

CAHIER DU LAMSADE

Laboratoire d'Analyse et Modélisation de Systèmes pour l'Aide à la Décision
(Université Paris-Dauphine)
Unité de Recherche Associée au CNRS n° 825

UNE CLASSE POLYNOMIALE DE PROBLÈMES DE CONSTRUCTION DE VACATIONS

CAHIER N° 154
février 1998

Ariane PARTOUCHE ¹
Daniel VANDERPOOTEN ¹

reçu: janvier 1998.

¹ LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75775 Paris Cedex 16, France (e-mail: {partouche,vdp}@lamsade.dauphine.fr).

Table des matières

Abstract-Résumé	i
1 Introduction	1
2 Un cadre d'analyse	2
2.1 Concepts fondamentaux	2
2.2 Graphes générateurs et matrices de vacances	4
2.3 Interprétation temporelle des graphes générateurs	7
3 Une classe de PCV polynomiaux	8
3.1 Une classe générale de matrices de vacances totalement unimodulaires	8
3.2 Le cas des vacances sans pause	9
3.3 Le cas des vacances avec pauses	10
3.3.1 Cas avec plage de pause imposée	10
3.3.2 Un procédé typique de construction d'arbres générateurs	12
3.3.3 Un cas particulier avec plage de pause non imposée	12
3.3.4 Un cas particulier avec fenêtres de pause	13
4 Une modélisation par les flots	15
4.1 Cas de réseaux issus d'arbres générateurs	16
4.2 Une classe étendue de PCV polynomiaux	16
4.3 Un exemple illustratif	19
Références	20

A class of polynomial shift scheduling problems

Abstract

The shift scheduling problem consists in determining a set of shifts to cover a requirement curve while minimizing costs. This problem can be modeled as a general set-covering problem : $\min\{cx : Ax \geq b, x \in \mathbb{N}^n\}$.

The set-covering problem is NP-hard. However, the special structure of the A matrix enables to display some classes of polynomial shift scheduling problems. More precisely, we characterize shift sets for which the associated shift matrix is totally unimodular. Solving the continuous relaxation of the set-covering problem then ensures integer solutions, for any requirement values and for any costs. Thus, we identify a first class of easy shift scheduling problems.

We show also that any problem of this class can be reformulated as a network-flow problem. Thanks to a slight restriction on costs, this new formulation allows us to extend the class of polynomial shift scheduling problems.

Keywords: shift scheduling problem, set covering problem, total unimodularity, network-flow.

Une classe polynomiale de problèmes de construction de vacances

Résumé

Le problème de construction de vacances (PCV) consiste à déterminer un ensemble d'horaires de travail permettant de couvrir une courbe de charge à coût minimal. Le PCV peut être formulé comme un problème de couverture d'ensembles généralisé (PCG) : $\min\{cx : Ax \geq b, x \in \mathbb{N}^n\}$.

Le PCG est un problème NP-difficile. Néanmoins, la structure particulière de la matrice A permet de mettre en évidence des classes de PCV polynomiaux. Plus précisément, nous caractérisons des familles de vacances pour lesquelles la matrice de vacances associée est totalement unimodulaire. La résolution du PCG en variables continues donne alors des solutions entières, et ce pour toutes valeurs de la charge et toute structure de coût. On identifie ainsi une première classe de PCV pouvant être résolus efficacement.

On montre d'autre part que tous les problèmes de cette classe peuvent être reformulés comme des problèmes de flot. Moyennant une légère restriction sur les coûts, cette nouvelle modélisation permet d'étendre la classe de problèmes polynomiaux.

Mots-clés : problème de construction de vacances, problème de couverture d'ensembles, totale unimodularité, flot.

1 Introduction

Le Problème de Construction de Vacances (PCV) constitue un important problème de planification d'horaires lorsqu'une courbe de charge étendue et variable doit être couverte par des employés en horaires décalés. Il en va ainsi par exemple des sociétés de service aéroportuaire, des centres de renseignement ou réservation téléphonique, des postes, des banques, de la grande distribution, etc...

Considérant une charge de travail définie par périodes de temps et un ensemble de vacances admissibles, chaque vacation étant caractérisée par des périodes travaillées et un coût, le PCV consiste à sélectionner les vacances permettant de couvrir la charge à moindre coût.

Depuis Dantzig (1954), qui formule le PCV comme un problème de couverture généralisé, de nombreuses approches heuristiques s'appuyant sur la résolution du problème de couverture relâché des contraintes d'intégrité se sont avérées particulièrement efficaces (voir Partouche (1998) pour un état de l'art). En fait, différents travaux mettent en évidence une forte proportion de d'instances pour lesquels la valeur de l'optimum continu égale la valeur de l'optimum entier (Henderson et Berry (1976); Morris et Showalter (1983); Bechtold et Jacobs (1991), ...). Morris et Showalter (1983), qui considèrent un coût identique pour toutes les vacances, précisent que sur 210 problèmes résolus sans les contraintes d'intégrité, 47,1% ont une solution optimale entière. Partouche (1998) confirme le caractère souvent « facile » de ce problème : sur 139 problèmes de grande taille traités, au moins 77% admettent un optimum entier de valeur égale à celle de l'optimum continu et pour 29% des 139 problèmes résolus, l'optimum entier est obtenu directement en résolvant le problème relâché.

Ainsi, alors que le problème de couverture est un problème combinatoire NP-difficile, sa résolution est souvent étonnamment facile dans le cadre de la construction de vacances. De telles observations nous ont conduits à nous interroger sur les caractéristiques du problème de couverture dans le cas particulier de la construction de vacances et à identifier des classes générales de PCV pouvant être résolus efficacement, c'est-à-dire en temps polynomial.

Un PCV étant défini par un ensemble de vacances admissibles, leurs coûts respectifs et une charge de travail, caractériser une classe de problèmes signifie déterminer des conditions sur les vacances, les coûts et la charge. Nous nous attacherons à imposer les conditions les plus faibles de manière à décrire des classes de portée suffisamment générale. Aucune restriction ne sera donc imposée sur la charge, les conditions portant principalement sur les types de vacances admissibles.

Après avoir rappelé quelques concepts théoriques classiques et introduit le concept central de graphe générateur (section 2), nous étudions le PCV formulé comme un problème de couverture généralisé. Nous identifions une première classe de problèmes dont la résolution par programmation linéaire (en variables continues) fournit une solution optimale entière pour toute courbe de charge et toute

structure de coût des vacances (section 3). On montre ensuite que tous les problèmes de cette classe peuvent être reformulés comme des problèmes de flot à coût minimal. Moyennant une légère restriction sur la structure de coût des vacances, cette modélisation permet d'enrichir la classe de problèmes pouvant être résolus efficacement (section 4).

2 Un cadre d'analyse

Considérons le PCV formulé comme un problème de couverture généralisé :

$$(P) \quad \begin{cases} \min & cx \\ \text{s.c.} & Ax \geq b \\ & x \geq 0, \quad x \text{ entier} \end{cases}$$

où :

A est une matrice ($m \times n$)

avec $a_{ij} = \begin{cases} 1 & \text{si la période } i \text{ est couverte par la vacation } j \\ 0 & \text{sinon} \end{cases}$

c est le vecteur ($1 \times n$) des coûts de vacances,

b est le vecteur ($m \times 1$) des charges à couvrir,

x est le vecteur ($n \times 1$) des nombres de vacances de chaque type.

Dans toute sa généralité, le problème (P) est un problème NP-difficile. Cependant, dans le contexte des PCV, la matrice A que nous appellerons *matrice de vacances*, présente certaines spécificités structurelles. En effet, chaque colonne de A , correspondant à une vacation, se présente comme :

- une séquence de 0 (éventuellement vide),
- une séquence de 1 (éventuellement interrompue par quelques 0 traduisant les périodes de pause),
- une séquence de 0 (éventuellement vide).

Quelques exemples de matrices de vacances sont présentées dans les tableaux 1-5.

2.1 Concepts fondamentaux

Nous présentons d'abord quelques concepts et résultats généraux utilisés tout au long de cet article

Rappelons premièrement quelques concepts élémentaires de la théorie des graphes (voir, par exemple, Roy (1970); Christofides (1975); Ahuja *et al.* (1993); Gondran et Minoux (1995)).

Considérons un graphe orienté $G = (X, U)$ où X représente un ensemble de sommets et $U \subseteq X \times X$ un ensemble d'arcs.

Définition 2.1 (*chaîne, chemin, cycle, circuit*)

- Une chaîne de longueur q est une séquence de q arcs u_1, u_2, \dots, u_q telle que chaque arc u_i ($i = 2, \dots, q-1$) ait une extrémité commune avec u_{i-1} et l'autre avec u_{i+1} . L'extrémité de u_1 non adjacente à u_2 et l'extrémité de u_q non adjacente à u_{q-1} constituent les extrémités de la chaîne.
- Un chemin est une chaîne dont tous les arcs ont la même orientation.
- Un cycle est une chaîne dont les extrémités coïncident.
- Un circuit est un chemin dont les extrémités coïncident.

Définition 2.2 Une chaîne est élémentaire si elle ne passe qu'une fois en chacun de ses sommets.

Définition 2.3 Un graphe est dit connexe si pour tout couple de sommets (i, j) , il existe une chaîne joignant i et j .

Définition 2.4 Un arbre est un graphe connexe et sans cycle.

Présentons maintenant quelques concepts et résultats issus de la théorie de la programmation linéaire en nombres entiers (voir, par exemple, Schrijver (1986); Nemhauser et Wolsey (1988)).

Définition 2.5 Un polyèdre $\mathcal{P} \subseteq \mathbb{R}_+^n$ non vide est dit intégral si et seulement si tous ses points extrêmes sont entiers.

On cherche donc des conditions sur A telles que le polyèdre $\mathcal{P}(b) = \{x \in \mathbb{R}_+^n : Ax \geq b\}$ soit intégral pour tout vecteur b entier. Il est alors évident que la résolution du problème (P) relâché des contraintes d'intégrité conduit à une solution entière, pour tout vecteur c .

Ces conditions sont données par le théorème classique suivant :

Théorème 2.1 (Hoffman et Kruskal (1956)) Soit A une matrice à coefficients entiers. Le polyèdre $\mathcal{P}(b) = \{x \in \mathbb{R}_+^n : Ax \geq b\}$ est intégral pour tout b entier si et seulement si A est totalement unimodulaire.

On rappelle que :

Définition 2.6 Une matrice $A(m \times n)$ à coefficients entiers est dite totalement unimodulaire si et seulement si les déterminants de toutes ses sous-matrices carrées valent 0, 1 ou -1.

Il existe de nombreuses caractérisations de matrices totalement unimodulaires (voir articles de synthèse de Gondran (1973); Padberg (1976) ou ouvrages de Schrijver (1986); Nemhauser et Wolsey (1988)). La plupart de ces conditions nécessaires et suffisantes sont néanmoins peu exploitables car leur vérification est combinatoire. De plus, on s'intéresse ici à des conditions *constructives* permettant de décrire ou générer des familles de matrices totalement unimodulaires plutôt que de vérifier cette propriété sur une matrice existante. Dans cette perspective, on utilise une condition proposée par Camion (1963) qui permet de construire des matrices totalement unimodulaires à partir de graphes particuliers.

Soit $G = (X, U)$ un graphe orienté où $|U| = m$. Considérant dans G une chaîne μ reliant deux sommets x et y , on lui associe le vecteur v à m composantes défini par :

$$v_j = \begin{cases} +1 & \text{si } u_j \in \mu \text{ et est orienté de } x \text{ vers } y \\ -1 & \text{si } u_j \in \mu \text{ et est orienté de } y \text{ vers } x \\ 0 & \text{si } u_j \notin \mu \end{cases}$$

On appelle *matrice de chaînes* définies sur G une matrice dont les colonnes sont les vecteurs associés à des chaînes de G .

Théorème 2.2 (Camion (1963)) Toute matrice de chaînes définies sur un arbre orienté est totalement unimodulaire.

Cette condition de totale unimodularité n'est que suffisante. Elle s'avère néanmoins couvrir un très grand nombre de matrices totalement unimodulaires, comme le montre Seymour (1980).

Les matrices de chaînes définies sur des arbres ont également été introduites sous le terme *matrices de réseau* (voir, par exemple, Tutte (1965); Schrijver (1986); Nemhauser et Wolsey (1988)).

Dans la mesure où les matrices de vacations sont des matrices 0/1, nous exploitons le résultat précédent dans le cas particulier des matrices de *chemins* définis sur des arbres orientés.

2.2 Graphes générateurs et matrices de vacations

Afin de décrire un ensemble de vacations, on introduit le concept de *graphe générateur*.

Définition 2.7 Considérant $I = \{1, \dots, m\}$ l'ensemble des périodes de temps à couvrir, on appelle graphe générateur (sous forme développée) un graphe orienté

$G = (X, U \cup U')$ où :

- $X = \{t_i, f_i, i \in I\}$ représente l'ensemble des débuts et fins des périodes de I ,
- $U = \{u_i = (t_i, f_i), i \in I\}$ représente l'ensemble des périodes de travail (arcs de période),
- U' est un sous-ensemble d'arcs de type $u'_{ij} = (f_i, t_j), i \in I, j \in I$ représentant les successions possibles entre périodes (arcs de succession).

Dans un graphe générateur, tout chemin μ décrit une vacation, admissible ou non, dont les périodes travaillées correspondent aux arcs de période appartenant à μ . Tout ensemble de chemins définis sur un graphe générateur décrit donc un ensemble de vacations.

La figure 1 présente un exemple de graphe générateur pour 10 périodes de travail où des pauses de durée une ou deux périodes peuvent être placées entre la seconde et la neuvième période.

Par exemple, le chemin $\mu = (u_2, u'_{2,3}, u_3, u'_{3,6}, u_6, u'_{6,7}, u_7, u'_{7,8}, u_8)$ défini sur ce graphe décrit la vacation telle que les périodes 2, 3, 6, 7 et 8 sont travaillées et les périodes 4 et 5 sont en pause.

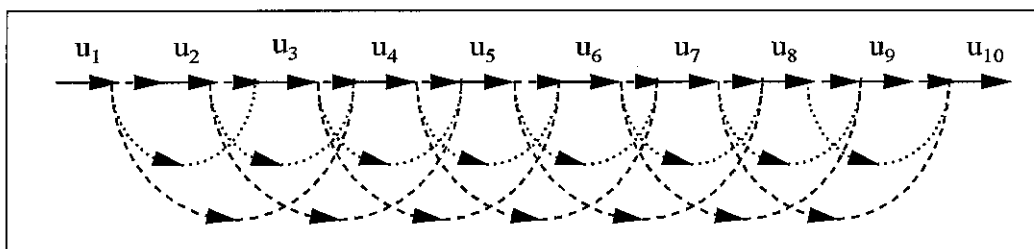


FIG. 1 - Graphe générateur pour 10 périodes et pauses de durée 1 ou 2 périodes

On notera l'analogie de représentation entre un graphe générateur et un graphe potentiels-étapes modélisant un problème central d'ordonnancement (voir, par exemple, Roy (1970); Carlier et Chrétienne (1988)). En effet, les arcs de U , qui traduisent les périodes de travail, s'assimilent aux tâches à réaliser de la modélisation potentiels-étapes; les arcs de U' , qui représentent les successions possibles entre périodes, s'assimilent aux tâches fictives, qui modélisent les contraintes de successions entre tâches réelles.

De même que la modélisation potentiels-étapes autorise la contraction de certaines tâches fictives dès lors que la représentation des successions entre tâches n'est pas altérée, on pourra dans certains cas contracter certains arcs de U' aboutissant ainsi à un graphe générateur *sous forme réduite*. Il en va ainsi par exemple lorsque le graphe générateur sous forme développée contient l'arc u'_{ij} et qu'il

n'existe aucun autre chemin reliant f_i à t_j . On peut alors, sans perte d'information, contracter u'_{ij} reliant ainsi directement u_i et u_j (voir figure 2). Après cette contraction, le sommet extrémité terminale de u_i et extrémité initiale de u_j sera appelé indifféremment f_i ou t_j .

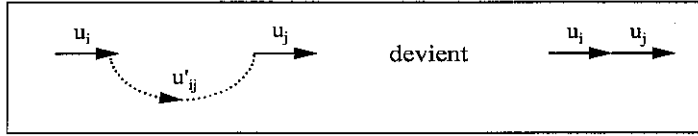


FIG. 2 – Exemple de contraction

Notons que ce type de contraction préserve toutes les successions possibles mais peut aussi en créer de nouvelles (voir figure 3).

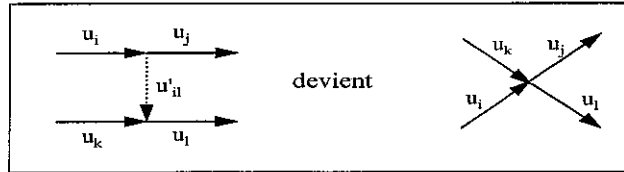


FIG. 3 – Exemple de contraction générant de nouvelles successions

A l'inverse de la modélisation potentiels-étapes, où la création de toute contrainte additionnelle est à proscrire, la création de nouvelles successions conduit ici à des possibilités supplémentaires d'enchaînement entre périodes (possibilités que l'on peut exploiter ou non pour générer des vacances supplémentaires).

On appellera graphe générateur *totalelement réduit* un graphe générateur tel que $U' = \emptyset$. Tout graphe générateur n'admet pas nécessairement de représentation totalement réduite. On notera \tilde{G} la représentation totalement réduite de G lorsqu'elle existe.

Montrons maintenant comment construire une matrice de vacances à partir d'un graphe générateur G . Il suffit pour cela de construire une matrice C de chemins définis sur G . Celle-ci peut s'exprimer, après permutation éventuelle des lignes, sous la forme :

$$C = \begin{pmatrix} A \\ A' \end{pmatrix} \quad (1)$$

où la sous-matrice A correspond aux arcs de U (arcs de période) et la sous-matrice A' correspond aux arcs de U' (arcs de succession). La matrice A est une matrice de vacances qui s'obtient donc à partir de C en supprimant les lignes correspondant aux arcs de succession. Notons finalement que si G est un graphe générateur totalement réduit, toute matrice de chemins définis sur G correspond à une matrice de vacances.

2.3 Interprétation temporelle des graphes générateurs

La définition du concept de graphe générateur ne conduit pas en elle-même à interpréter de façon temporelle les successions rendues possibles par les arcs de U' . L'existence d'un arc u'_{ij} indique simplement que les périodes i et j peuvent être travaillées au sein d'une même vacation. On peut donc avoir soit $i < j$, soit $i > j$. L'interprétation temporelle, qui consiste à considérer que l'existence d'un arc u'_{ij} indique que les périodes i et j vont se succéder temporellement, oblige à avoir $i < j$. Notons que le recours à l'interprétation temporelle entraîne également que tout chemin défini sur un graphe générateur fait apparaître les arcs de période par indice croissant.

Cette interprétation temporelle apparaît fort naturelle et aide grandement à construire un graphe générateur. Il importe cependant de déterminer si le respect de cette interprétation est restrictif au sens où cela ne permettrait plus de représenter certaines vacations.

Proposition 2.3 *Tout ensemble de vacations peut être décrit par un graphe générateur respectant l'interprétation temporelle.*

Démonstration: Soit G le graphe générateur complet tel que :

$$U' = \{u'_{ij} = (f_i, t_j), i = 1, \dots, m-1, j = i+1, \dots, m\} \quad (2)$$

La définition de U' entraîne que tout arc u_j situé postérieurement à un arc u_i sur un chemin quelconque de G est tel que $j > i$. G respecte donc l'interprétation temporelle.

G autorise toutes les successions temporelles possibles entre périodes et permet donc de décrire toutes les vacations envisageables sur un horizon de travail comportant m périodes. \square

Notons que le graphe générateur complet est sans circuit. Sa représentation par rangs croissants des sommets correspond à une représentation où les arcs de période u_i se succèdent par indice croissant le long d'un chemin principal. Nous appellerons cette représentation, *représentation linéaire* d'un graphe générateur.

Considérant un ensemble J de vacations à décrire, on peut soit utiliser le graphe générateur complet, soit un graphe partiel de ce graphe complet omettant ceux des arcs de U' qui traduisent les successions qui ne se produisent jamais dans J (où U' est défini par (2)). Ce second graphe étant un graphe partiel du graphe générateur complet, il peut également être représenté de façon linéaire (voir par exemple le graphe présenté à la figure 1 où les 10 arcs de période se succèdent par indice croissant le long du chemin principal horizontal).

Le recours à une interprétation temporelle présente donc l'avantage considérable de pouvoir construire un graphe générateur de façon intuitive et selon une structure relativement simple. Nous l'utiliserons donc chaque fois que possible.

Il convient néanmoins de souligner que le caractère non restrictif de cette interprétation temporelle et de la représentation linéaire qui s'ensuit provient de la grande flexibilité que fournissent les arcs de U' . On notera en particulier que tout arc de type $u'_{i,i+k}$ où $k \in \{2, \dots, m-i\}$ crée un cycle dans le graphe générateur. Or, nous serons amenés à nous intéresser dans la section suivante à des graphes générateurs sans cycle (en particulier à des arbres générateurs). On constatera alors qu'il convient d'abandonner l'interprétation temporelle et la représentation linéaire associée si l'on souhaite pouvoir représenter, dans le cadre restreint des arbres générateurs, certains ensembles de vacances (voir sous-section 3.3, cas avec plage de pause non imposée et cas avec fenêtre de pause).

3 Une classe de PCV polynomiaux

3.1 Une classe générale de matrices de vacances totalement unimodulaires

On identifie ici une classe de PCV, caractérisée par les types de vacances admissibles, telle que, quel que soit le coût des vacances et le niveau de la charge, la résolution puisse se faire en temps polynomial. Nous proposons à cet effet les résultats suivants.

Proposition 3.1 *Toute matrice de vacances construite sur un graphe générateur G qui est un arbre est totalement unimodulaire.*

Démonstration: Toute matrice de vacances A construite sur G est obtenue à partir d'une matrice de chemins C comme indiqué en (1). G étant un arbre, on sait d'après le théorème 2.2 que C est totalement unimodulaire. Toute ligne de la sous-matrice A' peut être éliminée en préservant la totale unimodularité. En effet, chaque suppression de ligne correspond à la contraction d'un arc dans G . Or cette opération de contraction préserve la connexité et ne crée pas de cycle. Le graphe réduit obtenu reste donc un arbre. Toutes les lignes de A' ayant été éliminées de C , A apparaît comme une matrice de chemins définis sur un arbre générateur (correspondant à \tilde{G}). \square

Nous déduisons de cette proposition le résultat général suivant :

Théorème 3.2 *Le problème $\min\{cx : Ax \geq b, x \in \mathbb{R}^n\}$ où A est une matrice de vacances construite sur un arbre générateur admet une solution optimale entière pour tout vecteur b entier et tout vecteur c réel.*

Démonstration: Immédiat d'après la proposition précédente et le théorème 2.1. \square

Nous identifions ainsi une classe relativement générale de PVC dont la résolution peut se faire efficacement à l'aide d'un algorithme de type simplicial. Nous montrons à la section 4 que tous ces problèmes peuvent également être formulés comme des problèmes de flot à coût minimal. Nous venons donc de caractériser une classe de PCV polynomiaux dont nous allons examiner quelques cas particuliers.

Notons auparavant que, d'après la démonstration de la proposition précédente, tout graphe générateur G qui est un arbre admet une représentation totalement réduite \tilde{G} . Il est alors plus aisé de travailler sur \tilde{G} car toute matrice de vacations apparaît alors directement comme une matrice de chemins définis sur \tilde{G} .

3.2 Le cas des vacations sans pause

L'arbre générateur le plus simple que l'on puisse envisager est celui composé de tous les arcs de périodes et uniquement des arcs de succession directe.

Considérons donc $G_1 = (X, U \cup U')$ où :

- $X = \{t_i, f_i, i \in \{1, \dots, m\}\}$,
- $U = \{u_i, i \in \{1, \dots, m\}\}$,
- $U' = \{u'_{i,i+1} = (f_i, t_{i+1}), i \in \{1, \dots, m-1\}\}$.

G_1 correspond à un graphe générateur de vacations sans pause. \tilde{G}_1 est obtenu en contractant tous les arcs de U' . Pour ne générer que des vacations de durée admissible, il suffit de considérer dans \tilde{G}_1 les chemins de longueur correspondante.

La figure 4 représente le graphe \tilde{G}_1 pour $m = 12$ périodes. La matrice de vacations générée à partir de \tilde{G}_1 pour des vacations de durée 5 ou 6 périodes correspond à la matrice des chemins de longueur 5 et 6 définis sur \tilde{G}_1 . Elle est représentée par le tableau 1.

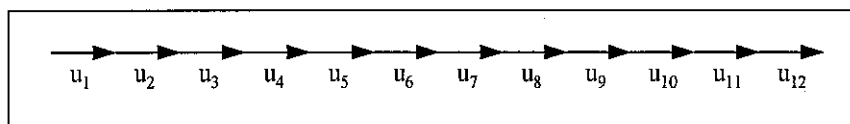


FIG. 4 - \tilde{G}_1 : Graphe générateur des vacations sans pauses

\tilde{G}_1 étant un arbre, on en déduit la proposition suivante :

Proposition 3.3 *Le PCV sans pause est un problème polynomial.*

Dans ce cas particulier des vacations sans pause, ce résultat peut être établi plus simplement en observant que la matrice de vacations sans pause est une *matrice d'intervalle* (chaque colonne comporte une succession de 0, une succession ininterrompue de 1 et une succession de 0). En effet, il est bien connu que les matrices d'intervalles sont totalement unimodulaires.

vacations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
période 1	1	1													
période 2	1	1	1	1											
période 3	1	1	1	1	1	1									
période 4	1	1	1	1	1	1	1	1							
période 5	1	1	1	1	1	1	1	1	1	1					
période 6		1	1	1	1	1	1	1	1	1	1	1			
période 7				1	1	1	1	1	1	1	1	1	1	1	
période 8						1	1	1	1	1	1	1	1	1	1
période 9								1	1	1	1	1	1	1	1
période 10										1	1	1	1	1	1
période 11												1	1	1	1
période 12														1	1

Les « 1 » correspondent aux périodes travaillées, les blancs aux périodes de repos.

TAB. 1 – Matrice de vacations de durées 5 et 6 construite sur \widetilde{G}_1

3.3 Le cas des vacations avec pauses

Il résulte de ce qui précède que la difficulté du PCV provient de la présence de pauses. Nous montrons maintenant que certaines classes de PCV demeurent polynomiales même si l'ensemble des vacations admissibles comporte des pauses.

Introduisons préalablement une distinction concernant les périodes susceptibles de donner lieu à pause.

Définition 3.1 Nous appellerons *plage de pause* un sous-ensemble de périodes consécutives devant donner lieu à pause pour toute vacation incluant des périodes travaillées avant et après la plage. Néanmoins, lorsque la plage de pause n'est pas imposée, on pourra autoriser l'existence de vacations commençant ou terminant pendant la plage de pause.

Nous appellerons *fenêtre de pause* un sous-ensemble de périodes consécutives pouvant donner lieu à pause.

Dans la suite, par souci de simplicité, nous nous limiterons à des cas avec plage ou fenêtre de pause unique. L'extension à plages ou fenêtres multiples ne comporte pas de difficulté particulière.

3.3.1 Cas avec plage de pause imposée

On considère ici qu'aucune vacation ne peut comporter de période travaillée durant la plage de pause. Même si ce cas a une certaine portée pratique, il convient de noter que, pour que le PCV ait un sens, il faut nécessairement que la charge soit nulle pendant la plage de pause. Soit $i^*, i^* + 1, \dots, i^* + k - 1$ les k périodes consécutives correspondant à la plage.

Considérons $I = \{1, \dots, i^* - 1, i^* + k, \dots, m\}$ l'ensemble des périodes privé des périodes de la plage de pause et le graphe $G_2 = (X, U \cup U')$ défini par :

$$- X = \{t_i, f_i, i \in I\},$$

$$- U = \{u_i, i \in I\},$$

$$- U' = \{u'_{i,i+1}, i \in \{1, \dots, i^* - 2\} \cup \{i^* + k, \dots, m - 1\}\} \cup \{u'_{i^*-1, i^*+k}\}.$$

G_2 correspond à un graphe générateur de vacances avec plage de pause imposée. \widetilde{G}_2 est obtenu en contractant tous les arcs de U' .

La figure 5 représente \widetilde{G}_2 pour $m = 12$ périodes, avec une plage de pause incluant les périodes 5 à 9. La matrice de vacances générée à partir de \widetilde{G}_2 pour des vacances de durée 5 ou 6 périodes est représentée par le tableau 2. Notons que pour un nombre supérieur de périodes, il pourrait exister des vacances sans pause qui se terminent avant la plage de pause ou commencent après.

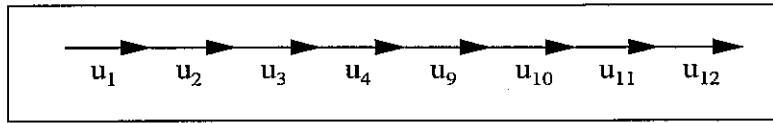


FIG. 5 – \widetilde{G}_2 : Graphe générateur avec plage de pause imposée

vacations	1	2	3	4	5	6	7
période 1	1	1					
période 2	1	1	1	1			
période 3	1	1	1	1	1	1	
période 4	1	1	1	1	1	1	1
période 5							
période 6							
période 7							
période 8							
période 9	1	1	1	1	1	1	1
période 10		1	1	1	1	1	1
période 11				1	1	1	1
période 12						1	1

TAB. 2 – Matrice associée au graphe \widetilde{G}_2

\widetilde{G}_2 est un arbre donc toute matrice de vacances construite sur \widetilde{G}_2 est totalement unimodulaire.

On en déduit la proposition suivante :

Proposition 3.4 *Le PCV avec plages de pause imposées est un problème polynomial.*

Ce résultat pouvait aussi être établi simplement à partir du précédent en considérant la matrice de vacances comme une matrice d'intervalles à laquelle on aurait ajouté des lignes de 0. Il est en effet évident que l'ajout de lignes de 0 ne perturbe pas la totale unimodularité (toute nouvelle sous-matrice résultant de cet ajout comporte une ligne de 0 et a donc un déterminant nul).

Notre cadre conceptuel, fondé sur l'utilisation de graphes générateurs et du théorème 3.2, permet également de caractériser des matrices de vacations plus générales qui sont aussi totalement unimodulaires. Nous en proposons deux exemples particuliers, le premier faisant intervenir une plage de pause, le second une fenêtre de pause.

Ces deux exemples sont construits selon un même procédé.

3.3.2 Un procédé typique de construction d'arbres générateurs

Quelles que soient les contraintes pratiques relatives à la construction d'un ensemble de vacations (durées et heures de début possibles des vacations, durées et heures de début possibles des pauses), on peut toujours bâtir un graphe générateur permettant de décrire toutes les vacations respectant ces contraintes. Dans la mesure où nous considérons ici uniquement des *arbres* générateurs, il n'est pas toujours possible de respecter toutes ces contraintes (en particulier celles afférentes au positionnement des pauses).

Pour construire un arbre générateur, on identifie d'abord les périodes qui ne peuvent donner lieu à pause et on enchaîne les arcs correspondant à ces périodes, définissant ainsi un chemin unique de la première à la dernière période travaillée. On raccorde ensuite à ce chemin les arcs correspondant aux périodes pouvant donner lieu à pause de manière à ne pas créer de cycle. Le choix des sommets de raccordement est à déterminer de manière à s'ajuster le mieux possible aux contraintes de définition des pauses.

On génère alors les vacations autorisées pour chaque durée d en cherchant sur le graphe les chemins de longueur d .

3.3.3 Un cas particulier avec plage de pause non imposée

Considérons à titre d'exemple le graphe générateur \tilde{G}_3 comportant une plage de pause composée des périodes 5 à 8 (voir figure 6).

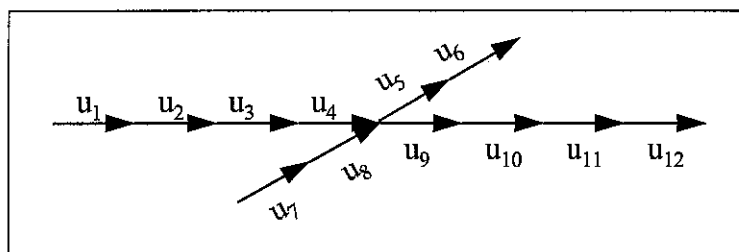


FIG. 6 – \tilde{G}_3 : Graphe générateur de vacations avec plage de pause

Toute vacation construite sur ce graphe qui travaille avant et après la plage de pause ne peut travailler pendant. En effet, tout chemin comportant les arcs u_4 et u_9 ne peut comporter aucun des arcs u_5 , u_6 , u_7 ou u_8 . En revanche, les

vacations qui travaillent pendant les premières périodes peuvent aussi travailler en 5 et éventuellement en 6 et s'arrêtent ensuite. De même, les vacations travaillant uniquement après la plage de pause peuvent commencer pendant la plage dès la période 8 ou même dès la période 7.

Remarquons qu'il serait aussi possible de générer des vacations travaillant uniquement pendant les périodes de la plage de pause, ce qui correspond ici au chemin (u_7, u_8, u_5, u_6) (on notera la succession u_8, u_5 non conforme à l'interprétation temporelle des successions).

La matrice des vacations de durées 5 ou 6 périodes construite sur le graphe \widetilde{G}_3 est représentée dans le tableau 3.

vacations	1	2	3	4	5	6	7	8	9	10	11	12	13
période 1	1	1	1	1									
période 2	1	1	1	1	1	1	1						
période 3	1	1	1	1	1	1	1	1	1				
période 4	1	1	1	1	1	1	1	1	1	1			
période 5	1	1			1								
période 6		1			1								
période 7											1	1	
période 8											1	1	1
période 9			1	1		1	1	1	1	1	1	1	1
période 10				1		1	1	1	1	1	1	1	1
période 11							1	1	1	1	1	1	1
période 12									1	1		1	1

TAB. 3 – Matrice associée au graphe \widetilde{G}_3 avec des vacations de durée 5 ou 6

Le type d'arbre que nous venons de construire illustre un procédé de portée relativement générale pour représenter des vacations avec plage de pause.

Soit $i^*, i^* + 1, \dots, i^* + k - 1$ les k périodes consécutives correspondant à la plage. Comme dans le cas avec plage de pause imposée, on construit tout d'abord le chemin incluant successivement les arcs $u_1, \dots, u_{i^*-1}, u_{i^*+k}, \dots, u_m$. Le sommet f_{i^*-1} , qui correspond également au sommet t_{i^*+k} , traduit à la fois le début et la fin de la plage de pause. Il suffit alors de raccorder à ce sommet, sans créer de cycle, tous les arcs u_j ($j = i^*, i^* + 1, \dots, i^* + k - 1$).

Il convient de noter que la description d'un tel procédé n'est possible qu'en raison d'une utilisation explicite de l'indigage des sommets et de l'interprétation temporelle des successions.

3.3.4 Un cas particulier avec fenêtres de pause

Considérons maintenant un exemple de graphe générateur avec une fenêtre de pause constituée des périodes 5 à 8 (voir figure 7).

Compte-tenu des sommets de raccordement aux périodes de pause, certaines vacations peuvent travailler pendant la fenêtre de pause ainsi qu'avant *et* après la fenêtre (à l'inverse du cas avec plage de pause), mais les vacations travaillant

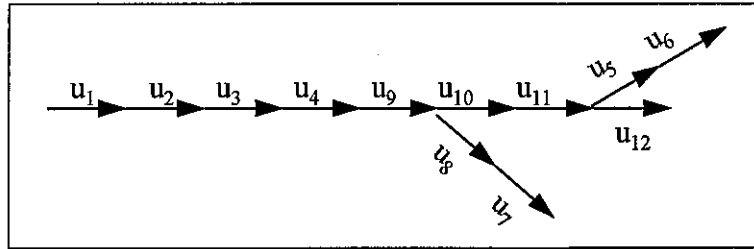


FIG. 7 - \widetilde{G}_4 : Graphe générateur de vacances avec fenêtre de pauses

pendant la période 8 (et éventuellement pendant la période 7) doivent nécessairement se terminer à la période 9. De même, les vacances travaillant pendant la période 5 (et éventuellement pendant la période 6) doivent nécessairement se terminer à la période 11.

La matrice des vacances de durées 5 ou 6 périodes construite sur le graphe \widetilde{G}_4 est représentée dans le tableau 4.

vacations	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
période 1	1	1	1												
période 2	1	1	1	1	1	1	1								
période 3	1	1	1	1	1	1	1	1	1	1	1				
période 4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
période 5											1		1	1	1
période 6														1	1
période 7							1			1					
période 8			1			1	1			1					
période 9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
période 10		1		1	1			1	1		1	1	1	1	1
période 11					1			1	1		1	1	1	1	1
période 12									1			1			

TAB. 4 - Matrice associée au graphe \widetilde{G}_4 avec des vacances de durée 5 ou 6

Nous constatons que la construction de ce type d'arbre requiert un positionnement relativement peu naturel des arcs traduisant les périodes de pause possible et contraint à abandonner l'interprétation temporelle des successions. C'est pourquoi cet exemple, au contraire du précédent, ne reflète qu'un cas spécifique de construction d'arbre générateur.

De nombreux autres ensembles de vacances définis avec fenêtres de pauses peuvent se présenter dans la pratique. Certains pourront être décrits par un arbre générateur, d'autres non. Dans la section suivante, nous présentons une autre modélisation basée sur les flots pour montrer que certains PCV, ne pouvant pas être décrits par un graphe générateur, sont néanmoins polynomiaux.

4 Une modélisation par les flots

Dans cette section, nous montrons que, sous certaines conditions, le PCV peut se formuler comme un problème de flot à coût minimal. Le réseau de transport qui supporte le flot est construit à partir du graphe générateur.

Considérons un graphe générateur tel qu'il a été introduit dans la section précédente. Comme nous l'avons vu, tout chemin sur ce graphe décrit une vacation, admissible ou non. A chaque vacation admissible commençant à la période i et finissant à la période k , est associé un « arc retour » v_{ki} allant de l'extrémité terminale de l'arc u_k à l'extrémité initiale de l'arc u_i .

Définition 4.1 *Considérant un graphe générateur $G = (X, U \cup U')$ et J un ensemble de vacances admissibles, on appelle graphe augmenté le graphe $\widehat{G} = (X, U \cup U' \cup V)$ où $V = \{v_{ki} = (f_k, t_i) \text{ tels qu'il existe au moins un chemin de } t_i \text{ à } f_k \text{ décrivant une vacation de } J\}$.*

La figure 8 représente le graphe augmenté associé au graphe générateur \widetilde{G}_3 et à l'ensemble des vacances du tableau 3. Les 13 arcs retour, représentés en traits pointillés, correspondent aux 13 colonnes de la matrice de vacances.

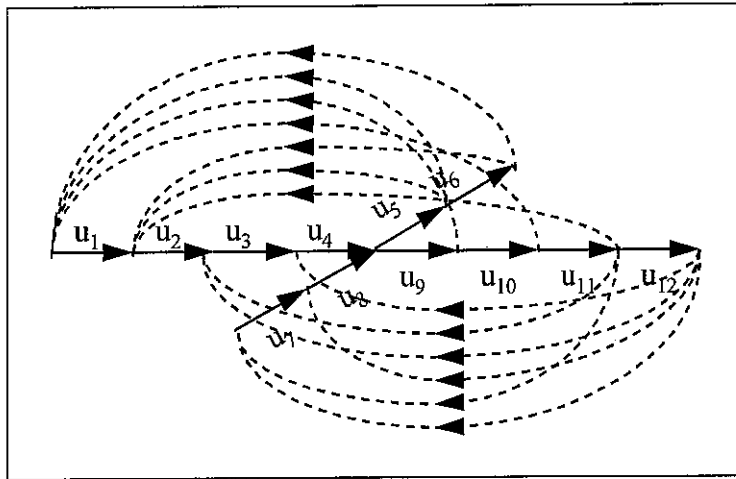


FIG. 8 - \widehat{G}_3 : Graphe augmenté du graphe générateur \widetilde{G}_3

Ce graphe augmenté définit un réseau de transport dont les arcs $u_i \in U$ sont valués par les bornes b_i ($i \in I$), correspondant à la charge à couvrir sur chaque période i (ces bornes sont donc entières). Les bornes des autres arcs sont nulles. Les capacités de tous les arcs sont infinies. Les coûts associés aux arcs seront définis différemment selon les cas (ils peuvent être réels).

4.1 Cas de réseaux issus d'arbres générateurs

Étudions d'abord le cas de graphes augmentés construits à partir d'arbres générateurs. On a les résultats suivants :

Proposition 4.1 *Tout graphe augmenté \hat{G} construit à partir d'un arbre générateur G en ajoutant p arcs retour crée p circuits élémentaires dans \hat{G} .*

Démonstration: Soit v_{k_1, i_1} le premier arc retour ajouté à G . Cet arc correspond à l'existence d'au moins un chemin de t_{i_1} à f_{k_1} . G étant un arbre, ce chemin est unique et l'on crée ainsi un circuit élémentaire unique.

On ajoute ensuite progressivement tous les arcs retour. Si l'ajout d'un arc retour v_{ki} conduit à ne créer que des circuits élémentaires incluant uniquement des arcs de G et aucun autre arc retour excepté v_{ki} , l'argument précédent demeure valable et, dans ce cas, on ne crée qu'un circuit élémentaire unique. Il suffit alors de montrer qu'aucun circuit élémentaire de \hat{G} ne peut inclure plus d'un arc retour. Supposons que \hat{G} contienne un circuit ω incluant plusieurs arcs retour. Par définition, chacun de ces arcs retour correspond dans G à un chemin de sens contraire avec les mêmes extrémités. Associons à ω défini sur \hat{G} un parcours dans G obtenu en conservant les arcs de $(U \cup U') \cap \omega$ et en remplaçant chacun des arcs retour de $V \cap \omega$ par le chemin de sens contraire qui lui correspond. Le parcours obtenu dans G est un cycle, ce qui contredit le fait que G est un arbre. \square

Théorème 4.2 *Tout PCV dont l'ensemble des vacations peut être décrit sur un arbre générateur peut se formuler comme un problème de flot à coût minimal.*

Démonstration: D'après la proposition précédente, tout arc retour v_{ki} est en bijection avec la vacation unique représentée par le chemin d'extrémité initiale t_i et d'extrémité terminale f_k . Il suffit alors d'associer à chaque arc v_{ki} le coût de la vacation correspondante c_{ki} et de chercher un flot de coût minimal sur \hat{G} . La valeur du flux sur chaque arc retour indique le nombre de vacations de type correspondant, utilisées dans la solution optimale. \square

Tous les cas de PCV que nous avons caractérisés dans la section précédente peuvent donc être reformulés comme des problèmes de flot à coût minimal (ou plus précisément des problèmes de circulation à coût minimal). Cette modélisation avait déjà été proposée par Segal (1974) dans le cas particulier des vacations sans pause. Nous l'avons généralisée aux ensembles de vacations pouvant être construits sur des arbres générateurs.

4.2 Une classe étendue de PCV polynomiaux

La modélisation par les flots permet également de mettre en évidence des cas de PCV polynomiaux pour lesquels la matrice de vacations n'est pas nécessairement totalement unimodulaire.

Nous travaillerons sur des réseaux issus de graphes générateurs quelconques. Nous considérerons donc, dans ce qui suit, que ces graphes générateurs respectent l'interprétation temporelle et sont représentés de façon linéaire (ce qui n'est pas restrictif comme nous l'avons montré en section 2.3).

Soit \hat{G} le graphe augmenté d'un graphe générateur quelconque G . Un arc retour v_{ki} , qui correspond à au moins une vacation admissible, peut ici faire partie de plusieurs circuits car il peut exister dans G plusieurs chemins de t_i à f_k . A la différence du cas précédent, un arc retour peut donc correspondre à plusieurs vacations.

En conséquence, il n'est plus possible de représenter des coûts de vacation quelconques. Plusieurs structures de coût particulières peuvent néanmoins être modélisées :

- (cas 1) le coût c_j d'une vacation j dépend uniquement de la première période travaillée i et de la dernière période travaillée k ; il convient alors d'associer un coût c_j à chaque arc retour v_{ki} ;
- (cas 2) le coût c_j d'une vacation j s'exprime comme la somme des coûts des périodes travaillées durant la vacation; il convient alors d'associer un coût q_i à chaque arc de période u_i pour avoir $c_j = \sum_{i=1}^m q_i a_{ij}$ ($j = 1, \dots, n$).

La portée pratique du cas 1 peut paraître limitée. Notons néanmoins que le cas particulier consistant à affecter un coût identique à chaque arc retour revient à minimiser le nombre de vacations sélectionnées. Cet objectif a d'ailleurs été envisagé dans divers travaux (voir, par exemple, Dantzig (1954); Morris et Showalter (1983)).

Le cas 2 n'est guère restrictif en pratique. En effet, on considère même le plus souvent un coût de vacation proportionnel à la durée travaillée, ce qui correspond au cas particulier où $q_i = q$ ($i = 1, \dots, m$).

On a les résultats suivants :

Proposition 4.3 *Tout circuit élémentaire de \hat{G} , où G est un graphe générateur représenté de façon linéaire, contient un arc retour unique.*

Démonstration: G , étant représenté de façon linéaire, ne contient pas de circuit. L'apparition de circuits dans \hat{G} provient de l'ajout des arcs retour. Tout circuit de \hat{G} comporte donc au moins un arc retour.

Supposons que \hat{G} contienne un circuit élémentaire incluant p (> 1) arcs retour $v_{k_1, i_1}, \dots, v_{k_p, i_p}$, où k_1 est l'indice le plus élevé (voir figure 9). Parcourons le circuit à partir du sommet f_{k_1} . Nous empruntons l'arc retour v_{k_1, i_1} pour atteindre le sommet t_{i_1} . On emprunte alors nécessairement une portion (éventuellement vide) d'un chemin de t_{i_1} à f_{k_1} en s'arrêtant obligatoirement avant f_{k_1} sinon le circuit ne

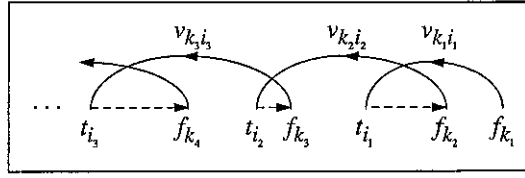


FIG. 9 - Parcours élémentaire sur \widehat{G}

serait pas élémentaire. Soit f_{k_2} , avec $k_2 < k_1$ le sommet ainsi atteint (correspondant éventuellement à t_{i_1} si la portion de chemin est vide). On emprunte alors un nouvel arc retour v_{k_2, i_2} et procédant comme ci-dessus on atteint un sommet f_{k_3} , avec $k_3 < k_2$. On parcourt ainsi un chemin élémentaire qui ne permettra jamais de revenir en f_{k_1} . Tout circuit de \widehat{G} comporte donc au plus un arc retour. \square

Le résultat précédent montre que tout circuit élémentaire de \widehat{G} est composé d'un arc retour de type v_{k_i} et d'un chemin de t_i à f_k . Tout circuit élémentaire est donc en bijection avec la vacation correspondant au chemin associé à chaque circuit.

Théorème 4.4 *Si le coût de chaque vacation dépend uniquement des première et dernière périodes travaillées (cas 1) ou s'exprime comme la somme des coûts des périodes qu'elle couvre (cas 2) et si tout circuit élémentaire du graphe augmenté \widehat{G} correspond à une vacation admissible, alors le PCV peut se formuler comme un problème de flot à coût minimal sur \widehat{G} .*

Démonstration: Associons une borne b_i à chaque arc u_i , ($i = 1, \dots, m$) et associons les coûts aux arcs retour (cas 1) ou aux arcs de période (cas 2) comme indiqué auparavant. Déterminons ensuite un flot de coût minimal sur \widehat{G} . Pour obtenir une solution optimale du PCV, il convient de décomposer le flot optimal sur les circuits élémentaires de \widehat{G} ce qui est toujours possible pour un flot à composantes positives ou nulles qui en fait une circulation (voir théorèmes de décomposition d'un flot dans Roy (1970); Christofides (1975); Ahuja *et al.* (1993) ou Gondran et Minoux (1995)). Les bornes étant entières, le flot optimal sera également entier, et il est possible de le décomposer en flots entiers sur les circuits élémentaires. Si tout circuit élémentaire de \widehat{G} correspond à une vacation admissible, on est assuré de faire correspondre au flot une solution du PCV. \square

Il est important de souligner que la décomposition du flot sur les circuits n'est en général pas unique. Le flot optimal représente donc plusieurs solutions optimales. On pourra donc orienter la procédure de décomposition de manière à privilégier la présence de certaines vacations dans la solution optimale. Il s'agit là d'un avantage spécifique de cette modélisation.

Notons que la restriction la plus importante de notre résultat provient de ce que l'on impose que tout circuit élémentaire doit correspondre à une vacation

admissible. En effet, si tel n'est pas le cas, on ne peut garantir que le flot optimal sur \hat{G} pourra se décomposer sur les seuls circuits élémentaires correspondant à des vacances admissibles. Néanmoins, le coût du flot optimal constitue une borne inférieure du coût de la solution optimale du PVC. De plus et surtout, il peut être intéressant de décomposer le flot en utilisant prioritairement les circuits correspondant à des vacances admissibles. Si le flot résiduel est de valeur faible, on peut soit accepter dans la solution la présence de quelques vacances initialement interdites soit accepter une sous-couverture de la charge.

4.3 Un exemple illustratif

A titre d'exemple considérons le graphe augmenté de la figure 10. Ce graphe permet de décrire 10 vacances (voir tableau 5). Les 5 premières vacances sont associées à l'arc retour v_1 , les 5 suivantes à l'arc retour v_2 .

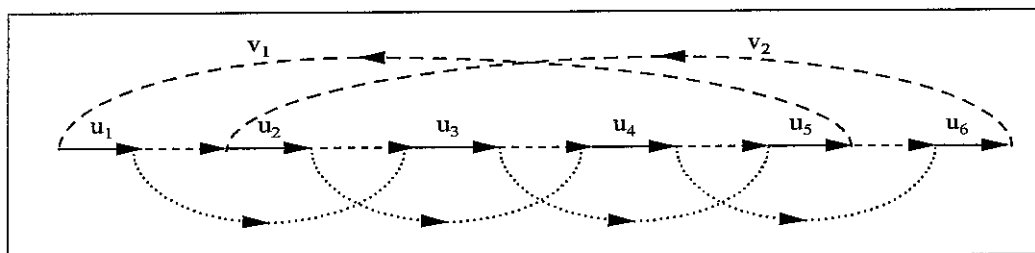


FIG. 10 – Graphe augmenté sur 6 périodes

vacations	1	2	3	4	5	6	7	8	9	10
période 1	1	1	1	1	1	0	0	0	0	0
période 2	1	0	1	1	0	1	1	1	1	1
période 3	1	1	0	1	1	1	0	1	1	0
période 4	1	1	1	0	0	1	1	0	1	1
période 5	1	1	1	1	1	1	1	1	0	0
période 6	0	0	0	0	0	1	1	1	1	1

TAB. 5 – Matrice associée au graphe augmenté

On constate tout d'abord que la matrice des vacances n'est pas totalement unimodulaire (la sous-matrice encadrée a un déterminant égal à 2).

Supposons ici que les coûts de vacances correspondent à leurs durées (cas 2). On associe donc à chaque arc u_i un coût $q_i = 1$ ainsi que des bornes b_i correspondant à la charge requise en période i ($i = 1, \dots, m$):

u_i	u_1	u_2	u_3	u_4	u_5	u_6
b_i	5	8	10	12	8	5

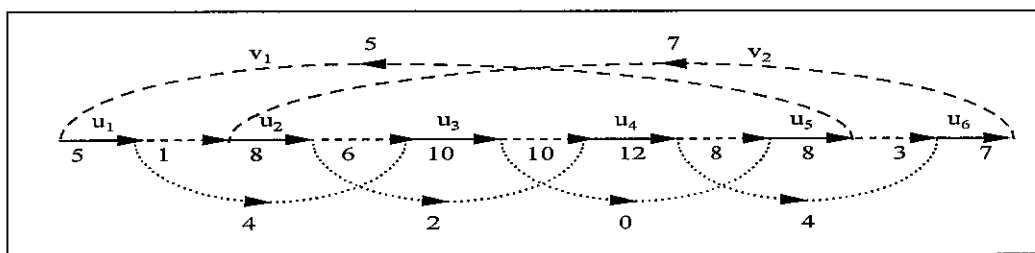


FIG. 11 – Flot de coût minimal

Le flot optimal, de coût 50, est présenté figure 11. Il fait intervenir 12 vacances (somme des flux sur les arcs retour).

Ce flot peut se décomposer de 4 façons, conduisant aux solutions optimales suivantes :

nb de vacances	1	2	3	4	5	6	7	8	9	10
solution 1	1	4				1	2		4	
solution 2	1	4				3			2	2
solution 3		4	1			2	1		4	
solution 4		4	1			3			3	1

Références

- Ahuja, R. K., Magnanti, T. L., et Orlin, J. B. (1993). *Networks Flows*. Prentice Hall, New Jersey.
- Bechtold, S. E. et Jacobs, L. W. (1991). Improvement of labor utilization in shift scheduling for services with implicit optimal modeling. *International Journal of Operations and Production Management*, **11**(2), 54-69.
- Camion, P. (1963). *Matrices Totalement Unimodulaires et Problèmes Combinatoires*. Ph.D. thesis, Université de Bruxelles.
- Carlier, J. et Chrétienne, P. (1988). *Problèmes d'ordonnancement - modélisation, complexité, algorithmes*. Masson, Paris.
- Christofides, N. (1975). *Graph Theory*. Academic Press, London.
- Dantzig, G. B. (1954). A comment on Edie's traffic delay at toll booths. *Operations Research*, **2**, 339-341.
- Gondran, M. (1973). Matrices totalement unimodulaires. *E.D.F. Bulletin de la Direction des Etudes et Recherches*, **1**, 55-74.
- Gondran, M. et Minoux, M. (1995). *Graphes et algorithmes*. Eyrolles, Paris, third edition.

- Henderson, W. B. et Berry, W. L. (1976). Heuristic methods for telephone operator shift scheduling: An experimental analysis. *Management Science*, **22**, 1372–1380.
- Hoffman, A. J. et Kruskal, J. B. (1956). Integral boundary points of convex polyhedra. In H. W. Kuhn et A. W. Tucker, editors, *Linear Inequalities and Related Systems, Annals of Mathematics Studies*, volume 38, pages 223–246. Princeton University Press.
- Morris, J. G. et Showalter, M. J. (1983). Simple approaches to shift, days-off and tour scheduling problems. *Management Science*, **29**(8), 942–950.
- Nemhauser, G. L. et Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. Wiley.
- Padberg, M. W. (1976). A note on the total unimodularity of matrices. *Discrete Mathematics*, **14**, 273–278.
- Partouche, A. (1998). *Planification d'horaires de travail: Méthodologie, modélisation et résolution à l'aide de la Programmation Linéaire en Nombres Entiers et de la Programmation Par Contraintes*. Ph.D. thesis, Université Paris-Dauphine, Paris.
- Roy, B. (1970). *Algèbre moderne et théorie des graphes*. Dunod, Paris.
- Schrijver, A. (1986). *Linear and Integer Programming*. Wiley.
- Segal, M. (1974). The operator-scheduling problem: A network-flow approach. *Operations Research*, **22**, 808–823.
- Seymour, P. D. (1980). Decomposition of regular matroids. *Journal of Combinatorial Theory*, **B28**, 305–359.
- Tutte, W. T. (1965). Lectures on matroids. *Journal of Research of the National Bureau of Standards*, **69B**, 1–48.