

Une Approche pour l'accélération de la génération de colonnes appliquées au problème de rotations d'équipages

Abdelkader LAMAMRI⁽¹⁾, **Hacène AIT HADDADENE**⁽²⁾, **Anass NAGIH**⁽³⁾

⁽¹⁾Université de Blida, Algérie.

⁽²⁾Laboratoire LAID3, USTHB, Algérie.

⁽³⁾Laboratoire LITA, Université Paul Verlaine – Metz, France.

⁽¹⁾prslamamri_inch@yahoo.fr, ⁽²⁾aithaddadenehacene@yahoo.fr, ⁽³⁾anass.nagih@univ-metz.fr

Abstract: We are interested in problems from combinatorial optimization, more precisely, the problem of construction crew rotations with resource constraints. The problem is to cover the cost of all flights of the company. Given the large size of the problems encountered in industry, these models are solved by an approach based on column generation that can handle implicitly all feasible solutions and a master problem determining the best solution. We propose in this paper an approach to improve the acceleration of the method of column generation for solving the problem of construction crew rotations, it is projected in each arc, the resources a vector of size smaller by using a Lagrangean relaxation algorithm to determine the coefficients of the projection arc combined with an algorithm for re-optimization, then generates a sub-set of complementary solutions to the master problem. The preliminary experiments of our technique gave good results on instances of random rotation of crews.

Keywords: Combinatorial optimization, mathematical programming, column generation,

Résumé :

Nous nous sommes intéressés aux problèmes issus de l'optimisation combinatoire, plus précisément, le problème de construction des rotations d'équipages avec contraintes de ressources. Le problème consiste à couvrir au moindre coût l'ensemble des vols de la compagnie. Etant donné la grande taille des problèmes rencontrés dans l'industrie, ces modèles sont résolus par une approche basée sur la génération de colonnes qui permet de gérer implicitement l'ensemble des solutions réalisables et un problème maître déterminant la meilleure solution. Nous proposons dans cet article une approche permettant d'améliorer l'accélération de la méthode de la génération de colonnes pour la résolution de problème de construction des rotations d'équipages, il s'agit de projeter, en chaque arc, les ressources sur un vecteur de dimension inférieure en utilisant un algorithme de relaxation lagrangienne pour déterminer les coefficients de la projection par arc combinée à un algorithme de ré-optimisation, puis génère un sous-ensemble des solutions complémentaires vers le problème maître. Les expérimentations préliminaires de notre technique ont donné de bons résultats sur des instances aléatoires de rotations d'équipages.

Mots clés : Optimisation combinatoire, programmation mathématique, Génération de colonnes,

1. Introduction :

Dans l'industrie du transport aérien, l'optimisation et l'automatisation de la construction des rotations d'équipages est un enjeu financier et organisationnel majeur. Le problème consiste à couvrir au moindre coût l'ensemble des vols de la compagnie, programmés sur un horizon de temps donné, par des équipages formés de personnel de cockpit (pilotes, co-pilotes) et de personnel de cabine (hôtesses, stewards). À périodicité de plusieurs jours (de l'ordre de la semaine), chaque équipage part de la base à laquelle il est affecté, enchaîne un certain nombre de vols et revient à la base. Cette séquence de vols avec retour à la base est appelé *rotation*. La construction des rotations d'une compagnie aérienne est extrêmement contrainte par les réglementations internationale, nationale et interne du travail, et par la disponibilité limitée des ressources.

Ces contraintes rendent le problème particulièrement difficile à résoudre. L'utilisation de modèles et des logiciels d'optimisation pour ce problème permet aux grandes compagnies d'effectuer des gains financiers substantiels. Il n'est pas rare qu'une réduction d'un pourcent sur le coût total des rotations se traduise par plusieurs dizaines de millions de dollars d'économie pour les grandes compagnies [1], d'où une recherche, fondamentale et appliquée, foisonnante sur le sujet. Le problème général de Construction des Rotations avec Contraintes des Ressources (PCR-CR) peut se formuler comme un problème de flot réalisable à coût minimum dans un réseau multiple, avec ajout des variables et des contraintes de ressources. Notons enfin que les contraintes de ressources rendent le problème (PCR-CR) NP-difficile. Les modalités de construction du réseau des rotations admissibles, de calcul du coût des rotations ainsi que le Programme Mathématique associé sont présentées en section 2. La section 3 donne un aperçu sur les techniques classiques de résolution, notamment la méthode de génération de colonnes, dont le sous-problème associé est traité dans cette section, en plus de notre contribution. La section 4 présente notre Algorithme. L'application fera sur des instances de rotations d'équipages qui seront présenté dans la section 5. La section 6 étudie le cas où la solution obtenue est fractionnaire. Enfin la section 7 conclut le document.

2. Présentation du problème

L'ensemble des vols à couvrir par les équipages est noté $V = \{1, \dots, n\}$. Le programme de vols et les horaires associés sont établis de façon quasi-certaine sur la période considérée, de l'ordre du mois ou de la semaine selon la taille de la compagnie.

Le terme de *vol* associé à chaque élément $i \in V$ est dans certains cas abusif, dans la mesure où i peut en réalité représenter une séquence de vols agrégée et insécable, i.e., ne pouvant être couverte que par un seul et même équipage dans sa totalité. Souvent également, la tâche à couvrir par un équipage n'est pas le seul vol mais un service de vol pouvant débuter avant et terminer après le vol proprement dit, pour pouvoir inclure le temps de préparation de l'avion et le temps d'accompagnement des passagers, par exemple. Cependant, nous maintiendrons cette terminologie de *vol*, par souci de lisibilité. On connaît pour chaque vol $i \in V$:

- (i) l'horaire de départ $t^{\rightarrow}(i)$,
- (ii) l'horaire d'arrivée $t^{\leftarrow}(i)$,
- (iii) l'aéroport de départ $a^{\rightarrow}(i)$,
- (iv) l'aéroport d'arrivée $a^{\leftarrow}(i)$.

Une rotation doit débuter et terminer à l'une des bases de la compagnie. L'ensemble \mathcal{B} des bases est généralement composé de grandes plates-formes d'interconnexion appelées *hubs*. Le problème de construction de rotations dans le transport aérien comporte le plus souvent des

contraintes de ressources sur les rotations. Afin de prendre en compte ces contraintes, valables pour chaque rotation prise individuellement, une modélisation classique associe à chaque équipage un sous-réseau construit de la façon suivante.

2.1 Construction des sous-réseaux

L'ensemble des équipages susceptibles d'être utilisés pour couvrir les vols de la compagnie est indicé par $k \in \mathcal{K} = \{1, \dots, K\}$. Pour $k \in \mathcal{K}$, $b^k \in \mathcal{B}$ désigne la base de départ et d'arrivée de l'équipage k . Un graphe $G^k = (X^k, A^k)$ est alors associé à l'équipage $k \in \mathcal{K}$, où X^k désigne l'ensemble des nœuds du réseau et A^k l'ensemble des arcs. L'ensemble X^k se décompose en trois sous-ensembles :

$$X^k = \{o^k\} \cup V^k \cup \{d^k\}$$

où, pour $k \in \mathcal{K}$:

- l'origine o^k (resp. la destination d^k) désigne la source (resp. le puits) du réseau G^k ,
- $V^k \subseteq V$ désigne l'ensemble des vols programmés par la compagnie pouvant être couverts par l'équipage k .

L'ensemble A^k des arcs du réseau se décompose de la façon suivante :

$$A^k = \mathcal{O}V^k \cup VV^k \cup V\mathcal{D}^k \cup \{(o^k, d^k)\}$$

Où

$$\begin{aligned} \mathcal{O}V^k &= \{(o^k, i) : i \in V^k, a^\rightarrow(i) = b^k\} \\ VV^k &\subseteq \mathcal{U}^k = \{(i, j) \in V^k \times V^k : a^\rightarrow(i) = a^\rightarrow(j), t^\rightarrow(j) \geq t^\rightarrow(i) + t_{min}(i, j)\} \\ V\mathcal{D}^k &= \{(i, d^k) : i \in V^k, a^\rightarrow(i) = b^k\} \end{aligned}$$

Le passage par l'arc (o^k, d^k) signifiera que l'équipage k ne sera pas utilisé. \mathcal{U}^k est l'ensemble des couples de vols (i, j) remplissant les conditions nécessaires évidentes rendant possible l'enchaînement de deux vols par le même équipage :

- (i) l'aéroport d'arrivée du vol i est l'aéroport de départ du vol j ,
- (ii) l'heure de départ du vol j est postérieure à l'heure d'arrivée du vol i , d'un écart supérieur ou égal à une valeur $t_{min}(i, j)$, fixée par la compagnie ou par les contraintes de transit de l'aéroport.

Généralement, $VV^k \neq \mathcal{U}^k$ car la réglementation du travail de la compagnie impose un certain nombre de contraintes supplémentaires (créneaux repas, pauses, découchés) restreignant les possibilités de connexion entre vols. De plus, des contraintes globales, variant d'une compagnie à l'autre, sont généralement associées à chaque rotation. Citant parmi l'ensemble des contraintes possibles les contraintes de borne suivantes :

- borne inférieure et/ou supérieure sur l'amplitude totale de la rotation,
- borne inférieure et/ou supérieure sur le temps de travail total (temps de travail total = temps de vol total + transferts + pauses légales)
- borne inférieure sur le nombre de jours de repos
- borne inférieure sur le nombre d'heures de repos quotidien
- borne supérieure sur le nombre de vols
- borne supérieure sur le nombre d'heures de travail consécutives.

Ces contraintes peuvent se modéliser par la donnée :

- d'un ensemble $Q = \{1, \dots, Q\}$ de ressources,

- de consommations des ressources $t_{ij}^{k,q}$ associées à chaque arc $(i,j) \in A^k$ et chaque ressource $q \in Q$,
- de seuils minimaux $a_i^{k,q}$ et maximaux $b_i^{k,q}$ de consommation de ressource $q \in Q$, à respecter pour chaque équipage $k \in \mathcal{K}$ et en chaque nœud i du réseau ; si les contraintes de ressources portent sur l'ensemble de la rotation, i.e., sur le seul nœud destination d et pas sur les nœuds intermédiaires, dans ce cas $a_i^{k,q} = 0$ et $b_i^{k,q} = \infty$ pour tout nœud $i \neq d$.

2.2 Coût des rotations

Le mode de calcul du coût d'une rotation est généralement complexe et varie selon les compagnies. Ce coût peut être une fonction non linéaire de plusieurs paramètres tels que la consommation de ressources, l'amplitude totale et le temps de vol total de la rotation [1][2]. Afin d'établir un modèle générique pour ce chapitre, nous considérerons que la fonction de coût est une approximation linéaire décomposable par équipage $k = 1, \dots, K$ et par arcs $(i,j) \in A^k$. Le coût de la rotation effectuée par l'équipage $k \in \mathcal{K}$ sera alors la somme des coûts c_{ij}^k associés aux arcs $(i,j) \in A^k$ qui composent cette rotation.

2.3 Modélisation mathématique

Le Problème de Construction des Rotations avec Contraintes des Ressources (PCRCR) peut se modéliser, si la fonction de coût est linéaire, par la Programmation Linéaire en variables mixtes. Nous avons un problème de flot réalisable à coût minimal sur l'ensemble des sous-réseaux, avec variables binaires de flot et variables continues de ressources (PCR-CR) :

$$(PCRCR) \equiv \begin{cases} \min \sum_{k=1}^K \sum_{(i,j) \in A^k} c_{ij}^k x_{ij}^k & (1) \\ \sum_{k=1}^K \sum_{j:(i,j) \in A^k} x_{ij}^k \geq 1 \text{ pour } i \in V = \{1, \dots, n\} & (2) \\ \sum_{i:(o^k,i) \in A^k} x_{oi}^k = 1 \text{ pour } k \in \mathcal{K} & (3) \\ \sum_{i:(i,d^k) \in A^k} x_{id}^k = 1 \text{ pour } k \in \mathcal{K} & (4) \\ \sum_{i:(i,j) \in A^k} x_{ij}^k = \sum_{l:(j,l) \in A^k} x_{jl}^k \text{ pour } j \in V^k & (5) \\ T_i^{k,q} + t_i^{k,q} - T_j^{k,q} \leq M(1 - x_{ij}^k) \text{ pour } (i,j) \in A^k, k \in \mathcal{K}, q \in Q & (6) \\ a_i^{k,q} \leq T_i^{k,q} \leq b_i^{k,q} \text{ pour } i \in V^k, k \in \mathcal{K}, q \in Q & (7) \\ x_{ij}^k \in \{0,1\}, T_i^{k,q} \geq 0 \text{ pour } (i,j) \in A^k, k \in \mathcal{K}, q \in Q & (8) \end{cases}$$

Les variables binaires x_{ij}^k indiquent si la rotation emprunte l'arc $(i,j) \in A^k$ (et donc enchaîne les vols i et j si $(i,j) \in VV^k$), tandis que les variables $T_i^{k,q}$ indiquent la consommation cumulée de chaque ressource q à chaque nœud i d'un réseau G^k .

L'objectif (1) minimise le coût total des rotations. Les contraintes (2) expriment la couverture de chaque vol par au moins un équipage si un seul équipage est autorisé par vol la contrainte est à l'égalité. Les contraintes (3 – 5) définissent une structure de chemin dans le sous-réseau G^k : passage d'un flux d'une unité (3 ou 4) et conservation du flux aux sommets (5). Les contraintes (6 – 7) sont les contraintes de ressources associées à chaque rotation. La contrainte (6), dans laquelle $M > 0$ est un paramètre très grand, peut aussi se trouver sous la forme non linéaire suivante :

$$x_{ij}^k (T_i^{k,q} + t_i^{k,q} - T_j^{k,q}) \leq 0 \quad (i, j) \in A^k, k \in \mathcal{K}, q \in \mathcal{Q} \quad (9)$$

L'inégalité dans (6) ou (9) stipule que l'attente est permise pour l'équipage, dans le cas contraire la contrainte s'écrit à l'égalité. Cette contrainte permet d'obtenir la consommation cumulée de ressource q au nœud j , puisqu'on a :

$$T_j^{k,q} = \max(a_i^{k,q}, T_i^{k,q} + t_i^{k,q})$$

Les contraintes (7) sont des contraintes de bornes aux noeuds du réseau (fenêtres de temps par exemple). Remarquons que les contraintes (3 – 7) sont des contraintes locales valables pour le seul sous-réseau G^k .

Seules les contraintes de couverture (2) sont des contraintes globales liant les K sous-réseaux. La relaxation de ces contraintes liantes et la décomposition du problème initial par sous-réseau sera donc une option de résolution intéressante. Notons enfin que les contraintes de ressources (6 – 7) rendent le problème (PCR-CR) NP-difficile. Même le problème de réalisabilité associé est NP-complet [10].

3. Approches de résolution

3.1 Principes de décomposition

On distingue deux types de contraintes dans le système (2) – (7) :

- (i) les contraintes de couverture (2), dites liantes ou globales, qui lient l'ensemble des équipages $k = 1, \dots, K$,
- (ii) les contraintes (3) – (7) propres à chaque équipage $k \in \{1, \dots, K\}$ et définissant un itinéraire légal.

La matrice associée aux contraintes (3) – (7) étant bloc-diagonale, et l'objectif (1) étant séparable (car linéaire), la résolution de la relaxation continue de ce modèle peut être basée sur la décomposition de Dantzig-Wolfe. Dans ce type de décomposition, les contraintes (3) – (7) définissent K sous-problèmes indépendants et les contraintes globales (2) sont conservées dans le problème maître. Dans un schéma de type génération de colonnes, il s'agit de résoudre alternativement le problème maître et les K sous-problèmes. Pour obtenir une solution entière, ce schéma peut être appliqué au niveau de chaque nœud de l'arbre de recherche. La difficulté majeure réside dans la résolution des sous-problèmes dont les espaces des états peuvent augmenter de façon exponentielle avec le nombre de ressources Q , rendant incontournable l'utilisation d'heuristiques. D'autre part, la convergence du schéma de génération de colonnes étant sensible à la qualité des solutions fournies par la résolution de ces sous-problèmes, la résolution effective d'instances réelles issues de l'industrie nécessite de trouver un bon compromis entre la qualité des solutions et le temps de résolution des sous-problèmes. Dans ce qui suit, nous détaillons le principe général de la génération de colonnes pour le problème(PCR-CR).

3.2 Génération de colonnes, problème maître et sous-problème

Les méthodes de génération de colonnes [3] ont été appliquées avec succès aux problèmes de construction de rotations [2], [5]. Dans cette approche, le problème maître est reformulé par un Problème de Couverture (PC) (*Set Covering* ou *Set Partitioning* selon que la contrainte de couverture des vols est une inégalité ou à l'égalité) :

$$(PC) \equiv \begin{cases} \min \sum_{r \in \mathcal{R}} c_r x_r & (10) \\ \sum_{r \in \mathcal{R}} a_{ir} x_r \geq 1 \text{ pour } i \in V = \{1, \dots, n\} & (11) \\ x_r \in \{0,1\} \text{ pour } r \in \mathcal{R} & (12) \end{cases}$$

Où \mathcal{R} désigne l'ensemble des rotations admissibles satisfaisant les contraintes de ressources et d'enchaînement entre vols, c_r représente le coût de la rotation $r \in \mathcal{R}$, $a_{ir} = 1$ si et seulement si la rotation r couvre le vol i , et la variable binaire x_r indique le choix ou non de la rotation r dans la solution.

On note (\overline{PC}) la relaxation continue du problème (PC) où les contraintes d'intégrité (12) sont remplacées par $x_r \geq 0$ pour $r \in \mathcal{R}$. Le nombre total de rotations admissibles $|\mathcal{R}|$ étant généralement une fonction exponentielle du nombre $n = |V|$ de vols à couvrir, l'énumération complète de \mathcal{R} est à proscrire. Pour autant, il est possible de trouver en un temps raisonnable une solution optimale de (\overline{PC}) en ne générant qu'un sous-ensemble restreint de rotations (i.e., de colonnes de la matrice de contraintes). Le principe est le suivant : Soit \mathcal{R}^0 une solution réalisable pour (PC), comprenant un nombre restreint de rotations de \mathcal{R} , et généré par une heuristique quelconque. Nous pouvons résoudre par la Programmation Linéaire (par exemple, par l'algorithme du Simplexe) le programme (\overline{PC}^0) , qui est la restriction de (\overline{PC}) au sous-ensemble de rotations \mathcal{R}^0 . Cette résolution fournit également un vecteur de multiplicateurs ou de variables duales $(\delta_1^0, \dots, \delta_n^0)$ associé aux n vols à couvrir. Le critère d'optimalité selon lequel toutes les rotations sont de coût réduit positif à l'optimum, conduit à rechercher la rotation de plus faible coût réduit négatif, soit

$$r^0 = \arg \min_{r \in \mathcal{R}} \left(c_r - \sum_{i=1}^n \delta_i^0 a_{ir} \right) \quad (13)$$

Si l'on parvient à trouver en temps raisonnable cette rotation r^0 , on peut alors relancer la résolution du programme de couverture (\overline{PC}) sur l'ensemble $\mathcal{R}^1 = \mathcal{R}^0 \cup \{r^0\}$, en ajoutant la colonne a_{r^0} à la matrice des contraintes. De façon générale, on résout à chaque itération t le problème maître restreint (\overline{PC}^t) :

$$(\overline{PC}^t) \equiv \begin{cases} \min \sum_{r \in \mathcal{R}^t} c_r x_r & (14) \\ \sum_{r \in \mathcal{R}^t} a_{ir} x_r \geq 1 \text{ pour } i \in V = \{1, \dots, n\} & (15) \\ x_r \geq 0 \text{ pour } r \in \mathcal{R}^t & (16) \end{cases}$$

tel que $\mathcal{R}^t = \mathcal{R}^{t-1} \cup \{r^{t-1}\}$

où, si δ^{t-1} désigne le vecteur de multiplicateurs associé aux n vols dans la résolution de (\overline{PC}^{t-1}) , la rotation r^{t-1} de plus faible coût réduit négatif est définie par

$$r^{t-1} = \arg \min_{r \in \mathcal{R}} \left(c_r - \sum_{i=1}^n \delta_i^{t-1} a_{ir} \right) \quad (17)$$

Le terme de *génération de colonnes* provient de l'ajout de la colonne $a_{r,t}$ à la matrice des contraintes du problème maître, à chaque itération t . Ce processus de résolution itérative du problème maître (14 – 16) et du sous-problème (17) est stoppé dès que toutes les rotations sont de coût réduit positif dans la résolution du sous problème, signe que l'optimum continu est atteint,

- soit à l'itération s telle que :

$$\min_{r \in \mathcal{R}} \left(c_r - \sum_{i=1}^n \delta_i^s a_{ir} \right) \geq 0$$

Une variante de cette méthode, permettant d'accélérer le processus en pratique [6], consiste à ajouter à chaque itération un sous-ensemble de rotations complémentaires de coût réduit négatif au lieu de la seule meilleure rotation du sous-problème (17)(voir [7]). La taille maximale souhaitée de ce sous-ensemble de colonnes entrantes pourra être paramétrée de manière à évoluer au cours de l'algorithme. La complexité globale de la méthode est fortement dépendante de la complexité du sous-problème, que les contraintes de ressources rendent NP-difficile. Il est souvent possible cependant de le résoudre en un temps raisonnable grâce à une énumération *implicite* de \mathcal{R} , en exploitant la structure de graphe du sous-problème et en appliquant des variantes d'algorithmes de plus court chemin.

3.3 Résolution du sous-problème pour la génération de colonnes

Notant que dans le cas de plusieurs sous réseaux $k = 1, \dots, K$, la résolution du sous problème(17)étant décomposable par sous réseaux, on omettra l'indice k et le graphe du sous problème sera noté $G = (\{o\} \cup V \cup \{d\}, A)$.

Le problème de plus court chemin avec contraintes des ressources $(PCC - CR)$, se formule comme suit :

$$(PCC - CR) \equiv \begin{cases} \min \sum_{(i,j) \in A} c_{ij} x_{ij} & (18) \\ \sum_{i:(o,i) \in A} x_{oi} = 1 & (19) \\ \sum_{i:(i,d) \in A} x_{id} = 1 & (20) \\ \sum_{i:(i,j) \in A} x_{ij} = \sum_{i:(i,j) \in A} x_{ji} \text{ pour } j \in V = \{1, \dots, n\} & (21) \\ x_{ij} (T_i^q + t_i^q - T_j^q) \leq 0 \text{ pour } (i,j) \in A, q \in \mathcal{Q} & (22) \\ a_i^q \leq T_i^q \leq b_i^q \text{ pour } i \in V, q \in \mathcal{Q} & (23) \\ x_{ij} \in \{0,1\}, T_i^q \geq 0 \text{ pour } (i,j) \in A, q \in \mathcal{Q} & (24) \end{cases}$$

Pour résoudre ce problème, Desrochers et Soumis [9] proposent un algorithme de programmation dynamique du type pulling.

Définition 1 A chaque chemin de l'origine o au nœud j , on associe une étiquette $E(C_j, T_j) = E(C_j, T_j^1, \dots, T_j^Q)$ représentant l'état de ses ressources et son coût.

Définition 2 Soient $E(C_j, T_j)$ et $E'(C'_j, T'_j)$ deux étiquettes associées à deux chemins réalisables P et P' de o à j . On dit que $E(C_j, T_j)$ domine $E'(C'_j, T'_j)$, (resp. P domine P'), et on note $E(C_j, T_j) \leq E'(C'_j, T'_j)$, (resp. $P \leq P'$) si et seulement si $C_j \leq C'_j$ et $T_j^q \leq T'^q_j, \forall q \in Q$.

Définition 3 Une étiquette associée à un chemin réalisable de o à j , est dite efficace si elle est minimale au sens de la relation d'ordre \leq . Un chemin est dit efficace s'il est associé à une étiquette efficace.

L'algorithme de programmation dynamique (APD) procède en trois grandes étapes. En chaque nœud $j \in V$, il effectue les opérations suivantes :

1. Prolongation des chemins (génération des étiquettes),
2. Filtrage (test de réalisabilité),
3. Dominance (élimination des étiquettes non efficaces).

Pour un nœud j donné, des étiquettes sont créées en prolongeant celles présentes aux nœuds i , tels que $(i, j) \in A$. Ainsi, une nouvelle étiquette $E(C_j, T_j)$ est donnée par

$$\begin{aligned} C_j &= C_i + c_{ij} \\ T_j^q &= \max\{a_j^q, T_i^q + t_{ij}^q\}, q \in Q \end{aligned}$$

En considérant que tous les prédécesseurs du nœud $j \in V$ sont déjà traités, la dominance au nœud j peut être interprétée comme la détermination des Pareto optimaux du problème multicritère à $|Q| + 1$ fonctions :

$$\begin{cases} \min_{i; (i,j) \in A} (C_i + c_{ij} ; \max\{a_j^q, (T_i^q + t_{ij}^q)\}, q \in Q) \\ T_i^q + t_{ij}^q \leq b_j^q, \quad q \in Q \end{cases}$$

Soit v^* sa valeur optimale.

La relation de dominance \leq étant une relation d'ordre partiel, le nombre d'étiquettes efficaces à traiter augmente de façon exponentielle en fonction du nombre de ressources, ce qui rend la procédure de prolongation très ardue.

Dans un récent travail Nagih et Soumis [4] proposent une méthode d'agrégation des ressources pour les PCC-CR par projection, en chaque nœud en utilisant simultanément un algorithme de programmation dynamique et une relaxation lagrangienne. Cependant, dans le cas général, cette approche ne donne pas toujours l'optimalité comme le montre l'exemple 1 suivant.

Exemple 1

Considérant le problème de plus court chemin avec une contrainte de ressource, représenté dans la figure 1. La solution optimale globale correspond à l'étiquette $E_3(10,6)$.

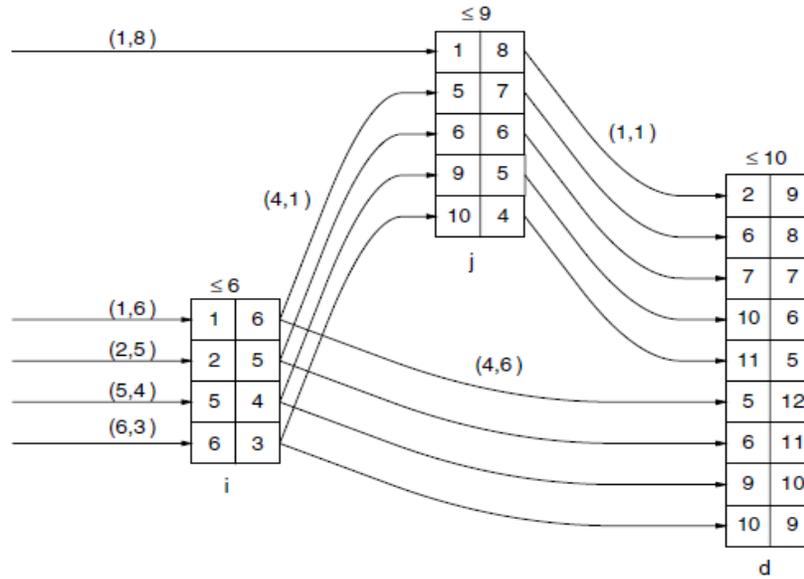


Figure 1 : Exemple 1-ensemble des étiquettes.

Comme on peut le voir sur la figure 2,

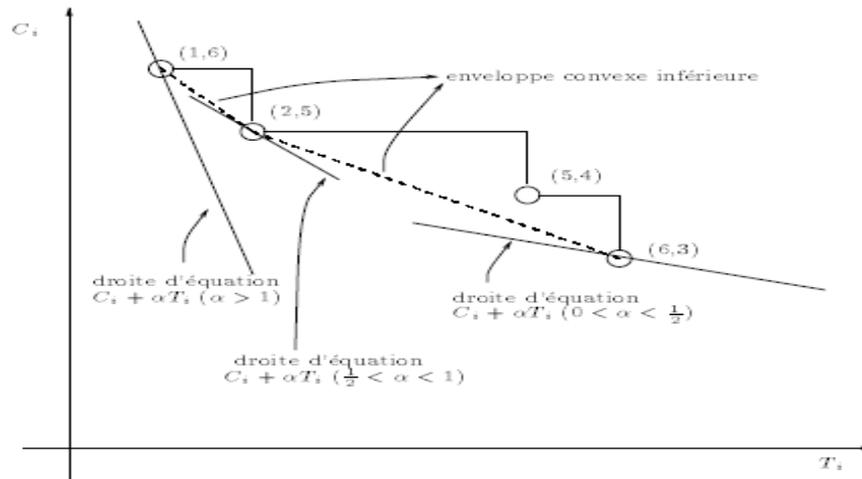


Figure 2 : Exemple 1-représentation des étiquettes au nœud 1 dans le plan (coût, ressource).

La dominance par rapport à la combinaison $(coût + u_j \times ressource)$ au nœud 1, conduit

$$\text{aux étiquettes } \begin{cases} E_4(6,3) \text{ si } u_1 \in \left[0, \frac{1}{2}\right] \\ E_2(2,5) \text{ si } u_1 \in \left[\frac{1}{2}, 1\right] \\ E_1(1,6) \text{ si } u_1 \in [1, +\infty[\end{cases}$$

Il n'existe donc pas de valeur de u_1 permettant de récupérer l'étiquette $E_3(5,4)$ au nœud 1, et par suite la solution optimale globale.

Par contre, en considérant un coefficient de projection qui dépend, en plus du nœud traité, du nœud prédécesseur, il existe toujours une instantiation de ces coefficients qui permet de récupérer la solution optimale globale, comme le montre le théorème 1.

Soit v_{noeud} la meilleure valeur obtenue par l'algorithme de Nagih et Soumis, et v_{arc} la meilleure valeur fournie par la deuxième technique.

Théorème 1

$$v^* = v_{arc} \leq v_{noeud}$$

Preuve : Soit (o, j_1) le premier arc du chemin optimal global. En prenant u_{ij_1} suffisamment grand pour tout $(i, j_1) \in A$ tel que $i \neq o$ et $u_{oj_1} = 0$, la solution au nœud j_1 correspondra au chemin (o, j_1) . Puis on réitère le même procédé jusqu'au nœud d . \square

Comme le nombre de coefficients à ajuster sera plus important pour l'approche de projection par arcs, trouver les multiplicateurs optimaux u_{ij}^* nécessitera plusieurs itérations successives de APD-L (voir l'annexe 1), cette méthode peut s'avérer coûteuse. Afin d'obtenir rapidement de bonnes solutions heuristique (réalisables), notre approche applique une seule fois APD-L puis applique APD-LND (voir l'annexe 1), en utilisant les multiplicateurs u_{ij} trouvés afin de produire des colonnes réalisables et de coût marginal négatif. Plus précisément, on choisit d'abord une suite de pas (p_k) telle que la série $(\sum p_k)$ est divergente et $\lim_{k \rightarrow \infty} p_k = 0$, autrement dit les conditions qui assurent la convergence de l'algorithme du sous-gradient aussi appelées conditions de Polyak [8]. On applique en premier lieu APD-L en utilisant les multiplicateurs u_{ij}^{k-1} de l'itération précédente, on trouve les sous-gradients $Sg_{ij}^k = t_{ij} - b_j$ correspondants à l'arc (i, j) ensuite on calcule les nouveaux multiplicateurs de Lagrange $u_{ij}^k = u_{ij}^{k-1} + p_k \times Sg_{ij}^k$. Cette heuristique est certainement basée sur le fait que lorsque k est grand, le vecteur C_k des coûts réduits sur les arcs du réseau ne change pas beaucoup d'une itération à une autre de l'algorithme de génération de colonnes. Ainsi pour k grand, on peut espérer voir u_{ij}^k converger vers une valeur optimale.

4. Algorithme (Méthode proposée « ALG.P.arc.CC »)

Les étapes principales de notre approche sont résumées ci-dessous :

Etap0	initialisation PMR^0
Etap1	(Résoudre PMR^k) par la méthode de simplex $\rightarrow (z_{PM}^k, x^k, \delta^k)$
Etap2	(Résoudre SP^k)
	- mettre à jour les coûts : $c_{ij} = c_{ij} - \delta_j$.
	- calculer les multiplicateurs de Lagrange u_{ij}^k .
	- calculer la solution $maxL(u_{ij}^k)$, en utilise APD-L
	- calculer les solutions réalisables $\Phi(u_{ij}^k)$, en utilise APD-LND
	- tester, Si $\min \Phi(u_{ij}^k) \geq 0$ alors stop, aller à l'étape 5. Si non aller à l'étape 3.
Etap3	généré la solution de meilleur coût négative et un sous-ensemble des solutions complémentaires qui peut être calculé par deux techniques (sélection, ou, résolution) $\rightarrow X^k$.
Etap4	posé $PMR^k = PMR^k \cup \{X^k\}$. Retournée à l'étape 1.
Etap5	$\rightarrow (z_{PM}^k, x^k)$ solution optimale.

5. Expérimentations

Cette section présente l'évaluation préliminaire de « ALG.P.arc.CC ». Jusqu'à maintenant, notre méthode n'a été testée que sur des problèmes de petite taille [voir l'annexe 2].

Les résultats de plusieurs tests sont présentés dans le tableau 1. Chaque ligne donne des informations sur la valeur optimale (la meilleure valeur) du problème trouvé par « ALG.P.s.M », « ALG.P.s.CC », « ALG.P.arc.M » (voir l'annexe 1), et notre méthode « ALG.P.arc.CC », le nombre d'itérations pour la méthode de génération de colonne.

Tableau. 1. Quelques résultats

problèmes	ALG.P.s.M		ALG.P.s.CC		ALG.P.arc.M		ALG.P.arc.CC	
	V	Nb.IT	V	Nb.IT	V	Nb.IT	V	Nb.IT
P1(initialisation 1)	7	1	7	1	6*	1	6*	1
P1(initialisation 2)	7	2	7	1	6*	2	6*	1
P2(initialisation 1)	15.5	10	15.5	6	15.5*	10	15.5*	4
P2(initialisation 2)	17	5	17	4	15.5*	4	15.5*	3
P3(initialisation 1)	23	15	23	7	23*	14	23*	4
P3(initialisation 2)	24	10	24	6	23*	10	23*	3

ALG.P.s.M : algorithme Agrégation des ressources par relaxation lagrangienne (Projection par sommet) avec meilleurs coûts réduits [4].
 ALG.P.s.CC : algorithme Agrégation des ressources par relaxation lagrangienne (Projection par sommet) avec colonnes complémentaires.
 ALG.P.arc.M : algorithme Agrégation des ressources par relaxation lagrangienne (Projection par arc) avec meilleurs coûts réduits.
 ALG.P.arc.CC : algorithme Agrégation des ressources par relaxation lagrangienne (Projection par arc) avec colonnes complémentaires.
 V : valeur optimal (la meilleure valeur) du problème trouver par cette technique ; Nb.IT : nombre d'itérations, (*) : valeur optimal.

Dans ce tableau, on considère scellement deux critères, la meilleure valeur du problème trouvé par cette technique et le nombre d'itérations globale.

Remarque

La comparaison entre les quatre techniques ont permis de constater que « ALG.P.arc.CC » a fourni de bons résultats. Ceux-ci sont meilleurs quand certaines conditions sont réunies : l'initialisation de l'algorithme, le choix des multiplicateurs de Lagrange et le pas de déplacement.

6. Méthode de séparation

La méthode de génération de colonnes est utilisée pour résoudre le problème relaxé au nœud U. Elle hybride l'algorithme de simplexe (méthode existant dans la librairie ILOG) avec une méthode nommée Pricing. Si la solution obtenue est fractionnaire alors une méthode de séparation est appliquée au problème P^u . Elle consiste à subdiviser l'ensemble des solutions entières S_u en deux sous ensembles disjoints, ceci a pour conséquence d'éliminer la réalisabilité de la solution fractionnaire pour les deux nouveaux problèmes qui sont des fils de P^u .

7. Conclusion et perspectives

Dans cet article consacré à la résolution du Problème de Construction des Rotations avec Contraintes de Ressources (PCR-CR), nous avons principalement développé les approches de génération de colonnes et de décomposition en problème maître et sous-problème. La

difficulté de la résolution du sous-problème étant directement liée au nombre de ressources, nous avons particulièrement étudié les techniques de réduction de l'espace des ressources, cette notion de réduction étant un élément-clé de l'efficacité de la résolution globale du problème. Les testes préliminaires effectués sur plusieurs instances ont montré l'efficacité de « ALG.P.arc.CC ». En effet, si dans une perspective de planification stratégique le temps de calcul peut s'avérer moins critique que le coût global du programme de rotations, en revanche dans un contexte opérationnel le gain de temps sur la résolution du sous-problème devient un enjeu majeur. Nos perspectives de recherche sur ce problème sont nombreuses. Parmi elles, citant les problèmes d'évalués la méthode « ALG.P.arc.CC » sur des instances de grande taille et sur des problèmes réels, aussi les problèmes de reconstruction d'une solution robuste suite à la perturbation, par un quelconque aléa, du planning initialement construit. Ces problématiques de ré-optimisation suscitent un intérêt croissant chez les ingénieurs chargés de la planification dans les grandes entreprises de transport et ouvrent des voies de recherche particulièrement intéressantes et prometteuses.

Références

- [1] DESAULNIERS G., DESROSIERS J., DUMAS Y., MARC S., RIOUX B., SOLOMON M., SOUMIS F., «Crew Pairing at Air France », *European Journal of Operational Research*, vol. 97, p. 245–259, 1997.
- [2] LAVOIE S., MINOUX M., ODIER E., « A new approach for crew pairing problems by column generation with an application to air transportation », *European Journal of Operational Research*, vol. 35, p. 45–58, 1988.
- [3] LOISEAU I., CESELLI A., N.MACULAN, SALANI M., « Génération de colonnes en programmation linéaire en nombres entiers », PASCHOS V. T., Ed., *Optimisation combinatoire : Concepts fondamentaux*, Hermès, Paris, 2005.
- [4] NAGIH A., SOUMIS F., « Nodal aggregation of resource constraints in a shortest path problem », *European Journal of Operational Research*, 2005.
- [5] CRAINIC T., ROUSSEAU J., « The column generation principle and the airline crew scheduling problem », *INFOR*, vol. 25, p. 136–151, 1987.
- [6] N. Touati, L. Létocart, and A. Nagih. Solutions diversification in a column generation scheme. En soumission à *Discrete Optimization*, 2008.
- [7] N. Touati, L. Létocart, and A. Nagih. Reoptimization in a column generation scheme. En soumission à *Computers and Operations Research*, 2008.
- [8] B.T. Polyak (1967). A General Method of Solving Extremum Problems, *Soviet Math. Doklady*, 8(3), 593–597.
- [9] M. Desrochers et F. Soumis (1988a), A Generalized Permanent Labelling Algorithm for the Shortest Path Problem with Time Windows, *INFOR* 26, 191–212.
- [10] Vangelis Paschos (2005), *Livre, optimisation combinatoire 3 : applications*, Hermès Science. ch 10.

Annexe 1

1-APD-L : Algorithme de programmation dynamique lagrangienne.

La prolongation se fait comme en (APD), la dominance au nœud j peut être interprétée comme la détermination des Pareto optimaux du problème multicritère a $|Q_2| + 1$ fonctions, ou $Q = Q_1 \cup Q_2$

$$\left\{ \begin{array}{l} \min_{i:(i,j) \in A} \left(C_i + c_{ij} + \sum_{q_1 \in Q_1} u_{ij}^{q_1} (\max\{a_j^{q_1}, (T_i^{q_1} + t_{ij}^{q_1})\} - b_j^{q_1}) ; \max\{a_j^{q_2}, (T_i^{q_2} + t_{ij}^{q_2})\} , q_1 \in Q_1, q_2 \in Q_2 \right) \\ T_i^{q_2} + t_{ij}^{q_2} \leq b_j^{q_2} , \quad q_2 \in Q_2 \end{array} \right.$$

2-APD-LND : Algorithme de programmation dynamique lagrangienne avec ressources non dominé. Cet algorithme est comme APD-L mais utilise la ré-optimisation pour construire des solutions réalisables.

3-ALG.P.s.M : algorithme Agrégation des ressources par relaxation lagrangienne (Projection par sommet) avec meilleurs coûts réduits [4].

Problème maître.	
Sour problème	<ul style="list-style-type: none"> – calculer les multiplicateurs de Lagrange u_j^k (Projection par sommet). – calculer la solution $maxL(u_j^k)$, en utilise APD – L – calculer les solutions réalisables $\Phi(u_j^k)$, en utilise APD – LND
Général la solution de meilleur coût négative.	

4-ALG.P.s.CC : algorithme Agrégation des ressources par relaxation lagrangienne (Projection par sommet) avec colonnes complémentaires.

Problème maître.	
Sour problème	<ul style="list-style-type: none"> – calculer les multiplicateurs de Lagrange u_j^k (Projection par sommet). – calculer la solution $maxL(u_j^k)$, en utilise APD – L – calculer les solutions réalisables $\Phi(u_j^k)$, en utilise APD – LND
Général la solution de meilleur coût négative et un sous-ensemble des solutions complémentaires.	

5-ALG.P.arc.M : algorithme Agrégation des ressources par relaxation lagrangienne (Projection par arc) avec meilleurs coûts réduits.

Problème maître.	
Sour problème	<ul style="list-style-type: none"> – calculer les multiplicateurs de Lagrange u_{ij}^k (Projection par arc). – calculer la solution $maxL(u_{ij}^k)$, en utilise APD – L – calculer les solutions réalisables $\Phi(u_{ij}^k)$, en utilise APD – LND
Général la solution de meilleur coût négative.	

Annexe 2

Problèmes	Nombre de vols	Arcs	Le nombre total de rotations
P1	4	15	16
P2	10	30	40
P3	20	70	102