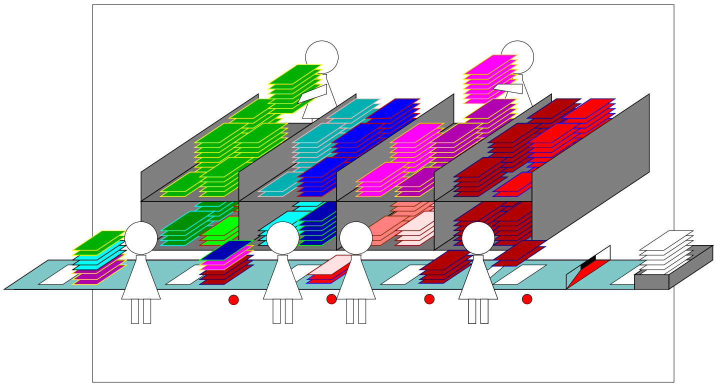
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Projet de semestre

Hiver 2002-2003

Préparation d'une distribution de publications



Daniela BRONNER

-MA- 3 ème année

Simona CEREGHETTI

-MA- 4 ème année

Responsables :

Prof. Dominique DE WERRA

Ivo BLÖCHLIGER

David SCHINDL

Table des matières

Introduction

Une maison de presse s'occupe de la distribution de journaux et de magasins dans tous les points de vente de la Suisse romande. Chaque jour, dans le centre de distribution de cette entreprise, les titres qui seront distribués le lendemain sont choisis parmi tous les titres présents dans l'entrepôt. Ce choix se fait selon des contraintes imposées par les éditeurs et selon des contraintes de précedence. Les journaux ainsi choisis sont triés en fonction de contraintes bien précises et divisés en douze groupes. Chaque groupe est affecté à un *module*. Un module est une place de travail, occupée par un opérateur ou une opératrice (figure 1 : opérateurs 1, 2, 3 et 4), et se compose de deux tables superposées où sont placées les piles de journaux. Sur ces deux tables il est possible de poser jusqu'à quatorze tas (dans la figure 1 seuls quatre tas par module sont représentés). Ceux-ci sont continûment réapprovisionnés par d'autres employés (figure 1 : employés 1 et 2).

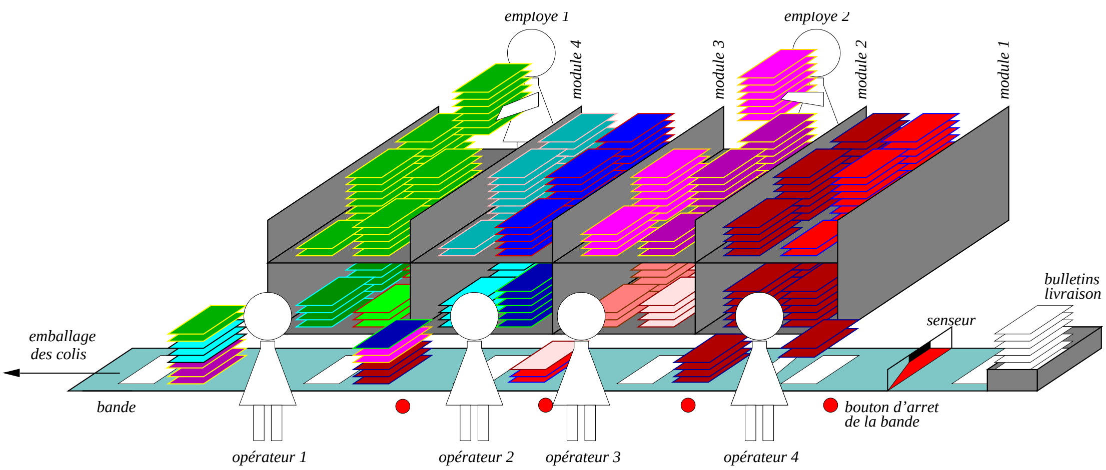


FIG. 1 –

Devant les modules se trouve une bande sur laquelle sont posées les commandes des clients. Nous appellerons une feuille de commande un *bulletin de livraison* (voir figure 2).

La bande se déplace devant les opérateurs qui lisent sur les bulletins de livraison le nombre de copies requises pour chaque titre de journal et posent cette quantité à côté du bulletin de livraison en question. Pour chaque type de journal requis par un bulletin de livraison un opérateur doit faire un geste. Nous appellerons ce geste un *pick*. À la fin de la chaîne, lorsque le bulletin de livraison est passé devant tous les modules, la pile de journaux est emballée et le paquet envoyé au client. Chaque client peut recevoir plusieurs paquets, donc pour chaque client il existe un nombre plus ou moins grand de bulletins de livraison qui défileront sur la bande. Ainsi la répartition des journaux sur les bullettins peut être faite de plusieurs façons.

	nombre	*titres*
module A	*7*	*journal 1*
	3	*journal 2*
	2	*journal 3*
module B	*17*	*journal 4*
module C	*4*	*journal 5*
	7	*journal 6*
	16	*journal 7*
module D	*4*	*journal 8*
	4	*journal 9*

FIG. 2 – bulletin de livraison

Au début de la bande un senseur détecte le passage des bulletins de livraison, de façon à régler la fréquence de passage des bulletins sur la bande. Cette fréquence peut être modifiée selon les besoins.
Il arrive que les opérateurs n'aient pas le temps d'effectuer tous les picks requis par les bulletins de livraison et soient ainsi obligés d'arrêter la bande. Il est clair que si l'un des opérateurs arrête la bande durant une minute, tous les opérateurs des autres modules ne peuvent continuer leur travail pendant cette minute.

Actuellement l'affectation des titres sur les modules se fait grâce à l'expérience des employés. Par contre la répartition des titres sur le bulletin de livraison est donnée par un algorithme. Celui-ci fournit une solution qui, selon les responsables de l'entreprise, peut être améliorée par une modélisation soigneuse du problème et de ses contraintes.
Notre projet consiste en l'amélioration de ce système, en proposant une modélisation qui puisse fournir une répartition optimale des titres de journaux sur les bulletins de livraison et sur les modules, de façon à ce que le temps d'arrêt de la bande soit minimal, tout en respectant des contraintes de poids et de hauteur des colis, ainsi que d'autres contraintes qui seront explicitées par la suite.

Notre travail sera partagé en deux parties : la première consiste en l'étude du problème de la répartition des journaux sur les modules, alors que la deuxième consiste en l'étude du problème de la fabrication des bulletins de livraison. Dans chaque partie nous allons proposer une modélisation mathématique, une modélisation en termes de graphes et pour finir nous allons proposer des algorithmes de résolution pour les deux problèmes.

Chapitre 1

Données

Dans ce premier chapitre nous expliciterons les ensembles décrivant les données fournies par les responsables de l'entreprise. Ces données étant des données générales, nous nous servirons de ces définitions dans le traitement des deux problèmes.

Nous dirons que :

N	est le nombre de titres de journaux (N vaut environ 115) ;
N_k	est le nombre maximal de positions du haut ou du bas par module (=7) ;
N_m	est le nombre de modules (=12) ;
p_j	est le poids d'un journal de titre j ;
f_j	est le format d'un journal de titre j ;
h_j	est la hauteur d'un journal de titre j ;
p_{max}	est le poids maximal d'un colis ;
h_{max}	est la hauteur maximale d'un colis ;
k_j	est le nombre de positions nécessaires pour le titre j ;
c_{jt}	est le nombre de journaux de titre j commandés par le client t.

$\mathcal{J}$	est l'ensemble des titres de journaux j ;
$\mathcal{M}$	est l'ensemble des modules m ;
$\mathcal{B}$	est l'ensemble des bulletins de livraison b ;
$\mathcal{J}_j$	est l'ensemble des titres de journaux qui ne peuvent pas être placés sur le même module que le journal j ;
$\mathcal{J}_{A5}$	est l'ensemble des titres de journaux de format A5 ;
$\mathcal{J}_{plier}$	est l'ensemble des titres de journaux à plier ;
$\mathcal{J}_s$	est l'ensemble des titres de journaux non-sécables ;
$\mathcal{J}_t$	est l'ensemble des titres de journaux requis par le client t.

Chapitre 2

Variables

Dans ce chapitre nous allons décrire les variables que nous avons choisies pour décrire les deux problèmes.

2.1 Variables pour le problème de la répartition des titres de journaux sur les modules

Dans notre modélisation nous avons choisi de chercher une affectation des titres de journaux sur les modules, en tenant compte des modules (de 1 jusqu'à N_m) avec la variable x_{jm} et en tenant compte des positions sur les modules (pour le module 1 de 1 jusqu'à 7, pour le module 2 de 1 jusqu'à 7, et ainsi de suite). Nous avons choisi cette numérotation car il nous était nécessaire de faire la différence entre la table du haut et celle du bas pour chaque module. Nous avons décidé de faire ceci au moyen de deux variables, y_{jmi} et z_{jmi} définies pour tout journal j, pour tout module m et pour toute position i.

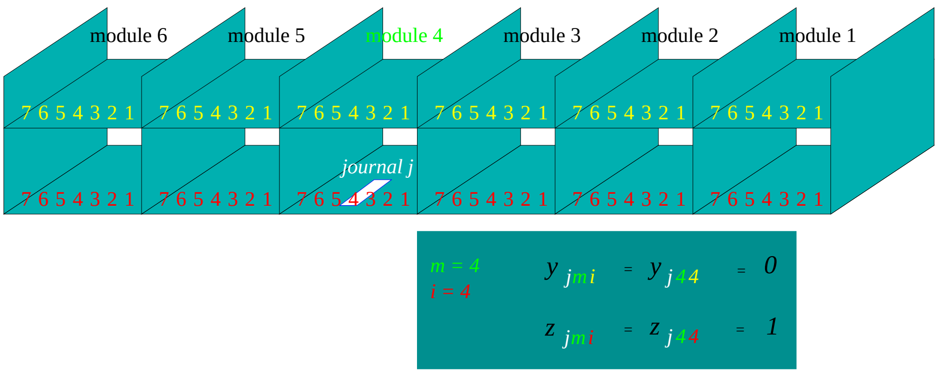


FIG. 2.1 – représentation des indices

Les variables y_{jmi} et z_{jmi} définissent déjà toutes les positions sur tous les modules, mais nous introduisons quand même la variable x_{jm} car celle-ci nous permet de décrire certaines contraintes

par une seule équation plutôt que par deux.
Ceci afin de simplifier la forme mathématique des contraintes et de rendre le texte plus lisible.

$$x_{jm} = \begin{cases} 1 & \text{si le titre de journal } j \text{ est sur le module } m, \\ 0 & \text{sinon ;} \end{cases}$$

$$y_{jmi} = \begin{cases} 1 & \text{si le titre de journal } j \text{ est sur le module } m \text{ à la position } i \text{ du haut,} \\ 0 & \text{sinon ;} \end{cases}$$

$$z_{jmi} = \begin{cases} 1 & \text{si le titre de journal } j \text{ est sur le module } m \text{ à la position } i \text{ du bas,} \\ 0 & \text{sinon.} \end{cases}$$

2.2 Variables pour le problème de la répartition des journaux sur les bulletins de livraison

Une fois le problème de la répartition des titres de journaux sur les modules résolu, nous connaissons à quel module est affecté tout titre de journal.
Nous disposons ainsi d'une nouvelle donnée :

$\mathcal{J}_m$ est l'ensemble des titres de journaux affectés au module m.

Nous définissons encore les deux variables suivantes :

n_{jb} est le nombre de journaux de type j sur le bulletin de livraison b ;

$$w_{jb} = \begin{cases} 1 & \text{si le titre de journal } j \text{ est sur le bulletin de livraison } b, \\ 0 & \text{sinon.} \end{cases}$$

Cette deuxième variable est donnée implicitement par la première, mais nous l'introduisons quand même afin de simplifier les notations.

Chapitre 3

Contraintes

Dans ce chapitre nous allons décrire sous forme mathématique toutes les contraintes auxquelles les deux problèmes sont liés.
Toutes ces contraintes ont été imposées par l'entreprise (voir feuille annexe).

3.1 Contraintes pour la répartition des titres de journaux sur les modules

Les contraintes pour le problème de la répartition des titres de journaux sur les modules peuvent être classifiées en cinq catégories principales :

i) contraintes générales qui sont intrinsèques à la structure du problème et non imposées par l'entreprise ;
ii) contraintes « d'offres cadeaux » pour des journaux qui contiennent des objets plus ou moins épais ;
iii) contraintes sur le poids des journaux ;
iv) contraintes de qualité que nous définirons par la suite ;
v) contraintes sur la quantité d'exemplaires d'un titre de journal.

Voici donc la liste détaillée de toutes ces contraintes et de leur expression sous forme mathématique.

3.1.1 Contraintes générales

1. Chaque type j de journal doit être affecté exactement à un module m :

$$\sum_{m=1}^{N_m} x_{jm} = 1 \qquad \forall\, j = 1, \ldots, N.$$

2. Sur chaque module il est possible d'empiler des journaux sur au maximum N_k positions pour la table du haut :

$$\sum_{j \in \mathcal{J}} y_{jmi} \leq 1 \qquad \begin{array}{l} \forall\ m = 1, \ldots, N_m \\ \forall\ i = 1, \ldots, N_k. \end{array}$$

3. De même, pour la table du bas nous aurons :

$$\sum_{j \in \mathcal{J}} z_{jmi} \leq 1 \qquad \begin{array}{l} \forall\ m = 1, \ldots, N_m \\ \forall\ i = 1, \ldots, N_k. \end{array}$$

4. Contrainte de compatibilité : il y a des titres de journaux qui ne peuvent pas être affectés à un même module, ce afin de ne pas confondre les opérateurs. Par exemple, si l'on a deux journaux avec titres respectifs « mots-croisés » et « mots-croisés en compagnie », il est souhaitable que ces deux journaux ne se trouvent pas sur le même module :

$$x_{jm} + \sum_{j \in \mathcal{J}_j} x_{jm} \leq 1 \qquad \begin{array}{l} \forall\ m = 1, \ldots, N_m \\ \forall\ j = 1, \ldots, N \end{array}$$

3.1.2 Contraintes des « offres cadeaux »

1. Si le titre de journal j contient un *CD plat*, il peut uniquement être affecté aux modules 6 à 12 :

$$x_{jm} = 0 \qquad \forall\ m = 1, \ldots, 5$$

2. Si le titre de journal j contient un *CD boîte*, une *cassette audio* ou un *gadget plat*, il peut uniquement être affecté aux modules 10 à 12 :

$$x_{jm} = 0 \qquad \forall\ m = 1, \ldots, 9$$

3. Si le titre de journal j contient un *DVD boîte* ou bien une *cassette vidéo*, il peut uniquement être affecté aux modules 10 à 12 sur les tables du haut :

$$\begin{array}{ll} y_{jmi} = 0 & \forall\ m = 1, \ldots, 9 \\ z_{jmi} = 0 & \forall\ m = 1, \ldots, N_m \end{array}$$

4. Si le titre de journal j contient une *boîte* ou bien un *gros gadget*, il peut uniquement être affecté au module 12 sur la table du haut :

$$\begin{array}{ll} y_{jmi} = 0 & \text{pour } m = 1, \ldots, 11 \\ z_{jmi} = 0 & \forall\ m = 1, \ldots, N_m \end{array}$$

5. Si le titre de journal j contient une *boîte*, il peut uniquement être affecté aux positions tout à gauche en dernier :

$$y_{jmi} = 0 \qquad \begin{array}{l} \forall\ i = 1, \ldots, 6 \\ \forall\ m = 1, \ldots, N_m \end{array}$$

6. Si le titre de journal j contient un *gadget épais*, il peut uniquement être affecté aux modules 11 et 12 sur la table du haut :

$$\begin{array}{ll} y_{jmi} = 0 & \forall\ m = 1, \ldots, 10 \\ z_{jmi} = 0 & \forall\ m = 1, \ldots, N_m \end{array}$$

3.1.3 Contraintes de format

1. Si le format du titre de journal j est un format A3, ce titre peut uniquement être affecté au module 1 sur la table du bas :

$$\begin{array}{ll} x_{jm} = 1 & \text{pour } m = 1 \\ y_{jmi} = 0 & \forall\ m = 1, \ldots, N_m \end{array}$$

2. Si le format du titre de journal j est un format A3, ce titre peut uniquement être affecté à la première position :

$$\begin{array}{ll} z_{jmi} = 0 & \forall\ i = 2, \ldots, N_k \\ & \forall\ m = 1, \ldots, N_m \end{array}$$

3. Si le format du titre de journal j est un format A4+, ce titre peut uniquement être affecté aux modules 1 à 5 sur la table du haut :

$$\begin{array}{ll} y_{jmi} = 0 & \forall\ m = 6, \ldots, N_m \\ z_{jmi} = 0 & \forall\ m = 1, \ldots, N_m \end{array}$$

4. Si le format du titre de journal j est un format A4+, ce titre peut uniquement être affecté à la première position :

$$\begin{array}{ll} y_{jmi} = 0 & \forall\ i = 2, \ldots, N_k \\ & \forall\ m = 1, \ldots, N_m \end{array}$$

5. Si le format du titre de journal j est un format A5, ce titre doit être affecté aux tables du haut :

$$z_{jmi} = 0 \qquad \forall\ m = 1, \ldots, N_m$$

   De plus sur un module nous pouvons avoir au maximum trois titres de ce type de journal :

$$\sum_{j \in \mathcal{J}_{A5}} x_{jm} \leq 3 \qquad \forall\ m = 1, \ldots, N_m$$

6. Si le format du titre de journal j est un format à plier, ce titre peut uniquement être affecté aux modules 6 à 12 sur la table du haut :

$$\begin{array}{ll} y_{jmi} = 0 & \forall\ m = 1, \ldots, 5 \\ z_{jmi} = 0 & \forall\ m = 1, \ldots, N_m \end{array}$$

   De plus sur un module nous pouvons avoir au maximum un titre de ce type de journal :

$$\sum_{j \in \mathcal{J}_{plier}} x_{jm} \leq 1 \qquad \forall\ m = 1, \ldots, N_m$$

3.1.4 Contraintes de poids

1. Si un titre de journal j à un poids inférieur à 100 grammes, il doit être affecté aux tables du haut :

$$z_{jmi} = 0 \qquad \forall\ m = 1, \ldots, N_m$$

2. Si un titre de journal j à un poids supérieur à 800 grammes, il peut uniquement être affecté aux modules 1 à 4 :

$$x_{jm} = 0 \qquad \forall\ m = 5, \ldots, N_m$$

3.1.5 Contraintes de qualité

1. Si le titre de journal j est bombé, il peut uniquement être affecté aux modules 6 à 12 :

$$x_{jm} = 0 \qquad \forall\, m = 1, \ldots, 5$$

3.1.6 Contraintes de quantité

1. Si le titre de journal j a un nombre d'exemplaires inférieur à 1000, ce titre peut uniquement être affecté aux tables du haut :

$$z_{jmi} = 0 \qquad \forall\, m = 1, \ldots, N_m$$

2. Si le titre de journal j a un nombre d'exemplaires entre 3000 et 8000, ce titre peut uniquement être affecté aux tables du bas :

$$y_{jmi} = 0 \qquad \forall\, m = 1, \ldots, N_m$$

3. Si le titre de journal j a un nombre d'exemplaires supérieur à 8000, ce titre peut uniquement être affecté aux modules 1 à 8 sur la table du bas :

$$\begin{aligned} x_{jm} &= 0 \qquad \forall\, m = 9, \ldots, N_m \\ y_{jmi} &= 0 \qquad \forall\, m = 1, \ldots, N_m \end{aligned}$$

4. Si le titre de journal j a un nombre d'exemplaires inférieur à 250, ce titre peut uniquement être affecté aux positions de gauche :

$$\begin{aligned} y_{jmi} = 0 \qquad &\forall\, i = 1, \ldots, 5 \\ &\forall\, m = 1, \ldots, N_m \end{aligned}$$

5. Si le titre de journal j a un nombre d'exemplaires entre 250 et 1000, ce titre peut uniquement être affecté aux positions de droite :

$$\begin{aligned} y_{jmi} = 0 \qquad &\forall\, i = 3, \ldots, N_k \\ &\forall\, m = 1, \ldots, N_m \end{aligned}$$

6. Si le titre de journal j a un nombre d'exemplaires entre 1000 et 3000, ce titre peut uniquement être affecté aux positions du milieu :

$$\begin{aligned} y_{jmi} = 0 \qquad &\forall\, i = 1, 2, 5, 7 \\ &\forall\, m = 1, \ldots, N_m \\ z_{jmi} = 0 \qquad &\forall\, i = 1, 2, 5, 7 \\ &\forall\, m = 1, \ldots, N_m \end{aligned}$$

7. Si le titre de journal j a un nombre d'exemplaires supérieur à 3000, ce titre peut uniquement être affecté aux positions de droite :

$$\begin{aligned} z_{jmi} = 0 \qquad &\forall\, i = 3, \ldots, N_k \\ &\forall\, m = 1, \ldots, N_m \end{aligned}$$

3.2 Contraintes pour la répartition des journaux sur les bulletins de livraison

Nous identifions chaque colis à son bulletin de livraison b et nous rappelons que n_{jb} est le nombre de journaux de titre j affectés au module m :

1. Un colis doit peser au maximum p_{max} :

$$\sum_{j \in b} p_j \cdot n_{jb} \leq p_{max} \qquad \forall \ b \in \mathcal{B}$$

2. Un colis doit avoir au maximum une hauteur de h_{max} :

$$\sum_{j \in b} h_j \cdot n_{jb} \leq h_{max} \qquad \forall \ b \in \mathcal{B}$$

3. Si le titre de journal j est non-sécable, il doit être mis sur le même bulletin de livraison c'est-à-dire si un client t commande n journaux de titre j, alors ces n journaux doivent être mis dans le même colis (donc ils doivent se trouver sur le même bulletin de livraison) :

$$\exists \ b^* \text{ tel que } \forall \ j \ \text{ non-sécable} \ \left\{ \begin{array}{lcl} n_{jb^*} & = & c_{jt} \\ n_{jb} & = & 0 \qquad \forall b \neq b^* \end{array} \right.$$

Chapitre 4

Fonction objectif

L'entreprise voulant réduire le temps de travail afin de pouvoir commencer plus tôt la distribution de journaux aux clients et afin de ne pas devoir rattraper le travail non accompli pendant la journée le matin suivant, nous cherchons à rendre le travail plus fluide de façon à ce que les opérateurs soient obligés d'arrêter la bande le moins possible.
Les objectifs proposés par l'entreprise sont donc les suivants :

1. Minimiser le nombre d'arrêts de la bande ;
2. Minimiser le temps total durant lequel la bande est immobile.

Il est évident que le nombre d'arrêts de la bande ne peut pas être prévu. Par contre, nous pourrions optimiser ces objectifs en trouvant une solution pour la fonction objectif suivante :

3. Minimiser le nombre de picks (nombre de types de journal) requis par le bulletin de livraison b pour l'opérateur du module m.

Toutefois, avec cette fonction objectif, nous pourrions avoir une solution où des opérateurs doivent faire deux picks par bulletin de livraison pendant un certain temps et zéro picks pour le temps restant. Ceci ne serait pas optimal car les opérateurs en question pourraient devoir souvent arrêter la bande pendant le premier intervalle de temps, alors qu'ils seraient inactifs pendant le deuxième intervalle de temps.
Pour améliorer cette situation, nous proposons l'objectif suivant :

4. Pour chaque module, nous demandons que sur deux bulletins de livraison consécutifs il y ait au maximum deux picks. Ainsi chaque opérateur aura au plus un pick par bulletin de livraison à effectuer :
$$\sum_{j \in \mathcal{J}_m} w_{jb_i} + w_{jb_{i+1}} \leq 2 \qquad \forall\ b_i \in \mathcal{B},$$
où b_{i+1} est le bulletin de livraison qui suit immédiatement le bulletin b_i

Il est clair que cet objectif représente la situation idéale, mais ce n'est pas une fonction objectif proprement dite. Il s'agit plutôt d'une contrainte.

Aussi considèrerons-nous comme véritable fonction objectif la fonction suivante :

5. Pour chaque module, nous cherchons une solution minimisant le nombre k de picks requis par deux bulletins de livraison consécutifs.
   Ainsi chaque opérateur aura en moyenne au plus $\frac{k}{2}$ gestes par bulletin de livraison à faire. Ceci s'exprime par :

$$\min \sum_{j \in \mathcal{J}_m} w_{jb_i} + w_{jb_{i+1}} \qquad \forall\ b_i \in \mathcal{B},$$

   où b_{i+1} est le bulletin de livraison qui suit immédiatement le bulletin b_i

Celle-ci sera donc la fonction objectif pour le problème de la répartition des journaux sur les bulletins de livraison, lorsque pour le problème de la répartition des titres de journaux sur les modules, nous nous « contenterons » de chercher une solution qui soit compatible avec toutes les contraintes.

Chapitre 5

Quelques rappels sur la théorie des graphes

5.1 Quelques définitions

Définition 5.1.1 Un *graphe non orienté*, appelé simplement *graphe* et noté $G = (V, E)$, est le couple formé :

i) d'un ensemble fini V dont les éléments sont appelés *sommets* ($|V| = n$) ;

ii) d'un ensemble fini E dont les éléments sont appelés *arêtes* ($|E| = m$) ;

A chaque arête e de E on associe une paire non ordonnée de sommets $\{u(e), v(e)\}$. Si $u(e) = a$ et $v(e) = b$ on dira que l'arête e *relie les sommets* a et b ou que les sommets a et b sont *adjacents*. Dans le cas où la paire $(u(e), v(e))$ est ordonnée, le graphe est appelé un *graphe orienté* et ses arêtes sont appelées *arcs*.

Lorsqu'on associe un poids k_{ij} à chaque arête (i, j) on parle de *réseau* et on le note $R = (V, E, K)$. De même il est possible d'associer des poids aux sommets.

Exemple 5.1.2 Graphe à quatre sommets et à quatre arêtes dans les cas non orienté, orienté et pondéré :

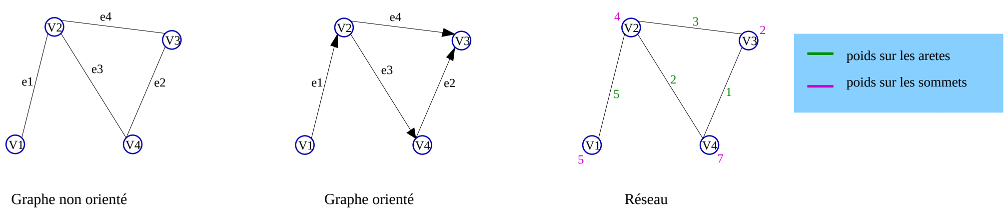


FIG. 5.1 – exemples de graphes

Définition 5.1.3 G' est un *sous-graphe* de G induit par $W \subseteq V$, si $G' = (W, E(W))$, où $E(W)$ est l'ensemble des arêtes ayant leurs deux extrémités dans W.

Exemple 5.1.4 Sous-graphe à quatre sommets et trois arêtes tiré d'un graphe à cinq sommets et cinq arêtes :

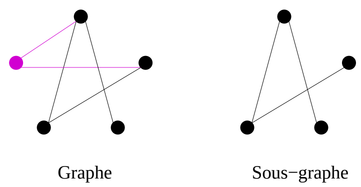


FIG. 5.2 – sous graphe

Définition 5.1.5 Un graphe $G = (V, E)$ est dit *biparti* s'il existe une partition de V en deux ensembles V_1 et V_2 telle que pour toute arête $\{i, j\}$ de E on ait une extremité dans V_1 et l'autre dans V_2.

Exemple 5.1.6 Graphe biparti à neuf sommets et neuf arêtes :

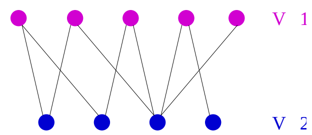


FIG. 5.3 – graphe biparti

Définition 5.1.7 Le concept d'*hypergraphe* est une extension du concept de graphe : une « arête » d'un hypergraphe, appelée *hyperarête*, peut relier plus que deux sommets à la fois.

Exemple 5.1.8 Hypergraphe à six sommets et trois hyperarêtes :

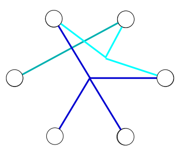


FIG. 5.4 – hypergraphe

Définition 5.1.9 Soit $G = (V, E)$. Un *couplage* $C \subseteq E$ est un ensemble d'arêtes deux à deux disjointes, c'est-à-dire $\{x, y\} \cap \{v, w\} = \emptyset$ pour tout $(x, y), (v, w) \in C$.
Un couplage de cardinalité maximale (tel que $C = argmax_{C' \ couplage}|C'|$) est appelé *couplage maximum*.

Exemple 5.1.10 Nous illustrons la différence entre couplage maximum et couplage de poids maximal par les graphes suivants. En général l'ensemble des arêtes qui forment le couplage maximum n'est pas égal à l'ensemble des arêtes qui forment le couplage de poids maximal.

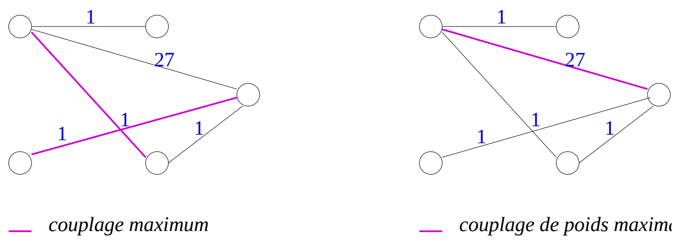


FIG. 5.5 – couplages

Définition 5.1.11 Soit $G = (V, E)$. Un *stable* $S \subseteq V$ est un ensemble de sommets deux à deux non adjacents, c'est-à-dire $x, y \in S \Rightarrow (x, y) \notin E$.
Un stable S de cardinalité maximale (tel que $S = argmax_{S' \ stable}|S'|$) est appelé *stable maximum*.

Exemple 5.1.12 Comme pour le couplage, nous illustrerons la différence entre stable maximum et stable de poids maximal par les graphes suivants :

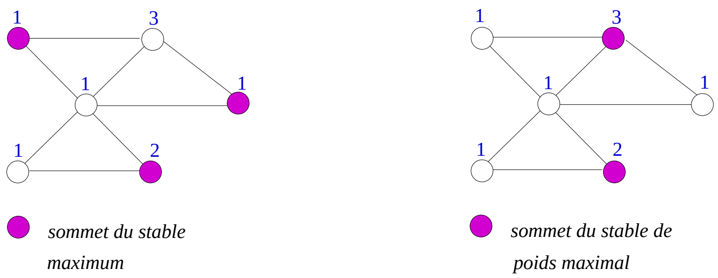


FIG. 5.6 – stables

Définition 5.1.13 Une *clique* est un ensemble de sommets K tel qu'il existe une arête reliant tout couple de sommets dans K.

Exemple 5.1.14 Clique à six sommets dans un graphe à dix sommets :

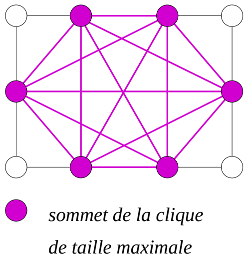


FIG. 5.7 – clique

Définition 5.1.15 Une *coloration* est une partition de l'ensemble des sommets en classes telles que deux sommets appartenant à une même classe ne sont pas reliés par une arête.

Exemple 5.1.16 Dans le graphe 1, contrairement au graphe 2, nous n'avons pas une coloration car nous avons deux sommets adjacents ayant même couleur (et donc appartenant à la même classe).

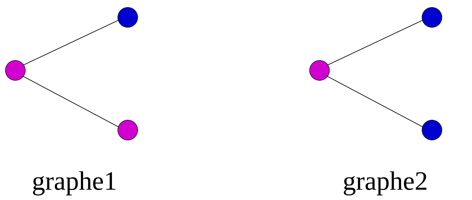


FIG. 5.8 – coloration

Remarque 5.1.17 Les classes données par une coloration sont de fait des stables.

Chapitre 6

Modélisation en terme de graphes du problème de la répartition des titres sur les modules

Description du problème en terme de graphes : étape 1

Le problème de l'affectation des titres de journaux sur les modules peut être modelisé par un hypergraphe G biparti. Les sommets de ce graphe seront donc partagés en deux sous-ensembles de sommets : les sommets de V_1 représentent les titres de journaux (V_1 compte donc N sommets) et les sommets de V_2 représentent toutes les positions sur les modules (et compte donc $2 \cdot N_k \cdot N_m$ sommets).

Nous savons que le titre de journal j doit occuper k_j positions sur les modules et que ces k_j positions doivent être consécutives. Nous joignons donc le sommet j de V_1 aux sommets $p_{i_1}, \ldots, p_{i_{k_j}}$ par une hyperarête si le titre de journal j peut être affecté aux positions $p_{i_1}, \ldots, p_{i_{k_j}}$. Ainsi, si par exemple le titre de journal j doit occuper trois positions, toutes les hyperarêtes qui partent du sommet j vont toucher trois sommets dans V_2. Ces trois sommets doivent représenter trois positions consécutives et appartenant au même module.

Afin d'indiquer les préférences par rapport aux positions pouvant être affectées à chaque journal, nous avons choisi d'attribuer un poids à chacune des hyperarêtes.

Dans cette première étape nous ne considérons pas encore les contraintes d'incompatibilité entre titres de journaux (deux titres semblables de journaux ne doivent pas être affectés a un même module). Ce type de contrainte ne peut pas être exprimé dans l'hypergraphe que nous venons de construire, mais sera pris en compte lors de la dernière étape de notre modélisation.

Exemple 1

Nous traitons ici un petit exemple avec quatre titres de journaux. Ces titres sont représentés par les quatre sommets de gauche et par quatre couleurs différentes. Dans notre exemple, nous ne considérons que deux modules et cinq positions par module (pour un total de dix positions). Voici les contraintes que nous avons choisies :

i) Le titre de journal 1 (en rouge) est de type normal et peut donc être affecté à n'importe quelle position, mais de préference sur le module 2. De plus le journal 1 nécessite trois positions. La contrainte de préference est décrite par la pondération suivante des hyperarêtes : les hyperarêtes touchant les sommets du deuxième module ont le poids 3, tandis que les hyperarêtes touchant les sommets du premier module ont le poids 1. La contrainte concernant le nombre de positions s'exprime par le fait que les hyperarêtes touchent chacune trois sommets de droite.

ii) Le titre de journal 2 (en bleu clair) doit occuper seulement une position et peut être affecté à n'importe quelle position, mais de préference vers la fin des modules.

iii) Le titre de journal 3 (en bleu foncé) doit occuper deux positions, peut uniquement être affecté au deuxième module et doit se trouver le plus proche possible de la fin du module. Aussi associons-nous un poids aux hyperarêtes de façon à ce que ces poids soient décroissants en fonction de leur distance à la fin du module.

iv) Le titre de journal 4 (en vert) doit occuper trois positions et peut uniquement être affecté aux positions finales du deuxième module.

Ainsi nous pouvons contruire l'hypergraphe associé à ce problème comme nous l'avons décrit ci-dessus. L'hypergraphe G obtenu est celui de la figure 6.1.

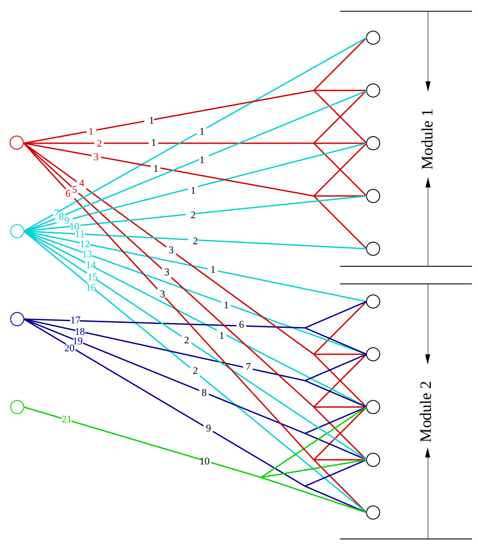


FIG. 6.1 – Hypergraphe G

Suite de la modélisation : étape 2

Avec la construction précedente, certaines hyperarêtes de l'hypergraphe G sont superflues : en effet certains titres de journaux doivent être affectés à une position bien définie. Dès lors, pour simplifier l'hypergraphe initial, nous ne considèrerons plus les titres ayant des positions fixées.

Ces titres sont ceux possédant une unique hyperarête incidente au sommet qui les représente. Nous nous occuperons ainsi d'un *sous-hypergraphe* de l'hypergraphe initial, qui ne contient ni les sommets associés aux titres à position fixée ni les sommets qui représentent les positions occupées par ces titres.

Dans ce nouvel hypergraphe G_{induit}, nous cherchons un couplage maximum de poids total maximal, ce afin d'établir quel titre de journal sera affecté à quelles positions.

Reprise de l'exemple 1

Il est clair que l'hyperarête verte de la figure 6.1 est fixée et que le titre de journal représenté par la couleur verte doit être affecté aux dernières positions du deuxième module. Nous retirons donc l'hyperarête verte ainsi que les sommets qui lui sont adjacents.
Voici l'hypergraphe obtenu :

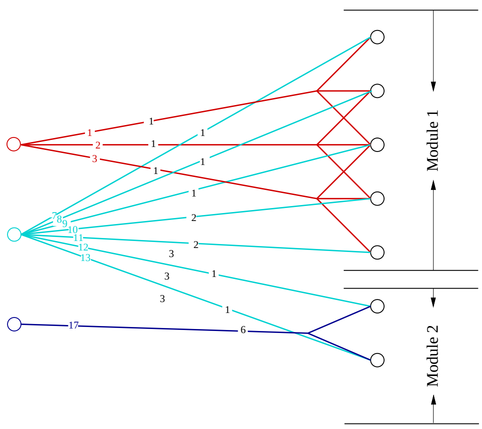


FIG. 6.2 – première réduction de G

De la même façon nous pouvons voir que dans ce nouvel hypergraphe, le titre de journal représenté par la couleur bleue n'est touché que par une unique hyperarête. Nous ôtons donc cette hyperarête ainsi que les sommets qui lui sont adjacents.
Nous obtenons l'hypergraphe suivant :

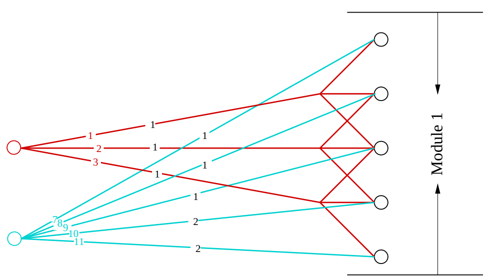


FIG. 6.3 – Hypergraphe G_{induit}

Cet hypergraphe ne pouvant plus être réduit, il s'agit donc de l'hypergraphe G_{induit} cherché pour notre modélisation.

Suite de la modélisation : étape 3

Le problème de la recherche d'un couplage maximum de poids total maximal dans un *hypergraphe* peut être ramené a un problème de recherche d'un stable maximum de poids total maximal dans un *graphe*.
Pour ce faire nous construisons à partir de l'hypergraphe G_{induit} un nouveau graphe G', que nous appellerons *graphe de conflit*, de la façon suivante :

i) les hyperarêtes de G_{induit} deviennent les sommets de G' et les poids des hyperarêtes de G_{induit} deviennent les poids des sommets de G' ;

ii) nous dessinons une arête entre les sommets i et j de G' s'il y a une situation de conflit dans l'hypergraphe G_{induit}, c'est-à-dire si les hyperarêtes i et j sont incidentes à un sommet commun.

Reprise de l'exemple 1

Appliquons le procédé décrit ci-dessus et représentons le graphe G' associé à l'hypergraphe G_{induit} de la figure 6.3. Nous obtenons le graphe suivant :

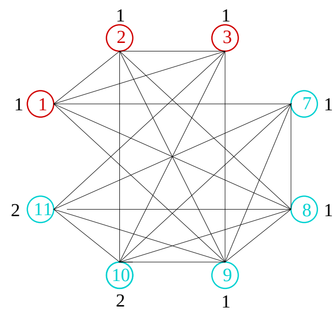


FIG. 6.4 – Graphe G'

La résolution du problème de la recherche d'un stable maximum de poids maximal pourrait par exemple nous donner comme solution les sommets 10 et 1.

Remarque 6.0.18 Nous pouvons remarquer que dans ce modèle tous les sommets représentant un même journal forment une clique.

Quatrième étape de la modélisation : les contraintes de compatibilité

Comme nous l'avons déjà mentionné, les contraintes d'incompatibilité entre titres de journaux n'ont pas encore été prises en compte jusqu'ici. Nous allons le faire maintenant en rajoutant dans le graphe G' (obtenu après l'étape 3) des arêtes supplémentaires si des hyperarêtes du graphe G_{induit} ne peuvent pas être choisies simultanément à cause d'une contrainte de compatibilité.

Ceci revient à rechercher un stable maximum de poids maximal dans ce graphe.

Remarque 6.0.19 Dans la résolution du probème réel, les poids associés aux préference ne seront pas pris en compte car nous ne disposons d'aucune donnée les décrivant.

Exemple 2

Prenons un nouvel exemple où G_{induit} est l'hypergraphe suivant :

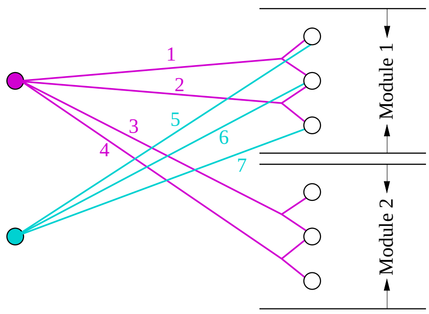


FIG. 6.5 – Graphe G_{induit}

Considérons une contrainte d'incompatibilité imposant que le titre de journal violet et le titre de journal bleu ne peuvent pas être affectés aux même module à cause de leur libellé.
Il est alors clair que les hyperarêtes 1 et 7 ainsi que les hyperarêtes 2 et 5 ne peuvent pas être choisies en même temps. Ainsi le graphe de conflit, complété avec les arêtes nécéssaires pour exprimer la contrainte d'incompatibilité, sera le suivant :

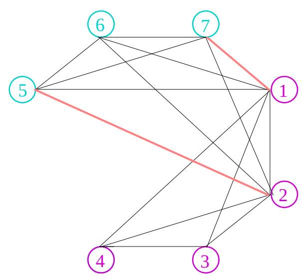
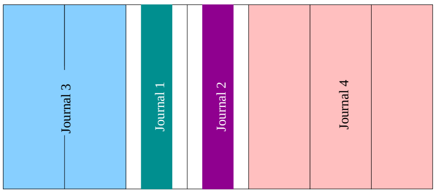


FIG. 6.6 – Graphe G'

Remarque

Nous avons considéré pour le moment qu'un journal qui doit être placé sur k colonnes avec k non entier, occupe par contre dans notre modèle $\lceil k \rceil$ colonnes (avec $\lceil k \rceil$ la partie entière supérieure de k). Avec cette simplification certaines demi-positions sur les modules pourraient rester inoccupées.
Considérons l'exemple suivant dans lequel les journaux 1 et 2 doivent occuper une demi-position, le journal 3 doit occuper deux positions, tandis que le journal 4 doit en occuper trois.



Distribution 1

Distribution 2

Si nous considérons que les journaux 1 et 2 occupent 1 position nous obtenons par exemple une distribution comme la distribution 1 de la figure précedente. De cette façon une position serait perdue.
Si par contre nous considérons la distribution 2, nous gagnons une position sur laquelle nous pourrons placer un autre journal.

Chapitre 7

Modélisation en terme de graphe du problème de la répartition des journaux sur les bulletins de livraison

Modélisation : étape 1

Pour modéliser en terme de graphe le problème de la répartition des journaux sur les bulletins de livraison, nous construisons pour chaque commande un graphe d'une façon que nous allons décrire ci-dessous. Ce problème sera ramené à un problème de coloration des sommets d'un graphe, avec un nombre de couleurs maximal donné d et tel que le nombre de conflits soit minimisé. Il y aura conflit lorsque deux sommets adjacents reçoivent la même couleur.
Le nombre d peut être estimé de la façon suivante : nous connaissons pour chaque client la liste des journaux qu'il désire recevoir. Ainsi nous pouvons calculer le poids total p_{tot} de tous ces journaux, ainsi que la hauteur totale h_{tot}. De plus nous savons qu'un colis doit peser au maximum p_{max} et avoir une hauteur plus petite que h_{max}. Ainsi une borne inférieure pour d sera donnée par

$$d \geq \max\{\frac{p_{tot}}{p_{max}}, \frac{h_{tot}}{h_{max}}\}$$

Il est toutefois possible qu'une coloration en $\max\{\frac{p_{tot}}{p_{max}}, \frac{h_{tot}}{h_{max}}\}$ couleurs n'existe pas. Il faudra en ce cas augmenter la valeur de d de une ou plus unités.

Nous rappelons que la répartition des titres de journaux sur les modules est déjà connue, grâce à la résolution du modèle décrit au chapitre précédent.
Considérons donc la commande du client t. Nous savons que le client t désire recevoir c_{jt} exemplaires d'un titre de journal j. Nous traiterons différemment les titres de journaux *non-sécables* (pour lesquels il est impossible de partager le nombre d'exemplaires d'un client en plusieurs colis) et les titres *sécables*. Dans le graphe que nous sommes en train de construire, nous attribuons un sommet à chacun des exemplaires des titres sécables requis lorsque nous attribuons un unique sommet à l'ensemble des exemplaires pour un titre non-sécable.

Colorer les sommets de ce graphe correspond à trouver un ensemble de journaux affectés à un même bulletin de livraison et donc à un même colis.

Exemple

Considérons la commande du client « Arlecchino » (A) suivante :
3 exemplaires du journal j_1 : $c_{j_1 A} = 3$,
2 exemplaires du journal j_2 : $c_{j_2 A} = 2$,
7 exemplaires du journal j_3 : $c_{j_3 A} = 7$.

Dans cet exemple, supposons que les titres de journaux j_1, j_2 et j_3 se trouvent sur le module 1, et rajoutons la contrainte que les exemplaires du journal j_3 soient non-sécables.

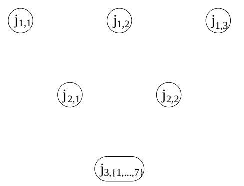


FIG. 7.1 – Exemple

Suite de la modélisation : étape 2

Nous rappelons que notre objectif est d'éviter que les opérateurs aient trop de picks à placer sur des bulletins de livraison consécutifs.
Pour exprimer cet objectif en terme de graphe, nous introduisons une arête entre tout couple de sommets représentant des journaux n'ayant pas le même titre, ceci seulement pour les couples de journaux appartenant au même module.
Ces arêtes expriment le fait que ces journaux ne doivent de préférence pas être affectés au même bulletin de livraison.

Reprise de l'exemple

Reprenons l'exemple précedent avec une augmentation de la commande et en représentant les arêtes.
Suite de la commande :
2 exemplaires du journal j_4 : $c_{j_4 A} = 2$,
8 exemplaires du journal j_5 : $c_{j_5 A} = 8$,
3 exemplaires du journal j_6 : $c_{j_6 A} = 3$.

Considérons en outre que les exemplaires du journal j_5 soient non-sécables et que les titres de journaux j_4, j_5 et j_6 se trouvent sur le module 2.

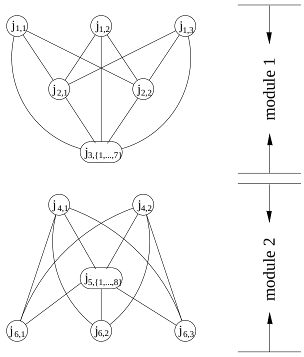


FIG. 7.2 – Exemple

Ce graphe n'est pas connexe car les sommets de modules différents ne sont pas liés par une arête.

Suite de la modélisation : étape 3

Attribuons à chaque sommet deux poids différents, indiquant la hauteur et le poids physique du journal représenté par le sommet en question. Dans le cas d'un sommet de journal non-sécable, les deux poids indiqueront respectivement la somme des hauteurs et la somme des poids des journaux qui le composent. Ces deux informations sont nécessaires, car chaque colis doit avoir une hauteur et un poids maximal à respecter. Il faudra donc que la somme des hauteurs des journaux appartenant au même bulletin de livraison soit inférieur à cette hauteur maximale, et de même pour le poids.

Reprise de l'exemple

Voici le graphe après avoir rajouté sur chaque sommet la hauteur et le poids :

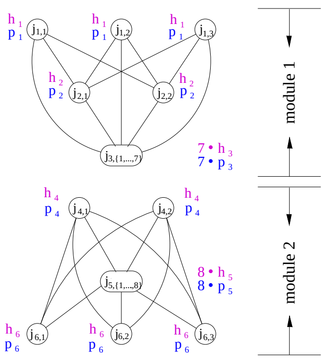


FIG. 7.3 – Exemple

Suite de la modélisation : étape 4

Notre problème est finalement ramené à un problème de recherche d'une coloration généralisée des sommets d'un graphe en d couleurs, coloration qui doit minimiser le nombre de conflits tout en réspectant des contraintes de poids et de hauteur des colis.

Il s'agit maintenant de définir un ordre de passage des bulletins de livraison d'un client sur la bande. En effet l'ordre de « traitement des clients » est fixé à l'avance. Pour faire cela, donc pour ordonner les bulletins de livraison, nous rajoutons la contrainte supplémentaire suivante : s'il y a conflit entre deux sommets d'un module m ayant une même couleur l, le bulletin de livraison représenté par la couleur l ne pourra pas être suivi par un bulletin de livraison qui contient des sommets appartenant au même module m.
Cette contrainte limite à deux le nombre de gestes pour deux bulletins de livraison consécutifs.

Reprise de l'exemple

Nous voulons colorer le graphe avec 5 couleurs (rappelons que le nombre de couleurs représente le nombre de bulletins de livraison, donc le nombre de colis).
En considérant des contraintes de poids maximal et de hauteur maximale des colis, nous obtenons par exemple la coloration suivante :

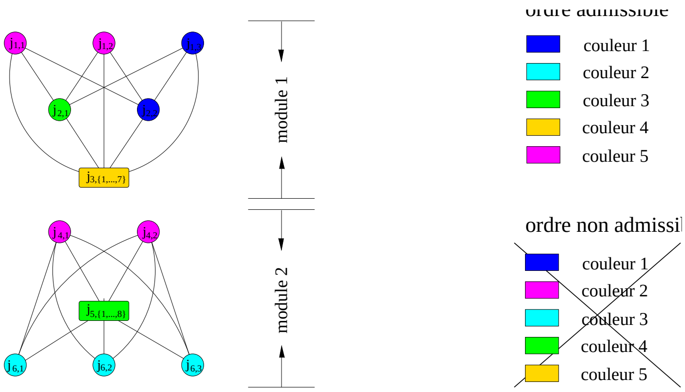


FIG. 7.4 – Exemple

Nous voyons qu'il y a une situation de conflit entre les sommets $j_{1,3}$ et $j_{2,2}$. Ainsi la couleur affectée à ces deux sommets – la couleur « bleu »– ne pourra pas être suivie d'une couleur appartenant au même module. Les couleurs « vert », « jaune » et « rose » ne pourront donc pas suivre immédiatement cette première couleur. Ceci nous donne par exemple un ordre de passage des couleurs comme décrit en haut à droite de la figure.

Voici comment seraient construits les bulletins de livraison par rapport à cet ordre :

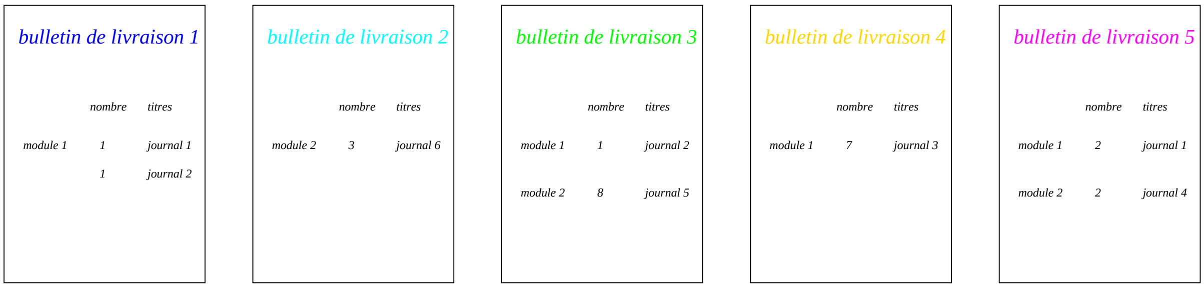


FIG. 7.5 – Ordre admissible

À l'opposé, la liste de passage donnée en bas à droite de la figure 7.4 n'est pas optimale comme nous pouvons facilement le remarquer par la représentation suivante :

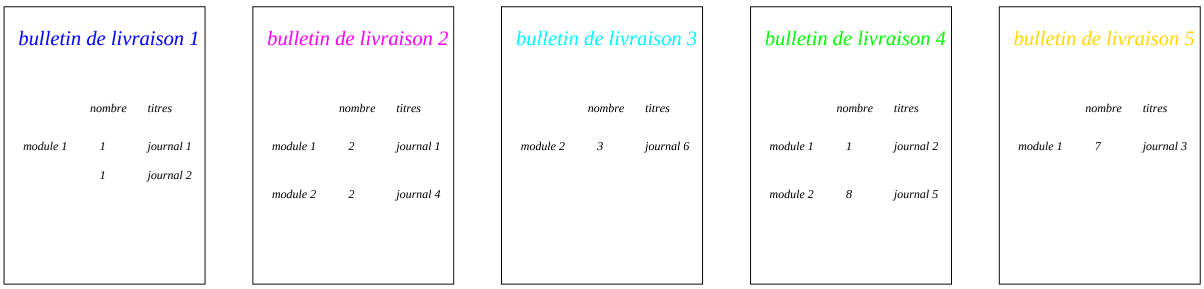


FIG. 7.6 – Ordre non admissible

Nous pouvons constater en effet, que la couleur « bleu » ne peut pas être suivie par la couleur « violet », sinon l'opérateur du module 1 aurait deux picks sur le bulletin de livraison 1 et un pick sur le bulletin de livraison suivant, ce qui pourrait entraîner l'arrêt de la bande. Nous sommes donc obligés de choisir la couleur « bleu ciel » comme suivant immédiatement la couleur « bleu » car c'est la seule qui ne se trouve pas sur le module où il y a conflit.
Pour les autres couleurs l'ordre est aléatoire car nous n'avons pas d'autres conflits.

Chapitre 8

Algorithmes de résolution

Lorsqu'on aborde un problème d'optimisation, il est possible d'utiliser différentes approches qui dépendent de la taille et de la difficulté spécifiques du problème, ainsi que des objectifs que l'on veut atteindre.
Lorsque la complexité et la taille du problème sont grandes, il est difficile de produire une solution qui soit optimale en des temps de calcul raisonnables. Il est alors utile de considérer une approche par des *heuristiques.* Une heuristique est une méthode de recherche qui ne fournit pas forcément *la* solution optimale, mais *une* solution que l'on peut espérer être bonne. L'avantage de telles méthodes est celui de fournir des solutions proches de l'optimum en des temps relativement courts.
La structure et l'idée de base de chaque heuristique sont en substance fixées, mais l'implémentation spécifique des pas de l'algorithme dépend du problème.

Nous avons décidé d'utiliser de telles méthodes pour la résolution de nos problèmes. En particulier, l'heuristique que nous proposons d'appliquer est la *méthode tabou.*

8.1 La méthode tabou

Le but de cette méthode est d'approcher la solution exacte d'un problème donné. Supposons que le problème consiste à minimiser une certaine fonction objectif f.

Pour la description de cette méthode nous avons besoin de quelques définitions :

Définition 8.1.1 *L'espace des solutions* S représente l'ensemble de toutes les solutions possibles pour un problème donné.

Définition 8.1.2 Deux solutions s et s' sont dites *voisines* si l'on peut obtenir s' en modifiant *légèrement* s selon une règle bien définie. Nous dirons que le passage de s à s' se fait par un *mouvement.*
Le *voisinage* $V(s)$ d'une solution $s \in S$ est l'ensemble des solutions voisines de s.

Une méthode tabou est une méthode qui, étant donné une *solution initiale*, effectue le meilleur mouvement dans le voisinage de cette solution (c'est-à-dire le mouvement qui donne la meilleure solution du voisinage) tout en interdisant des solutions qui se trouvent dans une liste T qui s'appelle *liste tabou*. La liste T contient les t dernières solutions visitées durant les t dernières itérations. Les solutions qui font partie de T sont appelés *solutions tabou*. Autrement dit, on choisit la meilleure solution s' de $V(s) \setminus T$.
D'autre part chercher chaque fois dans la liste tabou si une solution est interdite est très coûteux. Il y a néanmoins une autre approche de la méthode tabou qui consiste à ne pas interdire une solution déjà visitée, mais plutôt d'interdire certains mouvements définis pour le problème spécifique. Ainsi par cette variante, si un certain mouvement est effectué, le mouvement inverse ne pourra pas être effectué pendant les t itérations suivantes. Dans ce cas la liste tabou T contient les t derniers mouvements effectués durant les t dernières itérations. Les mouvements qui font partie de T sont appelés *mouvements tabou*. En géneral on admet quand même un mouvement tabou si celui-ci amène à une meilleure solution que la solution courante.
En considérant maintenant la nouvelle solution atteinte et en itérant cette même procédée, l'algorithme tabou retient la meilleure solution rencontrée. L'algorithme s'arrête seulement lorsqu'un *critère d'arrêt* est satisfait. Cette methode ne s'arrête donc pas si l'on détériore la solution courante, mais après :

i) un nombre d'itérations p fixé à l'avance, ou bien
ii) un nombre d'itérations p sans amélioration de la solution courante, ou bien
iii) un temps de calcul fixé à l'avance.

Une méthode tabou peut donc être décrite par :

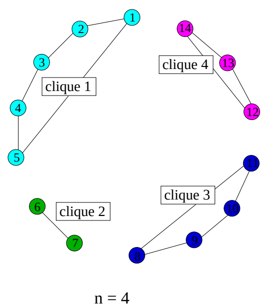
1. la définition de ce qu'est une solution admissible ;
2. la définition de la solution initiale (ou éventuellement la méthode pour la construire) ;
3. la définition du *voisinage* d'une solution ;
4. la manière de choisir une solution voisine à la solution courante ;
5. le (les) *critère(s) d'arrêt* ;
6. la liste tabou.

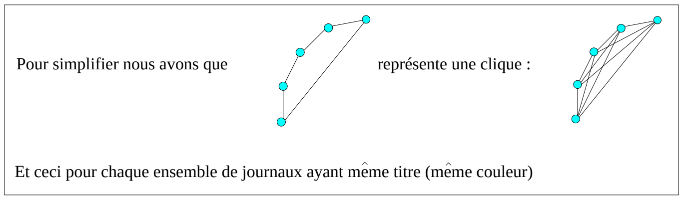
8.2 L'algorithme tabou

1. Construire une solution initiale s ;
2. Poser $s^* = s$ et $f^* = f(s)$ (meilleure solution rencontrée) ;
3. Tant qu'un *critère d'arrêt* n'est pas satisfait, faire :
   - si T contient des solutions, choisir la meilleure solution s' dans $V(s) \setminus T$;
   - si T contient de mouvements, effectuer le meilleure mouvement non tabou pour obtenir la solution voisine s'.
     mais on autorise un mouvement tabou si $f(s') < f(s^*)$
   - si $f(s') < f(s^*)$, poser $s^* = s'$ et $f^* = f(s')$;
   - poser $s = s'$ (la nouvelle solution courante est s').
4. Retourner la solution s^*.

8.3 Algorithme tabou pour la résolution du problème de la répartition des titres de journaux sur les modules

Nous allons définir dans ce paragraphe les notions décrivant les paramètres de la méthode tabou pour le premier problème que nous avons traité. Nous rappellons que nous cherchons à déterminer un stable de taille égale au nombre de titres de journaux à placer sur les modules. Nous cherchons ce stable dans un graphe qui a la particularité d'être tel que les sommets représentant les positions pour un même titre de journal forment une clique entre eux.





1. Sous ces conditions, une solution est un ensemble de sommets se trouvant chacun dans une clique différente. Nous pouvons représenter cette solution par un vecteur de sommets $S = (s_1, s_2, \ldots, s_n)$, où s_j est un sommet se trouvant dans la clique j pour tout j.
   La solution recherchée est donc un ensemble de tels sommets deux à deux disjoints.

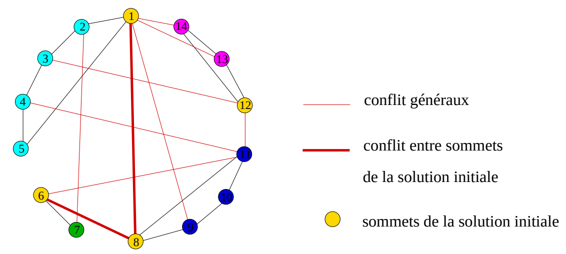
2. Nous proposons ici deux possibilités de choisir la solution initiale :
   - Nous pouvons choisir comme solution initiale celle qui comprend le « premier sommet » de chaque clique.
     Dans l'exemple ci-dessus notre solution initiale serait $(1, 6, 8, 12)$.
   - Le premier choix décrit ci-dessus n'est pas le choix optimal car dans un problème réel il arrive très souvent que les premiers sommets de chaque titre de journal correspondent aux premières positions sur les modules. En effet les structures des journaux sont souvent semblables, ce qui fait qu'ils peuvent être affectés aux mêmes positions.
     Cette situation nécessiterait d'un grande nombre d'itérations avant d'arriver à une solution optimale, car la solution initiale compterait un grand nombre de conflits.
     Aussi nous proposons une alternative pour éliminer ce problème tout simplement en introduisant un facteur non déterministe : nous choisirons comme solution initiale une solution qui comprend un sommet par clique, celui-ci étant choisi aléatoirement.
     Ce choix diminue le nombre d'itérations mais ne change pas le temps nécessaire à l'algorithme pour effectuer une itération.
     Dans l'exemple ci-dessus une telle solution initiale possible serait $(5, 6, 9, 13)$.

3. Les solutions voisines d'une solution $(s_1, s_2, \ldots, {s^k}_i, \ldots, s_n)$ sont toutes les solutions $(s_1, s_2, \ldots, {s^t}_i, \ldots, s_n)$ avec $t \neq k$ et où les sommets ${s^k}_i$ et ${s^t}_i$ font partie de la même clique i.

4. Nous proposons ici deux possibilités de choisir la solution voisine à la solution courante :

   – Première possibilité : parmi tous les sommets de S, nous choisissons le sommet qui a le plus grand nombre de conflits avec les autres sommets de S et nous l'échangeons avec le sommet de la même clique qui provoquera le moins de conflits avec tous les autres sommets de S.

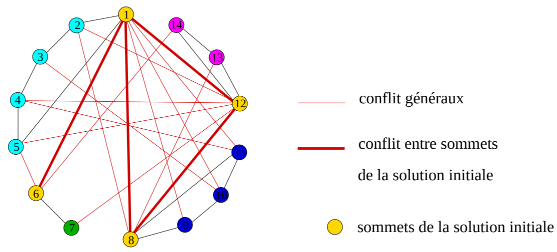
     Exemple : Pour illustrer le procédé que nous venons de décrire, considérons le graphe suivant à 14 sommets. Les arêtes en gras représentent les conflits entre les sommets de la solution courante.

     Dans l'exemple de la figure précedente nous considérions comme solution courante la solution initiale $(1, 6, 8, 12)$. Ainsi le sommet 1 engendre un unique conflit, le sommet 6 aussi, le sommet 8 en engendre deux et enfin le sommet 12 n'engendre pas de conflits. Donc nous échangerons le sommet 8 avec l'un des sommets 9, 10 ou 11.
     Si nous échangeons le sommet 8 avec le sommet 9, nous engendrons un conflit entre le sommet 9 et le sommet 1. Si nous échangeons le sommet 8 avec le sommet 10, aucun conflit n'est engendré. Enfin si nous échangeons le sommet 8 avec le sommet 11, nous engendrons deux conflits. Nous remplaçons donc le sommet 8 avec le sommet 10.
     Notre nouvelle solution est $(1, 6, 10, 12)$.
     Ici l'algorithme s'arrête car la solution obtenue est un ensemble de sommets formants un stable.

   – Deuxième possibilité : parmi tous les sommets dans chaque clique, nous choisissons le sommet qui a le meilleur rendement. Lorsque nous parlons de rendement, nous nous référons à la situation suivante : considérons une clique et supposons que le sommet s qui fait partie de cette clique et qui fait aussi partie de la solution courante engendre c conflits avec les autres sommets de la solution. Considérons maintenant un sommet quelconque de la même clique et supposons de l'insérer dans la solution courante à la place du sommet s. Ce sommet engendre c' conflits avec les autres sommets de la solution. Ce remplacement nous ferait gagner $c - c'$ conflits. Le nombre $c - c'$ représente le rendement de ce sommet. Nous choisissons donc parmi tous les sommets celui qui a le meilleur rendement et nous l'insérons dans la nouvelle solution.
     L'avantage de ce choix de solution voisine par rapport au choix décrit en (a) est celui de converger en un nombre plus petit d'itérations vers la solution cherchée. Ainsi pour la résolution de notre problème nous préférerons ce choix de mouvement au précédent.

Exemple : Comme avant illustrons le procédé par le graphe suivant :



Reprenons encore une fois l'exemple de la figure précédente, où nous considérions comme solution courante la solution initiale $(1, 6, 8, 12)$.
Nous pouvons remarquer que le sommet 1 engendre trois conflits.
Si nous échangeons le sommet 1 avec le sommet 2, nous engendrons deux conflits, donc le rendement du sommet 2 vaut $3 - 2 = 1$. Si par contre nous échangeons le sommet 1 avec le sommet 3, aucun conflit n'est engendré, donc le rendement du sommet 3 vaut $3 - 0 = 3$. Si nous échangeons le sommet 1 avec le sommet 4 ou le sommet 5, nous engendrons un conflit, donc les rendements des sommets 4 et 5 valent $3 - 1 = 2$.
Si pour chaque clique nous calculons le rendement de tous les sommets du graphe nous pourrions remarquer que le sommet 3 est celui qui a le meilleur rendement. Notre nouvelle solution est donc $(3, 6, 8, 12)$.

5. Le critère d'arrêt que nous imposons est celui où la solution optimale est atteinte car chaque journal doit être affecté à un module. Si ceci n'arrive pas après un nombre fixé d'itérations, nous éliminons dans la meilleure solution obtenue le sommet qui engendre le plus de conflits avec les autres sommets de la solution. L'algorithme sera relancé sur le nouvel ensemble de sommets.

6. La liste tabou que nous avons choisie contient les derniers M sommets qui viennent d'être inserés dans la solution durant les derniers M mouvements.
Nous proposons M entre 7 et 15 après avoir testé l'algorithme.

8.4 Algorithme tabou pour la résolution du problème de la répartition des journaux sur les bulletins de livraisons

Nous allons définir dans ce paragraphe les notions décrivant les paramètres de la méthode tabou pour le deuxième problème traité.
Nous rappelons que nous cherchons à déterminer une coloration généralisée en d couleurs d'un graphe qui comporte 12 composantes connexes (une par module). La coloration devra respecter la contrainte que la somme des poids des journaux associés aux sommets d'une même couleur soit plus petite qu'un poids maximal donné, de même pour la somme des hauteurs des journaux associés aux sommets d'une même couleur.

1. Sous ces conditions, une solution est une partition des sommets en d classes (c'est-à-dire en d couleurs).

Nous pouvons représenter une telle solution S par un ensemble contenant tous les sommets du graphe et partitionnés en d classes : $S = \{C_1, C_2, \ldots, C_i, \ldots, C_d\}$ où C_i est la classe contenant tous les sommets de la couleur i.
La solution optimale est une solution qui minimise le nombre de conflits, un conflit étant une arête entre deux sommets appartenant à une même classe.

2. Nous choisissons la solution initiale telle que les sommets du graphe soient colorés aléatoirement avec d couleurs différents.

3. Nous définissons la relation de voisinage de la manière suivante : une solution S est voisine d'une solution S' si les deux ne diffèrent que par la couleur d'un sommet.
Autrement dit, si $S = \{C_1, C_2, \ldots, C_i, \ldots, C_j, \ldots, C_d\}$ et si s est un sommet de C_i alors $S' = \{C_1, C_2, \ldots, C_i \setminus \{s\}, \ldots, C_j \cup \{s\}, \ldots, C_d\}$.

4. Comme pour le problème précédent nous pourrions choisir comme mouvement de la solution courante à une de ses solutions voisines le mouvement à meilleur rendement, c'est-à-dire le mouvement qui maximise la différence entre le nombre de conflits de la solution courante et le nombre de conflits d'une de ses solutions voisines.
Maximiser le rendement n'est pourtant pas suffisant, en effet il ne faut pas oublier que chaque colis doit respecter des contraintes de poids et de hauteur ; aussi nous introduisons ici un objectif qui traite ces contraintes et qui définit un mouvement. La solution voisine atteinte par le mouvement sera la solution qui minimise le terme suivant :

$$\min \quad -\gamma_0 \cdot \{\text{meilleur rendement}\} + \gamma_1 \cdot \sum\{\text{surplus en kg}\} + \gamma_2 \cdot \sum\{\text{ surplus en cm}\}$$

où
- $\gamma_0, \gamma_1, \gamma_2$ sont des paramètres encore à définir, la valeur desquels donne des poids de préference ;
- $\sum\{\text{surplus en kg}\}$ est la somme des eccès de poids pour chaque colis (représenté par une couleur) ;
- $\sum\{\text{ surplus en cm}\}$ est la somme des eccès de hauteur pour chaque colis.

5. À la différence du problème précédent, le critère d'arrêt que nous pouvons imposer est soit celui où nous avons trouvé une coloration sans conflits, soit une coloration avec un nombre de conflits par couleur plus petit qu'un nombre k donné. Ce nombre k représentera le nombre maximal de gestes qu'un opérateur devra faire pour un bulletin de livraison. Pour que notre optimisation soit efficace, il faudrait que $k \leq 2$.

6. La liste tabou que nous avons choisie contient les derniers M sommets qui viennent de passer d'une classe à l'autre durant les dernières M itérations de l'algorithme. Autrement dit, nous ne pourrons pas changer la couleur d'un sommet qui a changé de couleur durant les dernières M itérations.

Chapitre 9

Résultats de l'implémentation de l'algorithme pour le problème de la répartition des titres de journaux sur les modules

Dans ce chapitre nous rapportons les résultats obtenus en créant un programme capable de lire les données reçues, de créer les sommets et les arêtes de la façon décrite au chapitre 6 et en traitant ces données par l'algorithme proposé au chapitre 8.

Pour l'implémentation nous avons utilisé une numérotation des positions différente de celle utilisée pour la description du modèle mathématique.
Nous montrons cette numérotation dans la figure suivante afin que le lecteur comprenne les résultats fournis par l'algorithme.

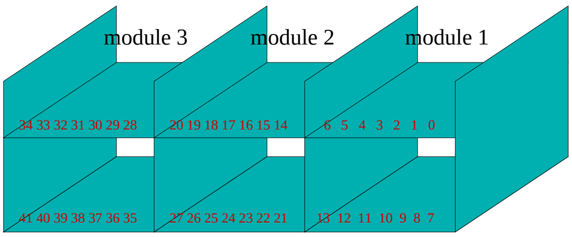


Fig. 9.1 – positions pour l'implémentation

9.1 Petit exemple

Nous avons demandé aux responsables de l'entreprise de nous fournire des données sous la forme suivante :

TITRE DU JOURNAL ;
objets-cadeaux contenus dans le journal ;
format du journal ;
poids en kilos du journal ;
hauteur du journal (donnée selons des catégories) ;
bombé ou normal ;
nombre d'exemplaires du journal à traiter ;
sécable (représenté par « O ») ou non-sécable (représenté par « N »).

Les données pour un titre de journal se trouvent sur une même ligne.

Nous proposons ici un minuscule exemple dans lequel nous allons expliquer les étapes de la résolution.
Nous avons choisi un exemple petit de façon à pouvoir montrer tous les sommets et toutes les arêtes créés, car un exemple de taille plus grande engendre beaucoup plus de sommets et d'arêtes et montrer toutes les étapes de résolution dans un exemple plus grand n'apporterait rien de plus que l'exemple proposé.

```
MIAM ;DVD boîte ;A5 ;0.24 ;B ;bombé ;102 ;O
MIAM CUISINE ;DVD boîte ;A4 ;0.229 ;A ;Normal ;120 ;O
VIVE LA PIZZA ;Cd boîte ;A4 ;0.271 ;A ;bombé ;376 ;O
```

En premier lieu, notre programme lit les données pour chaque journal, teste les possibilités des places sur les modules où le titre de journal pourrait être placé et crée les sommets correspondants.

```
***************************
 id :          0      id est le numéro associé au titre de journal
 titre :       MIAM
 produit :     3      les valeures suivantes ont été traduites comme
 format :      3      faisant partie de certaines catégorie
 poids :       1
 hauteur :     2
 qualite :     1
 quantite :    0
 secabilite :  1
***************************

****** Sommet 0 ************
 idjournal :          0
 libelle :            MIAM   le libellé donne les premières quatre lettres du titre
 premierPos :         131    sont la première et l'avant-dernière position sur les
 apresDernierePos :   133    modules sur lesquelles ce titre de journal peut être placé
***************************
```

****** Sommet 1 ************

idjournal :	0
libelle :	MIAM
premierPos :	145
apresDernierePos :	147

****

****** Sommet 2 ************

idjournal :	10
libelle :	MIAM
premierPos :	159
apresDernierePos :	161

****

Ainsi le journal MIAM peut être placé soit sur les positions 131 et 132, soit sur les positions 145 et 146, soit sur les positions 159 et 160.
On peut procéder de même pour les titres de journaux MIAM CUISINE et VIVE LA PIZZA :

****

id :	1
titre :	MIAM CUISINE
produit :	3
format :	1
poids :	1
hauteur :	1
qualite :	0
quantite :	0
secabilite :	1

****

****** Sommet 3 ************

idjournal :	1
libelle :	MIAM
premierPos :	131
apresDernierePos :	133

****

****** Sommet 4 ************

idjournal :	1
libelle :	MIAM
premierPos :	145
apresDernierePos :	147

****

****** Sommet 5 ************

idjournal :	1
libelle :	MIAM
premierPos :	159
apresDernierePos :	161

****

Dans ce cas le journal MIAM CUISINE, comme nous l'avons vu avant pour le journal MIAM, peut être placé soit sur les positions 131 et 132, soit sur les positions 145 et 146, soit sur les positions 159 et 160. Nous pouvons remarquer que les deux premiers journaux ont le même libellé et sont très semblables dans leur données. Il n'est donc pas étonnant que les positions qu'ils peuvent occuper soient les mêmes.

```
****************************
 id :           2
 titre :        VIVE LA PIZZA
 produit :      2
 format :       1
 poids :        1
 hauteur :      1
 qualite :      1
 quantite :     1
 secabilite :   1
****************************

****** Sommet 6 ************
 idjournal :           2
 libelle :             VIVE
 premierPos :          126
 apresDernierePos :    127
****************************

****** Sommet 7 ************
 idjournal :           2
 libelle :             VIVE
 premierPos :          127
 apresDernierePos :    128
****************************

****** Sommet 8 ************
 idjournal :           2
 libelle :             VIVE
 premierPos :          140
 apresDernierePos :    141
****************************

****** Sommet 9 ************
 idjournal :           2
 libelle :             VIVE
 premierPos :          141
 apresDernierePos :    142
****************************
```

****** Sommet 10 ***********

idjournal :	2
libelle :	VIVE
premierPos :	154
apresDernierePos :	155

****

****** Sommet 11 ***********

idjournal :	2
libelle :	VIVE
premierPos :	155
apresDernierePos :	156

****

Comme avant nous pouvons lire les positions possibles pour le titre de journal VIVE LA PIZZA.

Nous pouvons vérifier que les sommets créés sont bien compatibles avec toutes les contraintes du problème.
Une fois les sommets créés, notre programme regarde les conflits entre les sommets et crée les arêtes correspondantes. Nous rappelons qu'il y a une situation de conflit si le libellé de deux sommets représentant deux titres de journaux différents est le même et si ces deux sommets représentent des positions se trouvant sur un même module. De plus il y a une situation de conflit entre deux sommets représentant des positions qui se chevauchent.
Voici les arêtes générées pour cet petit exemple :

****
(x,y)=(0,3)
****
(x,y)=(1,4)
****
(x,y)=(2,5)
****

Un fois que le graphe est créé, notre programme construit la première des deux solutions initiales proposées au § 8.3 .
La liste affichée ci-dessous par le programme comprend le numéro des sommets faisant partie de la solution initiale ainsi que les positions qu'ils représentent.

sommet	premierPos	apresDernierePos
0	131	133
3	131	133
6	126	127

Nous pouvons remarquer que cette solution n'est pas un stable car le sommet 0 et le sommet 3 représentent à eux deux positions 131 et 132. Il y a donc une situation de conflit entre ces deux sommets.
Le programme calculera ensuite le rendement de chaque sommet, donnera la nouvelle solution et mettra à jour la liste tabou.

sommet	premierPos	apresDernierePos
1	145	147
3	131	133
6	126	127

Dans notre cas, nous pouvons aisément vérifier que la solution ci-dessus est la solution cherchée : il n'y a plus de conflits entre les sommets.
Ainsi l'algorithme s'arrête et nous pouvons lire la solution suivante : le journal MIAM ira sur les positions 145 et 146, le journal MIAM CUISINE ira sur les positions 131 et 133 tandis que le journal VIVE LA PIZZA sera affecté à la position 126.

9.2 Traitement des données pour une journée

Lorsque nous avons traité les données complètes, nous avons remarqué que certains journaux n'engendraient aucun sommet. Nous nous sommes ainsi rendues compte que les contraintes ne sont pas toujours cohérentes (il faudrait qu'il y ait au moins un sommet pour chaque journal, car sinon cela voudrait dire que les journaux en question ne pourraient être placés sur aucun module).

Voici dans la liste que nous avons reçue la sous-liste des journaux qui ont une structure telle qu'aucun sommet ne soit créé.

AUTOMOBILE MAG. ;Normal ;A4 ;0.381 ;A ;Normal ;**3468** ;O
CUISINE ACTUELLE HS ;Normal ;A4 ;0.361 ;A ;Normal ;**7461** ;O
EDELWEISS (NO.46/47) ;Normal ;A4 ;0.409 ;A ;Normal ;**11913** ;O
GEO FR ;Normal ;A4 ;0.56 ;A ;Normal ;**6535** ;O
MAXIMAL ;Normal ;A4 ;0.356 ;A ;Normal ;**3952** ;O
PC EXPERT ;CD plat ;A4 ;0.315 ;A ;bombé ;**3951** ;O
TYPE ;Normal ;A4 ;0.286 ;A ;Normal ;**6931** ;O
COURRIER INTL ;Normal ;A4+ ;0.209 ;A ;Normal ;**2269** ;O
HISTOIRES POUR LES PETITS ;Gros gadget ;A5 ;0.219 ;A ;bombé ;**2357** ;O
MONDE DE L'EDUCATION ;Normal ;**A4+** ;**0.182** ;A ;Normal ;231 ;O

Nous pouvons remarquer qu'en général le facteur commun dans la structure de ces journaux est leur quantité. Ceci peut provoquer des conflits entre les contraintes car en principe chaque journal a une contrainte de type « doit être affécté sur les modules aux positions de droite », « doit être affécté sur les modules aux positions de gauche ». Dans le modèle fourni par l'entreprise, ce type de contraintes impose déja une borne supérieure au nombre de positions possibles pour un journal. Cette borne étant de deux, il est clair que des journax devant occuper plus de deux positions à cause de leur quantité n'engendreront aucun sommet.
Pour le dernier journal de la liste par contre, aucun sommet n'est créé car ce journal doit occuper des positions de gauche (à cause de son format) et des positions de droite (à cause de son poids). Afin de résoudre ce problème l'entreprise devrait nous fournir une liste de préférences de relaxation des contraites pour un journal qui présente de telles particularités.

Pour le moment nous n'avons pas pris en considération tous les journaux problématiques.

Une fois l'algorithme lancé sur la partie restante des données, nous avons remarqué qu'il convergait très vite vers une « solution cuvette », c'est-à-dire qu'il trouve une solution qui est un minimum local et telle que les solutions qui se trouvent dans un certain voisinage sont aussi des minima locaux. Si ce voisinage a cardinalité plus grande que la cardinalité de la liste tabou l'algorithme cycle. Puisque ceci est ce qui arrive dans notre cas, nous avons imposé un mouvement vers une solution engendrée aléatoirement et qui soit atteignable de la solution courante en k mouvements. Ceci afin de permettre à l'algorithme de trouver des meilleures solutions en visitant des mauvaises solutions.

Afin de mieux illustrer cette situation, considérons le graphe suivant, dans lequel nous avons représenté chaque solution par un sommet dont le poids représente le nombre de conflits de la solution en question.

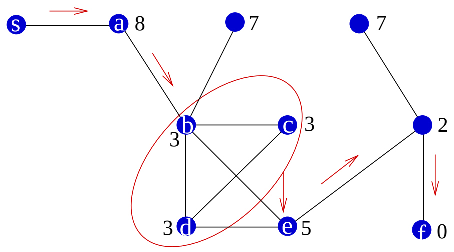


FIG. 9.2 – mouvement forcé

Supposons que l'algorithme se déplace sur les arêtes de ce graphe en venant par le sommet s, et supposons $k = 1$. Alors l'algorithme visitera premièrement le sommet a, puis le sommet b. Il est clair qu'il n'y a pas de solution voisine qui améliore la solution courante (c'est-à-dire qu'il n'y a aucun sommet ayant moins de conflits que b). Sans la condition décrite ci-dessus l'algorithme s'arrêterait après avoir visité les sommets c et d. Notre imposition le force par contre à se déplacer vers une solution voisine, même si celle-ci est moins bonne que la solution courante. Imaginons qu'un tel déplacement amène au sommet e. De cette solution l'algorithme pourra se déplacer vers la solution cherchée qui est représentée par le sommet f.

Aussi avons-nous relancé l'algorithme en imposant des valeurs de k de plus en plus grandes afin d'essayer de sortir du « voisinage cuvette ». Nous nous sommes vite rendues compte que malheureusement cette solution représente la solution qui minimise le nombre de conflits par rapport aux données reçues.
Encore une fois, les données sont incohérentes et une solution comme celle que nous cherchons n'existe pas. En effet le domaine des solutions admissibles est vide.
Il faudra donc par la suite disposer d'une liste de préférences de traitement des contraintes, c'est-à-dire d'une liste qui nous dise l'ordre dans lequel nous devrions relaxer les contraintes jusqu'à pouvoir disposer d'au moins une solution admissible.

Dans notre exemple, la meilleure solution visitée par l'algorithme compte 19 conflits !
Après avoir fait quelques tests, nous avons remarqué qu'il est en général possible de trouver une solution admissible pour une cinquantaine de sommets sans devoir relaxer aucune contrainte. Dans ce cas l'algorithme fournit la solution en quelques secondes.

Chapitre 10

Conclusions

Nous avons réussi avec succès à porter à terme l'optimisation du problème de la répartition des titres de journaux sur les modules, en proposant une modélisation et un algorithme efficaces pour sa résolution. Cependant, il reste des points à améliorer et des problèmes à résoudre :

- Premièrement – comme nous l'avons déjà mentionné – certains titres de journaux ont une structure telle qu'aucune position d'aucun module ne peut leur être affectée. Pour résoudre ce problème, il serait nécessaire de disposer d'un ordre d'importance des contraintes qui permette d'éliminer automatiquement certaines d'entre elles qui font que le problème n'a pas de domaine de solutions admissible. Par exemple, un titre de journal ayant un format tel qu'il doive se trouver sur un certain module, mais dont le poids le contraint à se trouver sur un autre, ne pourra être placé sur aucun des deux modules. Il faudra donc, pour ce titre de journal, relaxer une des deux contraintes selon cet ordre d'importance (que l'entreprise devra fournir) ;
- Deuxièmement, le domaine des solutions admissibles étant vide pour la liste de données qui nous a été fournie, il y a raison de croire qu'il sera vide pour les données de chaque jour. En effet si toutes les contraintes sont prises en compte et si le nombre de journaux à traiter est grand (ce qui est bien le cas) il y aura beaucoup de journaux qui ne pourront pas être placés. Il sera donc également nécessaire de disposer d'une liste de préférences de relaxation des contraintes pour tous les journaux.
- Enfin dans notre modèle nous avons imposé qu'un journal qui doit être placé sur k colonnes avec k non entier occuperait par contre $\lceil k \rceil$ colonnes (avec $\lceil k \rceil$ la partie entière supérieure de k). Une fois une solution trouvée, il serait donc intéressant de réconsidérer les demi-positions afin d'économiser des places et pouvoir ainsi les affecter à d'autres journaux.

En ce qui concerne le problème de la répartition des journaux sur les bulletins de livraison, la partie de la modélisation est terminée. Même si les deux problèmes sont très vastes, nous n'avons pas eu besoin de les simplifier : en effet notre modélisation englobe déjà le cas général !
Reste encore pour le deuxième problème l'implémentation de l'algorithme que nous avons proposé.

Par la suite il serait également intéressant d'essayer de traiter les deux problèmes comme un problème unique. Il s'agirait donc de chercher une modélisation qui puisse fournir une répartition des titres de journaux sur les modules ainsi que la répartition des journaux sur les bulletins de

livraison, et ces deux résultats de façon à être optimales l'un par rapport à l'autre.
Il serait peut-être utile de faire ceci en partant d'une première solution comprenant deux solutions trouvées pour chaque problème séparément.

Ce projet fut intéressant à plus d'un titre. En premier lieu, nous avons eu l'occasion de travailler sur un problème réel, ce qui nous a permis de voir le genre de difficultés pouvant survenir lorsqu'on aborde un problème de cette taille, comme par exemple le fait qu'il n'existe pas de solutions préfabriquées : chaque problème doit être traité individuellement.
Deuxièmement, nous avons pu constater qu'il n'est pas toujours facile de gérer des contacts de travail, notamment la longue attente des données et des réponses à nos questions, ainsi que d'autres difficultées dûes à des quiproquos causés par des problèmes de communication.
Enfin, le travail à deux s'est révélé très stimulant, dans la mesure où nous avons pu mettre nos différentes capacités en commun et résoudre les difficultés à deux.

Bibliographie

[1] Jean-François HÊCHE, Cours de recherche opérationnelle, 2000-2001.

[2] Nicolas ZUFFEREY, Cours de graphes et réseaux, 2001-2002.

[3] Martin RAJMAN, Cours de programmation, 2000-2001.

[4] Marylème MICHELOUD et Medard RIEDER, Programmation orientée objets en C++, Presses polytechniques et universitaires romandes, 1997.