles de calcul des d_{ij}^{t+1} sont les suivantes :

$$\left.\begin{array}{l} d_{li_t}^{t+1}=d_{li_t}^t=\sum\limits_{k\in A_{j_t}^{i_t}} e_{kl}^t \qquad \text{pour } l\in I-A_{j_t}^{i_t}, \\ d_{mk}^{t+1}=d_{mk}^t+e_{i_t m}^t \qquad \text{pour } k\in A_{j_t}^{i_t} \text{ et } m\in I-A_{j_t}^{i_t}; \end{array}\right\} \tag{19}$$

$$\left.\begin{array}{l} d_{rj_t}^{t+1}=d_{rj_t}^t=\sum\limits_{k\in A_{i_t}^{j_t}} e_{kr}^t \qquad \text{pour } r\in I-A_{i_t}^{j_t}, \\ d_{sk}^{t+1}=d_{s_k}^t+e_{js}^t \quad \text{pour } k\in A_{i_t}^{j_t}, \quad s\in I-A_{i_t}^{j_t}. \end{array}\right\} \tag{20}$$

D'où le tableau T_{t+1}.

Les relations (19) et (20) montrent qu'une partie seulement du tableau T est modifiée à chaque étape. Il suffit donc d'extraire le tableau $\hat{T}_t$ suivant pour calculer les nouvelles distances :

$$\hat{T}_t=[(I-A_{j_t}^{i_t})\times(\{i_t\}\cup A_{j_t}^{i_t})\cup(I-A_{i_t}^{j_t})\times(\{j_t\}\cup A_{i_t}^{j_t})]-K_t. \tag{21}$$

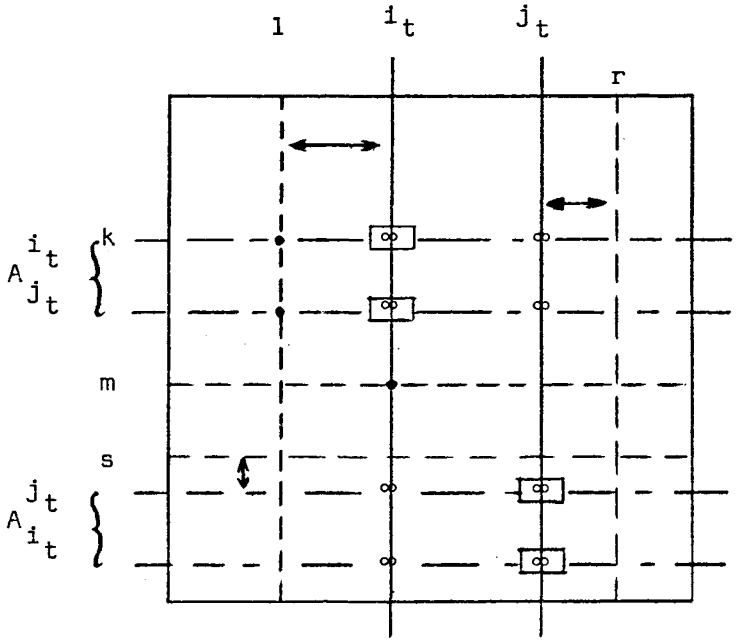
La procédure s'arrête lorsque le tableau T ne contient que des « 0 » ou des « ∞ ».

La relation (17) s'écrit aussi :

$$d_{j_t k}^{t+1}=d_{i_t k}^{t+1}=\mathrm{Sup}(d_{j_t k}^t,\, d_{i_t k}^t) \qquad \text{pour } k\in A_{i_t}^{j_t}\cup A_{j_t}^{i_t}.$$

Ainsi, les distances des éléments de l'ensemble $A_{i_t}^{j_t}\cup A_{j_t}^{i_t}$ au couple $\{i_t, j_t\}$ sont calculées selon le critère d'agrégation de l'« *ultramétrique supérieure minimale* » (*cf.* Benzecri [1]), généralisé à des distances infinies.

La méthode est donc une méthode de classification hiérarchique ascendante. A l'inverse de ce que l'on rencontre dans les problèmes de taxinomie, il n'y a ici aucun arbitraire dans le choix de l'indice de distance et du critère d'agrégation. Tout y est défini de manière canonique.



Il est possible de définir de manière analogue une méthode de classification descendante dérivée de l'algorithme *dual* du simplexe de la programmation linéaire (*cf.* Thépot-Lechenault [12]). Il s'agit d'une méthode itérative : à chaque étape l'on détermine un *recouvrement* de l'ensemble I de plus en plus fin jusqu'à obtenir une *partition*.

4. APPLICATIONS NUMÉRIQUES

Nous avons testé la méthode sur la coloration de la carte des 95 départements français. On sait en effet que le nombre chromatique d'un graphe planaire est inférieur ou égal à 4.

Les essais numériques ont été effectués sur l'ordinateur IBM 370/168 du C.I.R.C.E. d'Orsay.

Le programme informatique présente les caractéristiques suivantes :

(*a*) à chaque itération l'agrégation porte sur les représentants des classes de la partition;

(*b*) en cas d'indifférence dans l'application de la règle (16), le programme agrège les éléments qui vont donner naissance à la plus petite classe [*cf.* remarque (*a*) section 2.3]. Il est même possible d'intégrer cet élément directement dans le calcul des distances;

(*c*) à chaque itération, le programme détermine le tableau (21). Seules les distances effectivement modifiées sont recalculées.

4.1. Premiers résultats par application directe de la méthode

Nous avons appliqué le programme tel quel à la coloration de cartes de 35, 45, ..., 95 départements extraites de la carte (*voir* tableau). On constate donc que la méthode fournit une approximation bien grossière du nombre chromatique puisque l'on obtient cinq couleurs au-delà de 45 sommets. Par contre, les temps de calcul sont assez raisonnables. Il est connu qu'une méthode de classification hiérarchique nécessite un temps de calcul d'ordre $\sigma(N^3)$ et une occupation de mémoire centrale d'ordre $\sigma(N^2)$. Compte tenu du fait qu'à chaque itération fort peu de distances doivent être recalculées, le temps de calcul semble pouvoir être estimé par un polynôme de degré 2 seulement. Ce que confirme la relative stabilité du rapport T/N^2.

4.2. Amélioration des résultats par partition préalable du graphe

L'algorithme de résolution a été modifié en appliquant plusieurs fois de suite la méthode :

— on l'applique une première fois sur le graphe complémentaire, de manière à obtenir une partition de l'ensemble des sommets. Chaque classe de cette partition est ainsi constituée de départements voisins (4). On ordonne les classes obtenues selon une suite $G_1, G_2, \ldots, G_p$ de sorte que deux classes successives aient le plus grand nombre d'adjacences entre elles. Ceci revient à réordonner les sommets de telle façon que les « ∞ » se trouvent concentrés autour de la diagonale;

— on applique ensuite la méthode de manière progressive : on colorie G_1 puis $G_1 \cup G_2$ sans modifier la coloration obtenue sur G_1, puis $G_1 \cup G_2 \cup G_3$ sans modifier la coloration obtenue sur $G_1 \cup G_2$ et ainsi de suite.

En appliquant la méthode progressive aux mêmes cartes de 35, 45, ..., 95 départements, on obtient les résultats du tableau I :

(4) *Voir* Titli [14] pour une application des méthodes de coloration à la partition des graphes.

On constate :

– le nombre de couleurs obtenu est égal au nombre chromatique;

– le temps d'exécution n'a pas augmenté; il est même inférieur à celui de la méthode directe, alors que finalement on l'applique plusieurs fois !

TABLEAU I

	Méthode directe			Méthode progressive		
N	T	g	$(T/N^2).10^2$	T	g	$(T/N^2).10^2$
35.	0,87	4	0,071	0,91	4	0,074
45.	1,46	5	0,072	1,48	4	0,073
55.	2,28	5	0,075	2,17	4	0,071
65.	3,42	5	0,080	3,07	4	0,072
75.	4,95	5	0,088	4,23	4	0,075
85.	6,76	5	0,093	5,78	4	0,080
95.	8,17	5	0,090	7,36	4	0,081

N, nombre de sommets; T, temps d'exécution C.P.U. (en secondes); g, nombre de couleurs.

5. CONCLUSION

La méthode proposée s'interprète de deux façons :

– comme une méthode dérivée de l'algorithme du simplexe, c'est-à-dire une méthode itérative localement de plus grande croissance;

– comme une application de la classification hiérarchique ascendante.

On établit de la sorte un lien entre les problèmes combinatoires et les méthodes d'analyse des données. Il semble qu'il y ait là un domaine d'investigation assez intéressant.

A l'heure actuelle et sans préjuger d'une amélioration ultérieure de ses performances numériques, la méthode se révèle un peu moins bonne que les méthodes exactes de Brélaz [2] ou Laurière [8].

Mais il nous semble d'ores et déjà que la simplicité de la méthode est en soi un élément appréciable du point de vue des applications aux problèmes de planning et d'allocation de ressources.

BIBLIOGRAPHIE

[1] J. P. BENZECRI, *L'analyse des données*, Dunod, Paris, t. 1, 1973.

[2] D. BRELAZ, *New Methods to Color the Vertices of a Graph*, Comm. A.C.M., vol. 22, n° 4, 1979, p. 251-256.

[3] N. CHRISTOFIDES, *Graph Theory; an Algorithmic Approach*, Londres, Academic Press, 1975.

[4] D. G. CORNEIL et B. GRAHAM, *An Algorithm for Determining the Chromatic Number of a Graph*, S.I.A.M. J. Comput., vol. 2, n° 4, 1973, p. 311-318.

[5] A. DEFRENNE, *The time tabling Problem: a Survey*, Cahiers Centre d'études de Recherche opérationnelle, Bruxelles, vol. 20, n° 2, 1978, p. 163-169.

[6] P. HANSEN et M. DELATTRE, *Complete-Link Cluster Analysis by Graph Coloring*, J.A.S.A., vol. 73, n° 362, 1978, p. 397-403.

[7] M. JAMBU, *Classification Automatique en Analyse des Données*, Dunod, Paris, 1978.

[8] J. L. LAURIÈRE, *Un langage et un programme pour énoncer et résoudre des problèmes combinatoires*, Thèse d'État, Paris, 1976.

[9] D. W. MATULA, G. MARBLE et J. D. ISAACSON, *Graph Coloring Algorithms* dans *Graph Theory and Computing*, New York, Academic Press, 1972, p. 109-122.

[10] J. RANDALL-BROWN, *Chromatic Scheduling and the Chromatic Number Problems*, Management Science, vol. 19, n° 4, 1972, p. 456-463.

[11] J. THÉPOT, *A propos d'un problème de planning d'examens*, Enseignement et Gestion, Paris, vol. 12, 1979, p. 69-74.

[12] J. THÉPOT et G. LECHENAULT, *Méthode de Classification pour la Coloration des Graphes*, E.I.A.S.M., WP n° 79-29, Bruxelles 1979.

[13] A. THÉRANI, *Un algorithme de coloration*, Cahiers du Centre d'Études de Recherche opérationnelle, Bruxelles, vol. 117, n° 2-3-4, 1975, p. 395-398.

[14] A. TITLI, *et al.*, *Analyse et Commande des Systèmes complexes*, monographie A.F.C.E.T., Toulouse, Cepadues éditions, chap. V, 1979.

[15] C. G. WANG, *An Algorithm for the Chromatic Number of a Graph*, J. A.C.M., vol. 21, n° 3, 1974, p. 385-391.

[16] D. DE WERRA, *A Note on Graph Coloring*, R.A.I.R.O., R1, 1974, p. 49-53.