

ALGORITHMES DE COLORATION DE GRAPHE TENANT COMPTE D'UNE LARGE GAMME DE  
CONTRAINTES DANS L'ETABLISSEMENT D'UN HORAIRE DE COURS D'UNE INSTITUTION  
SUPERIEURE OU UNIVERSITAIRE

---

O. Henkes  
et  
J.P. Leclercq  
F.N.D.P. Namur

Abstract

This article is devoted to the building of timetables using graph-coloration algorithms.

A time-matrix is used in order to take into account the various kinds of pedagogical constraints.  
A lot of methods are described and compared on practical basis by means of simulations using a package implemented by O. Henkes.

Résumé

Nous avons élaboré un programme de calcul d'un horaire de cours d'une institution universitaire.  
Après enquête sur le terrain et suite à notre expérience personnelle il est apparu que pour être vraiment utile, ce programme devait pouvoir prendre en considération les habitudes pédagogiques des professeurs.  
Outre les classiques contraintes d'indisponibilités et d'incompatibilités il tient compte de desiderata plus particuliers tels que "il est souhaitable que les cours A et B soient consécutifs" ou "le professeur X ne désire donner qu'un cours par jour".

L'algorithme, utilisant la coloration des graphes, prend en compte l'ensemble des contraintes par l'intermédiaire d'une matrice booléenne dite "de temps" qui est mise à jour en cours de coloration.

Le logiciel comprend la plupart des méthodes de coloration connues (séquentielles par degrés, valeurs propres ou nombres d'éléments ; similarité ; saturation ; permutation) ainsi que des méthodes de perturbation permettant d'ajouter un petit nombre de cours ou de contraintes à un horaire existant en le

modifiant le moins possible.

La modularité du programme et la généralité d'emploi de la matrice de temps rendent particulièrement simple l'ajout de nouveaux types de contraintes et devraient permettre son adaptation à d'autres cas (enseignement moyen, affectation de personnel).

Outre la partie calcul décrite ci-avant, une gestion de fichiers de données conversationnelle et une édition dédiée (par local, classe, professeur, etc) complètent le produit.

Le présent article met l'accent sur les possibilités de modélisation liées à la matrice de temps ; en outre il constitue un petit "survey" des méthodes de coloration qui sont comparées sur base de simulations.

## 1. MODELISATION DU PROBLEME

### 1.1. Position du problème et le modèle de graphes

Soient  $X = \{x_r \mid r \in I_0\}$  l'ensemble des cours à donner

$C = \{c_i \mid i \in I_1\}$  l'ensemble des classes ou sections

$P = \{p_j \mid j \in I_2\}$  l'ensemble des professeurs

$L = \{l_k \mid k \in I_3\}$  l'ensemble des locaux

Un cours  $x$  est un n-uple inclus dans  $C \cup P \cup L$

Exemple : "Le professeur  $p_1$  donne le cours de programmation linéaire ( $x_7$ ) dans le local  $l_8$  aux 1e LIC. MATH ( $c_6$ ) et 2e LIC. INFO ( $c_{19}$ )"

se traduira pas  $x_7 = \{c_6, c_{19}, p_1, l_8\}$

Cette approche permet de tenir compte de cas spéciaux tels :

- un cours dont le titulaire et le local ne sont pas encore connus :  $x_r = \{c_i, , \}$

- deux cours différents  $x_r$  et  $x_s$  suivis en même temps par deux moitiés de classe

$x_r \cup x_s = x_{rs} = \{c_i, p_{j1}, p_{j2}, l_{k1}, l_{k2}\}$

(exemple : option néerlandais et anglais)

Les contraintes d'incompatibilité d'horaire entre deux cours peuvent se représenter au moyen d'un graphe  $G(X,U)$  où  $U = \{(x_i, x_j) \mid x_i \in X, x_j \in X; x_j \cap x_i \neq \emptyset; x_i \neq x_j\}$

$G$  est un graphe simple sans boucles dont la matrice d'incidence est  $M$ .

$M_{ij} = 0$  si  $(x_i, x_j) \notin U$

$M_{ij} = 1$  si  $(x_i, x_j) \in U$

### 1.2. Les méthodes de coloration

Une coloration des sommets d'un graphe  $G(X,U)$  est une application surjective  $f$  de  $X$  dans  $H$  (un ensemble de couleurs) telle que :

si  $(x_i, x_j) \in U$  alors  $f(x_i) \neq f(x_j)$

Dans la suite on ne parlera que de coloration de sommets en disant simplement coloration.

Si à chaque élément de  $H$  on associe une heure de cours, réaliser une coloration de  $G$  c'est établir un horaire tenant compte des contraintes d'incompatibilité.

Il existe plusieurs résultats théoriques donnant une borne supérieure du nombre chromatique  $N = \min \# H$ . Réf [1, 2, 14, 15, 17]  
{f}

### 1.2.1. Coloration séquentielle

L'idée de base de ces méthodes est de classer les sommets dans un ordre lié à leur difficulté de coloration puis ensuite de colorer les sommets un à un selon cet ordre. Réf [10].

#### 1.2.1.1. Par degrés

Intuitivement, si un sommet a un degré élevé, il est prioritaire car il comporte un professeur très occupé et/ou un local très demandé et/ou une classe à horaire chargé.

Deux méthodes de rangement ont été implémentées pour lesquelles on dispose de résultats théoriques :

- largest first (LF) . Ref [3,4]
- smallest last (SL) . Ref [6,7,8,9]

N.B. Dans la méthode SL on recalcule les degrés chaque fois qu'un sommet a été classé (en fin de liste), dans la méthode LF le degré est calculé une seule fois.

#### 1.2.1.2. Par valeur propre (VP)

La centralité d'un sommet dans le graphe peut être mesurée par sa contribution au vecteur propre associé à la valeur propre dominante du graphe. Dans cette méthode, les sommets seront classés selon les valeurs absolues décroissantes des composantes du dit vecteur propre.

#### 1.2.1.3. Par nombre d'éléments (NELE)

On classe les sommets  $x$  selon  $\# x$  :  
Cette méthode n'est intéressante que si certains cours sont suivis par plusieurs classes ; c'est un cas fréquent en candidature en Sciences, par exemple.

### 1.2.2. Coloration par saturation (SAT)

Dans les méthodes séquentielles (1.2.1) l'ordre des sommets est établi avant de débiter la coloration et n'est plus modifié. Or le degré de priorité d'un sommet peut changer au fur et à mesure de la coloration de ses voisins, les méthodes de coloration par saturation tiennent compte de ce fait et réordonnent les sommets non colorés après chaque coloration de l'un d'entre eux.

Pour le réordonnement, n'importe laquelle des quatre méthodes de 1.2.1 peut être utilisée.  
On trouvera un résultat théorique dans [11].

### 1.2.3. Coloration par similarité (SIMG)

Les méthodes précédentes coloraient individuellement les sommets ; une autre approche serait de trouver des sommets complémentaires et de leur attribuer la même couleur. Intuitivement, deux sommets sont complémentaires s'ils sont incompatibles (adjacents) avec beaucoup d'autres communs, sans l'être eux-mêmes.

Soit S la matrice de similarité de dimension (#X,#X) définie par

$$S_{ij} = 0 \quad \text{si } (x_i, x_j) \in U \quad \text{ou si } x_i = x_j$$
$$S_{ij} = \sum_{\substack{k \neq i \\ k \neq j}} M_{ik} M_{jk}$$

On coloriera en premier lieu les couples  $x_i, x_j$  tels que

$$S_{ij} \geq S_{rs} \quad \text{pour tout } (r,s)$$

Ici, l'algorithme est un peu plus complexe car à l'itération n le couple de similarité maximum trouvé peut déjà avoir été coloré partiellement ou entièrement lors d'une étape antérieure ; c'est pourquoi, nous donnons la procédure complète. [Réf 12]

1. Construire S

2.  $\text{Max} = \max_{ij} S_{ij}$

3. Soit (i,j) une paire non traitée telle que  $S_{ij} = \text{Max}$ .

- Si une telle paire existe aller en 4  
 Sinon :  $Max = Max - 1$   
     Si  $Max > 0$  aller en 3  
     Sinon aller en 7
4. Si  $i$  et  $j$  sont déjà colorés aller en 3.
5. Si  $i$  est coloré  $f(i)$  et  $j$  non coloré  
     Essayer de donner à  $j$  la couleur  $f(i)$   
         Si cela réussit, interdire la couleur  $f(i)$  aux adjacents de  $j$   
         Sinon aller en 3.
6. Si  $i$  et  $j$  sont non colorés  
     Donner à  $i$  et  $j$  la première couleur disponible soit  $r$   
     Interdire la couleur  $r$  aux adjacents de  $i$  et de  $j$   
     Aller en 3.
7. Colorer les sommets restants par une des méthodes précédentes.

#### 1.2.4. Coloration par permutation bichromatique

Les méthodes précédentes ne remettent pas en question la couleur des sommets traités aux étapes antérieures. La méthode de permutation essaie, lorsque pour colorer un sommet une nouvelle couleur s'avère nécessaire, de permuter les couleurs des sommets déjà colorés dans l'espoir d'éviter l'emploi de la nouvelle couleur. Réf [8].

A nouveau une des quatre méthodes de coloration séquentielle peut être utilisée comme support de cette méthode. On distinguera LFI, SLI, VPI, NELEI.

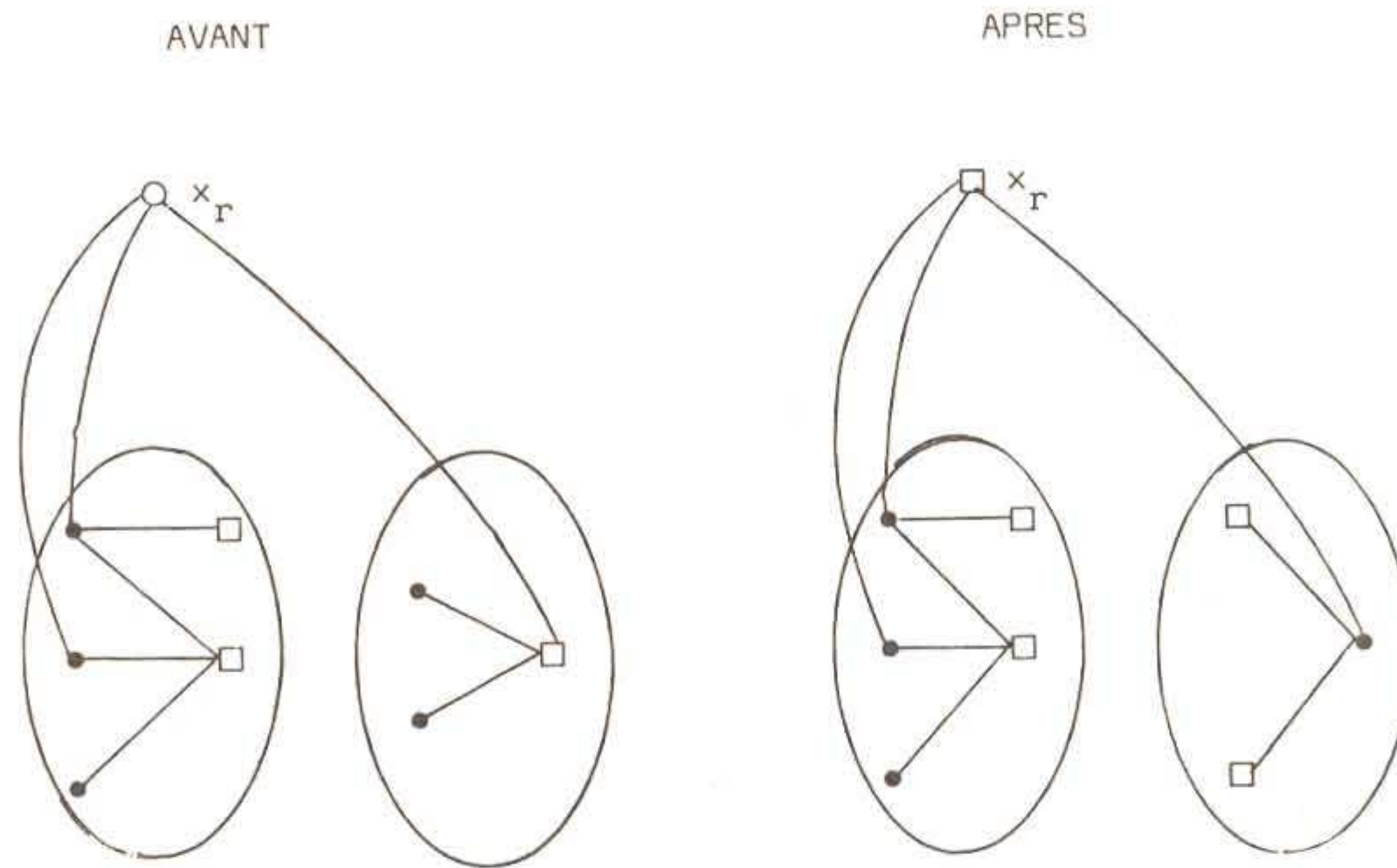
##### Principe

Soit  $\langle G_i \cup G_j \rangle$  le sous-graphe de  $G$  comportant seulement les sommets de couleur  $j$  (notés  $\square$ ) et de couleur  $i$  (notés  $\bullet$ ).

Soit  $x_r$  le sommet à colorer ne pouvant prendre ni la couleur  $i$  ni la couleur  $j$ .

Décomposons  $\langle G_i \cup G_j \rangle$  en composantes connexes.

Si  $x_r$  est adjacent à seulement un type de sommets ( $\square$  ou  $\bullet$ ) par composante connexe de  $\langle G_i \cup G_j \rangle$  une permutation est possible ; il suffit d'échanger les couleurs des sommets d'une série de composantes connexes pour que  $x_r$  ne soit plus adjacent qu'à un type de sommets et puisse donc prendre l'autre couleur.



### 1.2.5. Autres méthodes

Parmi les autres méthodes non implémentées citons les travaux de Matula [5], Rendall-Brown [13], Shanno [11], Zehrafi [16], Corneil et Graham [18], Christofides [19] et bien sur le livre de Berge [10].

### 1.3. Résultats numériques

On a comparé les méthodes VP, SL, LF, SAT, SIMG et les versions avec permutations sur la base de matrices d'incidence générées aléatoirement avec pour paramètres un taux de remplissage variant entre 20 et 80 % et une dimension variant entre 20 x 20 et 100 x 100.

Pour chaque méthode, chaque taux et chaque dimension, 25 matrices ont été générées, ce qui donne une idée des performances du programme qui a ainsi résolu  $9 \times 9 \times 25 = 2.025$  cas. Les tableaux suivants donnent le nombre de couleurs moyen des différentes méthodes pour les cas de 40 et 100 sommets.

	20 %	40 %	60 %	80 %		20 %	40 %	60 %	80 %
VP	6.01	8.96	12.07	17.42	VP	11.27	17.06	23.80	36.41
VPI	5.41	7.94	11.22	15.98	VPI	10.15	15.82	22.44	33.30
LF	5.68	8.60	12.20	16.92	LF	9.68	16.24	24.04	34.56
LFI	5.08	7.72	11.08	15.56	LFI	8.76	14.84	21.96	32.00
SL	5.44	8.56	12.20	17.08	SL	9.72	16.44	24.04	35.96
SLI	5.04	7.72	10.88	15.48	SLI	8.72	14.88	22.18	32.20
SAT	5.04	7.80	11.00	15.52	SAT	8.76	14.84	21.96	32.96
SIMG	5.80	8.52	11.68	16.56	SIMG	10.56	16.64	23.28	33.28

Nous ne reproduirons pas ici tous les tableaux de résultats que l'on pourra trouver dans [20] ; tirons seulement quelques conclusions :

- la méthode de permutation apporte un gain moyen de l'ordre de 8 %,
- la méthode VP est en général la moins performante,
- la méthode SIMG est peu performante pour de faibles taux de remplissage mais devient compétitive lorsque le taux croît,
- pour la plupart des taux et des dimensions les méthodes SLI, LFI et SAT sont les meilleures dans l'ordre avec de faibles écarts.

## 2. LE MODELE DE GRAPHE ET LES CONTRAINTES DIRECTES

### 2.1. Type de contraintes retenues

Par contraintes directes on entend une série de propriétés que l'on impose (souhaite imposer) à la fonction de coloration  $f$ . Celles-ci découlent de considérations pédagogiques, personnelles ou d'intendance. Elles sont obtenues au moyen d'une enquête auprès des enseignants et au secrétariat des départements. Nous avons retenu les suivantes :

- a) congé de la classe  $c_k$  à l'heure  $h_r$  :  $\forall x : c_k \in x$  on a  $f(x) \neq h_r$
- b) indisponibilité du professeur  $p_k$  à l'heure  $h_r$  :  $\forall x : p_k \in x$  on a  $f(x) \neq h_r$

- c) entretien du local  $l_k$  à l'heure  $h_r$  :  $\forall x : l_k \in x$  on a  $f(x) \neq h_r$ .
- d) professeur ne désirant pas deux cours par jour.
- e) professeur ne désirant pas deux cours consécutifs.
- f) cours  $c_k$  et  $c_l$  nécessairement consécutifs dans l'ordre.
- g) horaire d'autres facultés, déjà calculés, et concernant certains professeurs  $p_j \in P$ , classes  $c_i \in P$  ou locaux  $l_k \in P$ .
- h) cours  $x_k$  fixés d'avance à l'heure  $h_r$  :  $f(x_k) = h_r$ .

Une priorité a été établie sur ces contraintes :

- NIVEAU 1 : h
- NIVEAU 2 : a, b, c et g
- NIVEAU 3 : f
- NIVEAU 4 : d, e

Les contraintes de niveau j ne sont pas respectées si elles sont en contradiction avec une contrainte de niveau inférieur, mais un avertissement est fourni dans ce cas.

## 2.2. La matrice de temps L

La présentation des contraintes du 2.1 se fait au moyen d'une matrice L de dimension  $(\#X, NH)$  où NH est le nombre d'heures de cours sur la période (généralement semaine) considérée.

Si  $L(i,j) = 0$  alors le cours i ne peut se donner à l'heure j. La matrice L est initialisée à 1 puis les contraintes de niveau 1 sont passées en revue avant de démarrer la coloration.

$$f(x) \neq h_r \quad (\Rightarrow) \quad L(x,r) = 0 \quad ; \quad (\text{contraintes a,b,c})$$

$$f(x) = h_r \quad (\Rightarrow) \quad L(y,r) = 0 \quad \text{pour tout } y \cap x \neq \emptyset \quad ; \quad (\text{contrainte h})$$

La contrainte g ne pose pas de problèmes théoriques mais quelques difficultés combinatoires lorsque les bases horaires de l'horaire existant et de celui à calculer sont différentes. Les contraintes d, e, f n'interviennent qu'en cours de coloration.

Par exemple (contrainte e), si on a obtenu

$$f(x) = h_r \text{ avec } p_j \in x \text{ alors pour tout } y \in p_j$$
$$\begin{cases} L(y, r+1) = 0 \\ L(y, r-1) = 0 \end{cases}$$

pour autant que les heures r et r+1 (resp. r et r-1) appartiennent à la même demi-journée.

Bien entendu, la matrice L est mise à jour également par les contraintes indirectes (d'incompatibilités) en cours de coloration.

$$f(x) = h_r \rightarrow L(y, r) = 0 \text{ pour tout } y \cap x \neq \emptyset$$

### 2.3. Quelques méthodes de coloration supplémentaires

La définition de la matrice L suggère l'emploi de méthodes venant s'ajouter à celles du paragraphe 1 qu'il suffit d'adapter. Cette adaptation aux contraintes directes est relativement simple, sauf en ce qui concerne la méthode de permutation. Nous proposons trois nouvelles méthodes.

#### 2.3.1. Méthodes de la possibilité de coloration

Les méthodes se basent sur le rangement des sommets selon la fonction

$$POC(x) = \sum_{j=1}^{NH} L(x, j),$$

représentent le nombre d'heures encore disponibles pour le cours x.

- En ordonnant les sommets une fois pour toutes (avant de commencer la coloration) selon la fonction POC, on obtient la méthode séquentielle POCO.
- Lorsque POC est recalculé après chaque mise à jour de L, il s'agit d'une méthode de saturation, dite méthode par possibilité de coloration effective (EFF).

#### 2.3.2. Méthode de similarité de temps (SIMT)

Comparable à la similarité de graphe mais basée sur L au lieu de M ; elle utilise un indice de similarité

$$T_{ij} = 0 \quad \text{si } i=j \quad \text{ou si } M_{ij} = 1$$

$$= \sum_{k=1}^{NH} L(i,k) \quad L(j,k) \quad \text{sinon.}$$

### 2.3.3. Remarques

Les méthodes de coloration essaient de fournir une coloration utilisant au plus NH couleurs ; si elles échouent, il faut nécessairement relâcher certaines contraintes.

Inversément, comme les méthodes tendent à minimiser le nombre de couleurs utilisées, elles conduisent à des horaires compacts en début de semaine et libèrent parfois le vendredi totalement.

Si cela est jugé néfaste, il suffit par perturbation de libérer une ou deux après-midi pour aérer l'horaire.

Ces méthodes de perturbation font l'objet du paragraphe suivant.

### 2.4. Méthodes de perturbation (PER)

Leur but est de partir d'un horaire calculé et de le modifier le moins possible pour tenir compte de contraintes ou ajoutées non prévues initialement.

#### 2.4.1. Perturbation globale (GLO)

- ordonner les sommets selon une méthode LF, SL, NELE, VP, POCO ;
- donner aux sommets pris dans l'ordre la couleur qu'ils avaient dans l'ancien horaire si c'est possible malgré les nouvelles contraintes ; sinon leur donner la première couleur possible.

#### 2.4.2. Perturbation minimum (MIN)

- considérer l'ensemble des sommets respectant les contraintes nouvelles si on leur conserve leur ancienne couleur.
- colorer l'ensemble des sommets restants selon une des méthodes LF, SL, NELE, VP, POCO.

Si les modifications sont faibles et l'horaire assez "léger", la méthode (MIN) donne d'excellents résultats ; dans le cas contraire, préférer (GLO).

## 2.5. Résultats numériques

Les méthodes VP, SL, LF, POCO, SAT, EFF, SIMG, SIMT et les versions avec permutations ont été comparées sur base de matrices d'incidence et matrices de temps générées aléatoirement avec pour paramètres des taux de remplissage de 20 % à 80 %, respectivement de 60 %, 75 % et 90 %, et une dimension variant entre 20 x 20 et 100 x 100 (par pas de 10 x 10) ce qui permet de juger les performances sur un ensemble de  $9 \times 5 \times 3 \times 25 = 3.375$  cas différents.

Le tableau ci-après résume les performances des différentes méthodes pour une dimension 60 x 60.

Nous ne reproduirons pas tous les résultats que l'on pourra d'ailleurs trouver dans [20]. Tirons cependant quelques conclusions :

- Les méthodes utilisant la maîtrise de temps (POCO, POCOII, EFF et SIMT) se montrent particulièrement efficaces pour des taux de remplissage assez bas du graphe et de la matrice de temps.
- La méthode SIMT est en général la moins performante, surtout pour des graphes à taux de remplissage élevé.
- Les méthodes de permutation et de saturation sont une fois de plus très performantes.

## 3. LE PROGRAMME ET SON UTILISATION

### 3.1. Description

Le programme a été écrit en FORTRAN IV sur matériel DIGITAL EQUIPMENT CORPORATION 2060 ; son adaptation sur micro-ordinateur est à l'étude.

Sa structure, de type arborescent avec menu, permet son utilisation par le personnel non spécialisé en informatique.

Il comporte quatre parties :

- documentation : DOC
- gestion de fichiers de données : GFI
- calcul de l'horaire : EXE
- impression de résultats : IMP

Donnons une brève description de chacune d'elles.

Graphique à 60 sommets : nombre de couleurs

	20%			30%			40%			50%			60%			70%			80%		
	60%	75%	90%	60%	75%	90%	60%	75%	90%	60%	75%	90%	60%	75%	90%	60%	75%	90%	60%	75%	90%
	VP	10.52	8.96	7.48	13.40	11.08	10.00	15.40	13.48	12.28	16.96	15.68	14.20	20.52	18.60	16.92	24.32	21.88	20.08	28.48	25.20
VPI	9.96	8.32	7.04	11.84	10.40	9.04	13.92	12.56	11.20	16.40	14.36	13.32	19.04	16.68	15.64	22.36	19.88	18.56	25.88	23.40	20.40
LF	10.96	9.04	7.76	13.20	11.12	10.00	15.12	13.24	12.04	17.56	15.60	14.04	20.08	18.32	16.80	24.52	21.44	20.16	27.40	25.24	23.56
LFI	10.12	8.56	6.88	12.28	10.28	8.88	14.40	12.36	11.12	16.36	14.56	13.04	18.69	17.08	15.64	22.64	19.76	18.40	25.64	23.56	22.12
SL	11.20	9.08	7.84	12.20	10.96	9.76	15.68	13.64	12.12	17.00	16.04	14.16	20.00	18.28	17.60	23.96	21.76	20.44	28.20	25.52	24.48
POCO	10.52	9.36	8.04	12.64	11.28	10.40	14.88	13.96	13.40	17.36	16.44	15.24	20.72	18.96	17.88	24.28	22.36	20.36	28.52	26.08	25.40
POCOI	10.12	8.40	7.40	11.76	10.36	9.28	14.16	12.84	11.68	16.48	14.68	13.92	19.12	17.24	16.52	22.48	20.24	19.12	25.88	24.00	22.88
SAT	10.24	8.60	7.32	12.36	10.72	9.16	14.72	12.84	11.32	16.88	15.00	13.64	19.24	17.98	16.04	23.28	20.76	19.28	26.92	24.80	23.12
EFF	10.60	9.12	7.92	12.44	10.96	9.80	14.80	13.20	12.16	17.56	15.80	14.24	20.00	18.20	17.16	23.88	21.44	20.24	27.16	26.00	24.04
SIMG	13.48	10.52	8.32	15.80	11.92	10.20	16.88	13.76	12.60	18.88	16.80	14.44	21.24	18.36	16.76	24.60	20.80	19.16	27.60	24.16	22.68
SIMT	12.76	10.88	8.92	15.96	13.40	11.24	18.76	15.52	13.68	21.52	18.60	16.24	24.48	20.50	18.88	26.16	23.64	21.64	29.80	27.20	26.00
SLI	10.24	8.28	6.88	11.56	10.16	8.88	13.92	12.24	11.08	16.32	14.40	12.88	18.68	16.60	15.44	21.72	19.68	18.52	26.04	23.28	22.08

: temps CPU (calcul seul, DEC 2060) (en secondes)

VP : 0.47 | LF : 0.26 | SL : 1.27 | POCO : 0.25 | SAT : 1.71 | SINT : 0.69  
 VPI : 1.21 | LFI : 0.84 | SLI : 2.10 | POCOI : 0.63 | EFF : 1.03 | SIMT : 1.02

DOC

Permet l'accès "on line" à un manuel utilisateur synthétique.

GFI

Comporte la création, modification, destruction, remplacement de cours, professeurs, locaux, classes et contraintes directes de divers types.

Il gère :

- un fichier de cours à donner (intitulé, composants),
- un fichier classes, professeurs, locaux,
- un fichier par type de contraintes directes (a, ... g) (cfr. 2.1).

Ces fichiers sont consultables en mode conversationnel afin de contrôler aisément la validité de leur contenu.

Par exemple, une commande permet d'afficher la liste des indisponibilités d'un professeur donné ; une autre le cours obligatoirement consécutif à un autre donné, etc.

EXE

- détermination des fichiers contenant les contraintes directes et les cours préalablement créés par GFI.
- choix d'une méthode de coloration.
- calcul de l'horaire.
- affichage provisoire éventuel.
- mise sur fichier éventuel pour exploitation par IMP.

IMP

Cinq types de sorties ont été prévus :

- horaire total (synoptique toutes classes, heure par heure),
- horaire par classe,
- horaire par professeur,
- horaire par local,
- horaire de type programme de cours (par classe, mais rangé dans l'ordre des cours conforme au programme).

### 3.2. Premières expériences

Le programme a été expérimenté à ce jour aux Facultés universitaires N.-D. de la Paix à Namur par la Faculté des Sciences (candidatures uniquement) <sup>(\*)</sup> Le résultat est très satisfaisant car l'horaire obtenu respecte mieux les desiderata des professeurs que celui actuellement en vigueur et mis au point annuellement. Ce résultat a été obtenu grâce aux facilités de modification ponctuelle d'un horaire donné (méthodes de perturbation) qui permet de façonner plusieurs horaires convergeant progressivement vers un "idéal".

Une autre application fut l'élaboration d'un projet d'horaire pour les licences et maîtrises en informatique suite à un projet de réforme fondamentale de leur programme. Dans ce cas, partant de rien, un résultat intéressant fut fourni aux demandeurs après un jour et demi (codage de données et discussion avec les responsables compris). Certes, le nombre de classes et de professeurs était dans ce cas assez limité, mais la multiplicité des cours à option prévus, la totale liberté de choix souhaitée pour les étudiants, la découpe horaire variable selon les classes et les semestres (à cause de semaines libérées pour certains travaux personnels) ainsi que la répartition encore à définir des cours selon les semestres, compliquaient fortement le problème.

Ces deux applications précisent bien l'optique d'utilisation du produit, qui est celle d'un outil de travail pour le responsable des horaires évitant erreurs et permettant d'explorer nombre de solutions différentes parmi lesquelles il fera son choix ; il ne s'agit nullement d'un automate chargé d'apporter la solution "idéale" à un utilisateur anonyme et sans expérience (chose que nous croyons d'ailleurs peu réaliste, voire impossible).

### 3.3. Application à l'enseignement secondaire

Dans sa configuration actuelle, le programme permet de traiter des graphes de 500 sommets (heures de cours). Il semblerait qu'un établissement d'enseignement moyen dispense environ 1.000 heures de cours, différentes options comprises. Dans ce cas, une découpe cycle inférieur - cycle supérieur s'avère nécessaire. Si elle demeure insuffisante, on peut travailler année par année en utilisant la possibilité de tenir compte des horaires déjà calculés agissant

---

(\*) Soit 200 heures de cours, c'est-à-dire 200 sommets.

comme contraintes.

Il faut cependant noter que la limite de 500 est essentiellement liée au matériel dont nous disposons et non au programme lui-même.

La complexité théorique en  $n^2$  des algorithmes permettant d'envisager le traitement de problèmes de cette taille dans un temps CPU raisonnable (cfr tableau 3.1.)

#### 4. REFERENCES

- [1] R.L. BROOKS, "On colouring the Nodes of a Network", Proceedings Cambridge Phils. Society 37, 194-197 (1941).
- [2] H.S. WILF, "The Eigenvalues of a Graph and its Chromatic Number", J. London Math. Society, to appear.
- [3] D.J.A. WELSH & N.B. POWELL, "An upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems", Computing Journal, 10, 85-86 (1967).
- [4] J.A. BONDY, "Bounds for the Chromatic Number of a Graph", Journal of Combinatorial Theory 7, 96-98 (1969).
- [5] D.W. MATULA, "A Mini-Max Theorem for Graphs with Application to Graph Coloring", SIAM Review 10, 481-482 (1968).
- [6] D.W. MATULA, "k-Components, Clusters and Slicings in Graphs", SIAM Journal Appl. Math. 22, N° 3, 459-480 (1972).
- [7] G. SZEKERES & H.S. WILF, "An Inequality for the Chromatic Number of a Graph", Journal of Combinatorial Theory 4, 1-3 (1968).
- [8] D.W. MATULA, G. MARBLE & J.D. ISAACSON, "Graph Coloring Algorithms", Graph Theory and Computing, Academic Press, New York, 109-122 (1972).
- [9] D. ORE, "The Form Color Problem", Academic Press, New York (1972).
- [10] C. BERGE, "Graphes et hypergraphes", Dunod Université (1970).
- [11] D.F. SHANNO, "New Methods to color the Vertices of a Graph", Communications of the ACM 22, N° 4, 251-256 (1979).

- [ 12 ] D.C. WOOD, "A Technique for coloring a Graph applicable to Large Scale Timetabling Problems", Computer Journal 12, 317-319 (1969).
- [ 13 ] J. RENDALL-BROWN, "Chromatic Scheduling and the Chromatic Number Problems", Management Science 19, 4, Part I, 456-463 (1972).
- [ 14 ] J.A. FORMBY, "A Computer Procedure for bounding the Chromatic Number of a Graph", University of Birmingham (England), Combinational Mathematics and its Applications, Welsh, Academic Press (1971).
- [ 15 ] M. BEHAZED, "The Total Chromatic Number of a Graph : a Survey", National University of Iran, Combinational Mathematics and its Applications, Welsh, Academic Press (1971).
- [ 16 ] A. ZEHRANI, "Un algorithme de coloration", Cahiers du Centre d'Etudes de Recherche Opérationnelle, Bruxelles, vol. 17, 395-398 (1975).
- [ 17 ] K. MAGHOUT, "Applications de l'algèbre de Boole à la théorie de graphes et aux programmes linéaires et quadratiques", Cahiers du Centre d'Etudes de Recherche Opérationnelle, Bruxelles, vol 5, 193... (1963).
- [ 18 ] D.G. CORNEIL & B. GRAHAM, "An Algorithm for determining the Chromatic Number of a Graph", SIAM Journal of Computing, Vol. 2, N° 4, 311-318 (1973).
- [ 19 ] N. CHRISTOFIDES, "An Algorithm for the Chromatic Number of a Graph", The Computer Journal 14, N° 1, 38-39 (1970).
- [ 20 ] O. HENKES, "Etablissement d'horaires de cours par coloration de graphes", Mémoire à présenter pour l'obtention du titre de Licencié et Maître en Informatique, F.N.D.P., Namur, juin 1982.

°  
° °