RAIRO Operations Research
RAIRO Oper. Res. **39** (2005) 185–224
DOI: 10.1051/ro:2006003

FLOTS ENTIERS ET MULTIFLOTS FRACTIONNAIRES COUPLÉS PAR UNE CONTRAINTE DE CAPACITÉ

Alain Quilliot[1], Fatiha Bendali[1] et Jean Mailfert[1]

Abstract. We present here a Flow/Multicommodity Flow model for Transportation and Production Planning problems. We deal with this model through Lagrangean Relaxation and Hierarchical Decomposition techniques, which involve the resolution of a specific flow with least integral cost sub-problem, and which require the design of some agregation process. We deduce from this analysis several heuristic schemes, and we conclude by discussing numerical experiments.

Résumé. Nous modélisons ici plusieurs problèmes de Transport et de Gestion de Flux à l'aide d'un flot entier et d'un multiflot fractionnaire couplés par une contrainte de capacité. Pour le problème ainsi obtenu, nous proposons différents schémas de résolution par relaxation et décomposition, qui induisent la recherche d'un flot auxiliaire dont la partie entière supérieure doit minimiser un certain coût, et qui requièrent la mise en œuvre d'un processus d'agrégation. Nous en déduisons diverses heuristiques que nous testons.

Mots Clés. Flots, Multiflots, Circuits Négatifs, Routage, Transport.

Classification Mathématique. 90C25.

1. Introduction

Les problèmes de synthèse de réseaux issus de la modélisation de systèmes de Transport (voir [13, 22, 24, 33, 51, 53]), de Production (voir [2, 28, 48, 60, 64]), et surtout de Télécommunications ([10, 16, 17, 32, 36, 63]), constituent un secteur de recherche en expansion. Des ordinateurs de plus en plus puissants et des systèmes d'informations de plus en plus complets rendent en effet possible à la fois l'acquisition de données complexes (coûts, demandes, ...), et la mise en œuvre de traitements algorithmiques adaptés à des objets de très grandes tailles. Les grands

Reçu en 2002. Accepté le 12 septembre 2005.

[1] Bât. ISIMA, BP 125, Campus des Cézeaux, Université Blaise Pascal, 63173 Aubiere, France ; alain.quilliot@isima.fr

mouvements de déréglementation liés aux Transports et aux Télécommunications poussent par ailleurs les opérateurs à repenser, parfois profondément, l'organisation de leurs systèmes.

Configurer un tel système revient, de façon schématique, à déterminer des supports pour acheminer des ressources entre différents couples origine/destination. Ces supports sont, suivant les contextes :

- des vecteurs d'acheminement de l'information ou du signal, des nœuds de commutation ;
- des véhicules (bus, navettes, trains, avions) et des plate-formes d'échange ;
- des chariots filoguidés, des navettes ou jeux de palettes se déplaçant sur un site industriel ;
- des canalisations, pipelines, ...

La modélisation de ces situations fait souvent apparaître trois grandes classes de problèmes :

- les problèmes de *Localisation* : [7, 18, 20, 30, 31]. Il s'agit de localiser les infrastructures d'un système à l'intérieur d'un espace donné. Les méthodes associées sont le plus souvent de nature heuristique. Elles peuvent aussi dériver (méthodes polyédrales) d'une linéarisation du problème et du rôle des modèles linéaires en tant que formalisme universel au sein de la classe NP-Temps ;
- les problèmes de *Routage* : [3, 5, 19, 61]. Il s'agit de déterminer les cheminements que suivront les différents utilisateurs (signaux, usagers, produits...) du système à configurer, de façon à minimiser des taux de congestion. Ces problèmes, liés à des décisions tactiques ou opérationnelles, souvent issus du contexte des Télécommunications, sont en général traités par des techniques de flots et multiflots (voir [3, 5] : méthodes proximales, méthodes de déviation de flots...) ;
- les problèmes de *Dimensionnement* et *Topologie* : [3,11,14,23,25,55,57]. Il s'agit de déterminer les types d'infrastructures qui vont être utilisés par le système, leurs dimensions et leurs positionnements relatifs. L'objectif est alors de minimiser les coûts d'infrastructure, tout en assurant un routage efficace et en satisfaisant d'éventuelles contraintes de fiabilité (*survivability* : [6, 10]). Ces problèmes, liés à des niveaux de décision stratégique, sont très complexes, tant d'un point de vue pratique que théorique [45]. Ils peuvent faire l'objet de traitements heuristiques ou induire la mise en œuvre de techniques polyédrales ou de modèles de multiflots.

Dans tous les cas, le caractère prospectif des problèmes posés, le niveau d'approximation existant sur les entrées des modèles et la nature forcément réductrice de ceux-ci, font qu'il est de fait moins important d'obtenir des solutions numériques exactes, que de savoir gérer les principaux points durs de ces modèles de façon robuste.

La classe des problèmes que nous étudions ici s'inscrit dans cette dernière catégorie des problèmes de dimensionnement et de topologie et correspond à la formulation suivante :

Problème du Couplage Flot Entier/MultiflotFractionnaire (CFEMF) :

{Sur un réseau (graphe orienté) G = (X,E), on considère :
- *un sous-ensemble Support A de E ;*
- *un ensemble d'Indexation K ;*
- *un vecteur de capacité entier MAX = (MAX$_e$, e dans E) ;*
- *deux familles* $C\min$ *et* $C\max$ *de vecteurs de capacités* $C\min(k) = (C\min(k)_e,\ e \in E) \geq 0$ *et* $C\max(k) = (C\max(k)_e,$ *e dans E)* ≥ 0, *indexées sur k dans K ;*
- *2 vecteurs de coûts positifs c = (c$_e$, e dans E) et p =(p$_e$, e dans E) ;*

on ***cherche alors un flot F dit « véhicule »*** $\geq$ **0,** ***entier****, et* ***un multiflot f =*** *{f(k), k dans K}* $\geq$ *0, dit «* ***usager*** *»,* ***fractionnaire****, définis sur G, tels que :*

* *pour tout indice k dans K,* $C\ \min(k) \leq f(k) \leq C\ \max(k)$ *;* *(C1)*
(on notera cette contrainte : $C\ \min \leq f \leq C\max$*)*

* $F \leq MAX$ *; (C2)*

* *pour tout arc e dans A,* $\mathrm{Sum}(f)_e \leq F_e$; (***Contrainte Couplante***) $(C3)$

*et minimisant la quantité c.F + p.*Sum*(f)* $= \Sigma_{\ e\, dans\, E} c_e.F_e + \Sigma\Sigma_{e\, dans\, E,\, k\, dans\, K} f(k)_e.p_e\}$.

Un cas très important est celui où l'ensemble d'indexation K correspond de fait à une famille de couples de sommets origines et destinations de X, et où, pour chaque couple $k = (o,d)$ dans K, la contrainte (C1) se réécrit :

le flot f(k) achemine une quantité de flot D_k depuis o vers d ; (C1*)

on note alors **CFEMF-OD** le problème ainsi obtenu.

Remarque. La quantité Sum(f) est définie de façon algébrique. Mais le multiflot f étant assujetti à être positif, la contrainte (C3) peut néanmoins être vue comme une contrainte de capacité classique.

Formellement, ce problème linéaire en nombres mixtes est proche des modèles de type CFA (*Capacitated Flow Assignment*) : [9, 14, 25, 55]. Ceux-ci, hérités du contexte des Télécommunications, portent sur un vecteur entier F indexé sur les arcs du réseau G, et sur un multiflot f qui achemine des demandes émanant d'un ensemble K de couples de sommets origines/destinations, F dominant f sur l'ensemble des arcs de G et satisfaisant d'éventuelles contraintes de connectivité. La quantité à minimiser est alors l'expression d'un coût d'infrastructure linéaire et d'un coût (mesure de congestion) de qualité de service, éventuellement non linéaire.

Il comporte néanmoins des spécificités, justifiant un mode de traitement propre. En effet, comme cela sera illustré dans la section suivante, ce modèle dérive plus de considérations relatives aux systèmes de transport et de production que du contexte des Télécommunications. La contrainte de flot sur F exprime que celui-ci représente non pas un systèmes de connections fixes, mais un objet (ensemble de *véhicules*) circulant, que les *usagers* représentés par le multiflot f peuvent se passer d'utiliser. Il s'ensuit que la contrainte de couplage (C3) **ne concerne que certains arcs** (les arcs de l'ensemble *Support A*) du réseau. Il en découle :

- qu'il devient difficile d'introduire des coupes sur F, traduisant le fait que les arcs supports de F doivent permettre de connecter tout couple de sommets (o, d) de la famille K, et d'en déduire une approche polyédrale telle que pratiquée en [9, 14, 26, 52, 62] ;
- que la contrainte d'intégrité sur F traduit le besoin (très apparent dans les applications liées au Transport) qu'ont les utilisateurs de se regrouper afin de se partager un réseau d'infrastructure faiblement maillé par rapport au réseau initial $G = (X, E)$. La relaxation de la contrainte d'intégrité sur F fournit dès lors ici, dans la plupart des cas, de très mauvais résultats, reflétant cette absence de partage, et ce contrairement à ce qui est observable dans le cas des problèmes liés aux Télécommunications (voir par exemple [25] ;
- que le *flot agrégé* Sum(f), synthèse de la mutualisation par les usagers de leurs stratégies d'acheminement, devient dès lors *l'Objet Maître* à l'intérieur du problème **CFEMF**. Il est dès lors souhaitable, dans la perspective d'un traitement heuristique de ce problème, de pouvoir agir directement sur Sum(f) et F, et ce en tirant parti des spécificités des espaces de flots.

Ces constatations sont le moteur de notre démarche. Ce que nous voulons ici est concevoir un procédé algorithmique de nature heuristique, propre à traiter des instances de **CFEM** de tailles importantes, agissant directement sur le *flot agrégé* Sum(f) à l'aide de procédures de transformation locale à base de cycles et de chemins. Dans un premier temps (Sect. 2) nous décrivons 3 cas dont la modélisation renvoie à **CFEMF**, et qui permettent d'appréhender le sens de l'ensemble support A et de la contrainte de flot imposée à F. Nous présenterons ensuite (Sect. 3), 3 approches très générales pour traiter ce problème **CFEMF** :

- par relaxation lagrangienne de la contrainte de flot en F ;
- par relaxation lagrangienne de la contrainte de couplage (C3) ;
- par décomposition hiérarchique, l'objet multiflot f étant considéré comme l'objet Maitre ;

qui toutes recentrent le problème autour de l'objet agrégé Sum(f). Ce recentrage questionne la gestion de l'articulation entre f et Sum(f) (processus d'agrégation/ désagrégation), et celle d'un flot fractionnaire compatible avec des capacités et dont la partie entière supérieure minimise un certain coût. Nous proposons une heuristique pour ce dernier problème (Sect. 4.1) puis présentons (Sect. 4.2) un traitement du problème **CFEMF-OD** par action directe sur le flot agrégé Sum(f). Dans les deux cas, les algorithmes ainsi présentés tirent parti, tant pour la gestion de F que de Sum(f) et f, des procédés classiques de transformation de flots par chemins et cycles. Nous concluons cet article par la présentation d'un ensemble d'expérimentations numériques (Sect. 5).

2. Modèles et interprétations

2.1. Principales notations

Un réseau G, d'ensemble de sommets X et d'ensemble d'arcs E est noté $G = (X, E)$. La matrice d'incidence arcs/sommets de ce réseau est notée $M(G)$: si l'arc $e = [x, y]$ va de x vers y, alors $M(G)_{x,e} = 1$ et $M(G)_{y,e} = -1$.

Un multiflot f, défini sur G et dont les flots composants sont indexés sur un ensemble K, est noté $f = \{f(k),\, k \text{ dans } K\}$. On note $\mathrm{Sum}(f)$ le *Flot Agrégé* défini par $\mathrm{Sum}(f) = \Sigma_{k \,\mathrm{dans}\, K} f(k)$.

Si γ est un cycle (chemin), orienté de G, nous notons ϕ_γ le *flot-cycle* (*flot-chemin*) associé à γ, c'est-à-dire le vecteur flot qui vaut 1 sur les arcs de γ orientés comme γ, -1 sur les arcs de γ orientés contre γ, et 0 ailleurs.

On dit qu'un flot f *achemine* une quantité D de o vers d si f s'écrit $f = D.\ \phi_\gamma$, où γ est un chemin allant de o vers d.

Si x et y sont 2 sommets de G, et si D est un nombre réel, nous notons $D^{x,y}$ le vecteur indexé sur X, égal à D pour l'indice x, à $-D$ pour l'indice y, et à 0 ailleurs. Si z est un vecteur indexé sur X et si A est un sous-ensemble de X, alors z_A est la trace de z sur A. On note z^+ la partie positive du vecteur z, et z^- sa partie négative.

2.2. Modélisation d'un problème de tournées à l'aide d'un réseau dynamique

Sur un graphe orienté $H = (Z, U)$ représentatif d'un réseau urbain, nous considérons des sommets particuliers $y_1,\ y_2,\ \ldots,\ y_m$ correspondant à l'emplacement de sites de production (lieux de travail). À chaque sommet $x \in Z$, et à chaque site y_k est associée une quantité fixe $d_{x,yk,tj}$ d'usagers à acheminer depuis x vers y_k, avant la date t_j, $j \in J^k$. La durée d'acheminement ne doit pas excéder un seuil $T_{x,yk}$. On souhaite organiser un système de navettes, qui, pour un coût minimal, va assurer ces acheminements. On suppose que :

- chaque tournée de véhicule débute et s'achève en un sommet *Dépôt* unique ;
- à chaque véhicule correspond une capacité unique, et l'unité de mesure choisie pour les flux d'usagers est telle que cette capacité est égale à 1, et que les quantités $d_{x,yk,tj}$ sont fractionnaires ;
- le coût du système dépend linéairement du nombre de véhicules impliqués et des durées des tournées ;
- les déplacements des usagers se font par combinaison de déplacements en véhicule et de déplacements à pied. À chaque arc e de H sont donc associées une durée $l_p(e)$, *à pied*, et une durée $l_v(e)$, en *véhicule*.

Un réseau dynamique associé

Le problème se modélise sous forme **CFEMF-OD** grâce à un *réseau dynamique* (voir [4] pour la notion de réseau dynamique) qui intègre les contraintes temporelles sur les tournées. On considère (*Discrétisation du Temps*) une unité de temps discrète δ et un entier N, tels que la période correspondant à l'ensemble des acheminements se situe entre les instants 0 et $N\delta$. Pour chaque arc e de H, on pose :

$$l_p^*(e) = \lceil l_p(e)/\delta \rceil; l_v^*(e) = \lceil l_v(e)/\delta \rceil; t_j^* = [t_j/\delta];$$

À un sommet quelconque x de Z, on associe $(N+1)$ copies de x, indexées de 0 à N, qui représentent l'état de x aux instants 0, δ, …, $N\delta$. On rajoute un sommet D (dépôt), destiné à l'écriture des lois de Kirschoff, et on pose : $X = \{x_r,\ x \in X,\ r \in 0, \ldots, N\} \cup \{D\}$. Sur l'ensemble de sommets X, on définit :

- des arcs $[x_r,\ x_{r+1}]$ pour $x \in Z$ et $r \in 0, \ldots, N-1$, dédoublés selon un étiquetage *navette*, et *à-pied* ;
- des arcs étiquetés *navette* $[D, \text{Depot}_r]$ et $[\text{Depot}_r, D]$, $r \in 0, \ldots, N$;
- des arcs "*navette*" $[x_r,\ z_{r+l^*v(e)}]$, pour $e = [x, z] \in U$ et pour $r = 1...m$ tels que $0 \leq r \leq N - l_v^*(e)$;
- des arcs $[x_r,\ z_{r+l^*p(e)}]$ étiquetés *à-pied* pour $e = [x, z] \in U$ et r tels que $0 \leq r \leq N - l_p^*(e)$;

L'ensemble des arcs étiquetés *navette* est noté A.

À chaque arc e du réseau dynamique $G = (X, E)$ ainsi défini, on associe un coût L_e en posant :

- $L_e = l_v(e)$ si e est de la forme $[x_r,\ z_{r+l^*v(e)}]$;
- $L_e =$ une constante μ si e est de la forme $[x_r,\ x_{r+1}]$: ce coût *d'attente*, est un coût de fonctionnement ;
- $L_e =$ une constante α si e est du type $[D, \text{Depot}_r]$: ce coût est lié au nombre de véhicules utilisés ;
- $L_e = 0$ ailleurs.

Le modèle induit

Sur $G = (X,\ E)$ ainsi construit, on cherche alors un flot $F \geq 0$ entier, représentant les parcours de navettes, et un multiflot $f = \{f(k,j), k = 1...m, j$ dans $J^k\} \geq 0$, représentant les parcours des usagers, tels que :

- F est nul sur les arcs *à-pied* ;
- $f(k,j)$ *achemine* la demande $d_{x,yk,tj}$ du sommet $x_{w*(x,k,j)}$, où $w*(x,k,j) = \lceil (t_j - T_{x,yk})/\delta \rceil$, vers le sommet y_{t*j} ;
- $\text{Sum}(f)_e = \Sigma_{k,j} f(k,j)_e \leq F_e$, pour chaque arc e de E étiqueté *navette*

et qui minimisent une quantité $L.F = \Sigma_{(e \text{ dans } A)}\ L_e.F_e$.

Ce modèle, de type **CFEMF-OD**, permet l'expression implicite des contraintes temporelles et des contraintes de synchronisation, qui dérivent de la possibilité qu'a un usager d'avoir recours à plusieurs véhicules. Il serait possible de l'enrichir en intégrant un coût $p.f$ de *Qualité de Service* relatif aux usagers, qui exprimerait l'intérêt qu'ont ceux-ci à se voir proposer les parcours les plus rapides possibles.

On vérifie que la reconstitution d'un système cohérent de tournées découle directement de la connaissance du flot F. L'approximation induite par les remplacements respectifs de chaque temps $l_v(e)$ et $l_p(e)$ par $\delta.l_v^*(e)$ et $\delta.l_p^*(e)$ peut toutefois produire, dans le cas où δ est assez grand, des parcours relativement lents. Un traitement de ce problème des *Navettes* s'appuyant sur la modélisation ci-dessus, s'effectuera dès lors en 2 étapes :

1re étape : résolution de l'instance de **CFEM-OD** associée ;
2e étape : décomposition de F en un système de tournées ;
optimisation des temps de passage des véhicules durant ces tournées (*graphicage*).

On voit ici le rôle du vecteur F, qui représente la circulation de la flotte de véhicules, et celui de l'ensemble A, qui identifie ceux des déplacements des usagers à l'intérieur du réseau qui sont effectués à l'aide de ces véhicules. Relâcher la contrainte d'intégrité sur F revient ici à offrir à chaque usager sa propre fraction de véhicule, et induit donc une dislocation des parcours individuels, soit le contraire de ce qui est recherché.

2.3. Modélisation d'un système de production

On considère ici un atelier dont il faut planifier l'activité cyclique sur $n + 1$ périodes séparées par des interpériodes, pour produire des biens $b_1, b_2, \ldots, b_m$. Ces biens font partie d'une même gamme de produits, ce qui signifie qu'ils sont structurellement proches les uns des autres, et qui permet de les considérer de façon unifiée dès lors qu'il est question de capacités de production ou de coûts de stockage. On suppose que :

- la demande en biens b_k à la fin de la période i est $d_{i,k}$;
- le coût unique de stockage d'une unité de produit est α ;
- la capacité de stockage sur l'ensemble des produits pour une interpériode est β ;
- la capacité totale de production de l'atelier est c. Elle peut être augmentée à un niveau d moyennant la location d'une machine spécifique au coût γ par période. Les opérations de montage et démontage de cette machine doivent être effectuées lors des interpériodes aux coûts respectifs de γ_1 et γ_2.

L'activité de l'atelier (gestion des stocks, de la production avec ou sans machine supplémentaire. . .) doit dès lors être planifiée de façon à minimiser les coûts tout en satisfaisant les demandes.

On utilise, afin de définir un modèle **CFEMF** de ce problème, le réseau dynamique construit comme suit :

- pour chaque période $i = 0, \ldots, n$, on crée deux sommets x_i, y_i (début et fin de période), auxquels on ajoute trois sommets *machine, client, production ;*
- sur l'ensemble X de sommets ainsi construit, on crée :
  - pour chaque $i = 0, \ldots, n$, les arcs $[y_i,$ *client*$]$, $[$*production,* $x_i]$ et $[x_i, y_i]$, ce dernier arc étant dédoublé selon 2 étiquettes *Sans-Machine* et *Avec-Machine* : ces arcs vont respectivement porter les quantités de biens livrées, demandées et produites aux différentes périodes ;

– pour chaque $i = 0, \ldots, n$, un arc $[y_i,\ x_{i+1}]$ (à cause de la cyclicité, l'addition est prise modulo $n + 1$), dédoublé selon 2 étiquettes *Sans-Machine* et *Avec-Machine* : ces arcs vont respectivement porter les quantités de biens stockées et l'état de la machine entre les périodes i et $i + 1$;
– pour chaque $i = 0, \ldots, n$, un arc [*machine,* x_i] et un arc (y_i*, machine*], qui porteront l'information relative au montage et au démontage de la machine en amont et en aval de la période i ;
– un arc [*client, production*] de retour.

Sur ce réseau $G = (X, E)$, on cherche alors un multiflot $f = \{f(k), k = 1...m\} \geq 0$, représentatif de la circulation des biens $b_1, b_2, \ldots, b_m$, et un flot F en {0,1}, représentatif de l'état de la machine, tels que :

– $F = 0$ sur tout arc dont une extrémité est *client* ou *production*, ainsi que sur les arcs *Sans-Machine* ;
– $f = 0$ sur tout arc dont une extrémité est *machine*, ainsi que sur les arcs *Avec-Machine* du type $[y_i,\ x_{i+1}]$;
– $f(k)$ est égal à $d_{i,k}$ pour tout arc $[y_i,\ \mathit{Client}]$;
– pour tout arc e de la forme $[y_i,\ x_{i+1}]$ et étiqueté *Sans-Machine*, $\mathrm{Sum}(f)_e \leq \beta$; (*contrainte de Stockage*)
– pour tout arc $e = (x_i,\ y_i)$ et étiqueté *Sans-machine*, $\mathrm{Sum}(f)_e \leq c_e$; (*1re contrainte de capacité*)
– pour tout arc (ensemble *Support*) $e = (x_i,\ y_i)$, étiqueté *Avec-machine*, $\mathrm{Sum(f)}_e \leq F_e.(d_e - c_e)$;
– et minimisant la quantité : $\gamma_1.\Sigma_i F_{[\mathrm{mach},yi]} + \gamma_2.\Sigma_i F_{[xi,\mathrm{mach}]} + \gamma.\Sigma_i F_{[xi,yi]} + \Sigma_{i,k}\alpha.f(k)_{[yi,xi+1]})$.

Remarque. La contrainte de stockage et la 1re contrainte de capacité contraignent entre elles les composantes du multiflot f, ce qui déborde du cadre proposé pour **CFEMF**. Les méthodes proposées en sections 3 et 4 demeureront cependant adaptables, et ce d'autant plus aisément que ces contraintes portent de fait sur l'objet agrégé $\mathrm{Sum}(f)$, qui sera l'objet principal à l'intérieur de ces méthodes.

2.4. Modélisation du problème du circuit hamiltonien sous la forme CFEMF

Soit donc $H = (Y, U)$ un réseau, supposé fortement connexe, et soit x_0 un sommet de référence de H. Construisons un réseau auxiliaire $G = (X, E)$ comme suit :

– $X = Y \cup Y'$ où Y' est une copie de Y ;
– $E = \{[x, x'],\ x \in Y,\ x'$ étant sa copie dans $Y'\} \cup \{[x', y]$, pour tous les arcs $[x, y] \in U\} \cup \{[x_o, x'],\ x' \in Y'\}$.

On définit l'ensemble support A comme formé des arcs de la forme $[x', y], x' \in Y', y \in Y$, et on pose $n = [X]$. **Chercher un circuit hamiltonien** dans H

signifie alors chercher sur G un flot entier $F \geq 0$, et un flot fractionnaire $f \geq 0$ tels que :

– $\forall x'$ dans Y', $f_{[xo,x']} = 1/n$;
– pour tout arc e dans A, $\mathrm{Sum}(f)_e \leq F_e$;
– $F_e = 1$ pour tout arc e de la forme $e = [x, x'], x \in Y$.

On en déduit, dans la continuité des résultats relatifs à la complexité de la synthèse de réseau (voir [45]) :

Proposition 1. *Le problème CFEMF, considéré dans le cas particulier où $k = 1$ est NP-difficile.*

3. Schémas généraux de résolution pour CFEMF

Soit donc, conformément aux notations de la section 1, un réseau $G = (X, E)$, pour lequel sont donnés :

– un sous-ensemble *Support* A de E ;
  – un ensemble *d'Indexation* K,
  – un vecteur *Capacité* entier $MAX = (MAX_e,\ e$ dans $E)$,
  – deux familles $C\min$ et $C\max$ de vecteurs *Capacité* $C\min(k) = (C\min(k)_e,\ e$ dans $E) \geq 0$ et $C\max(k) = (C\max(k)_e,\ e$ dans $E) \geq 0$, indexées sur k dans K,
  – 2 vecteurs *Coût* positifs $c = (c_e,\ e$ dans $E)$ et $p = (c_e,\ e$ dans $E)$.

On cherche alors, sur G (problème **CFEMF**) :
un flot $F \geq 0$, entier, et un multiflot $f = \{f(k),\ k$ dans $K\} \geq 0$, respectivement compatibles avec le vecteur MAX et avec les familles $C\min$ et $C\max$, tels que :

> *pour tout arc e dans A, $\mathrm{Sum}(f)_e = \Sigma_{\ k\, dans\, K} f(k)_e \leq F_e$ (Contrainte couplante C3) et tels que la quantité $c.z + p.\mathrm{Sum}(f)$ soit la plus petite possible.*

Notations additionnelles.

On note V la valeur optimale du problème **CFEMF** et V^* la valeur optimale du problème **CFEMF*** déduit de **CFEMF** par relaxation de la contrainte d'intégrité sur F.

Un flot entier F^* étant donné, on note **CFEMF**^{F*} le problème **CFEMF** restreint en imposant $F = F^*$, V^{F*} la valeur optimale associée, et $\boldsymbol{DU}^{F*}$ le dual de **CFEMF**^{F*}.

Un multiflot f^* étant donné, on note **CFEMF**$_{f*}$ le problème **CFEMF** restreint en imposant $f = f^*$, V_{f*} la valeur optimale associée et $\boldsymbol{DU}_{f*}$ le dual de **CFEMF**$_{f*}$.

On note E^∞ l'ensemble des arcs e de E tels que $MAX_e = +\infty$.

Cette section traite donc de l'application au problème **CFEMF** de schémas de décomposition et de relaxation, et cela afin d'induire des propositions d'heuristiques. Comme il a été dit, relâcher la contrainte d'intégrité sur F déforme fortement le problème, et il est difficile de générer des coupes sur F qui compensent cette relaxation. Du fait du rôle central joué par le vecteur agrégé $\mathrm{Sum}(f)$, qui synthétise le *partage* de la ressource F par les flots usagers $f(k)$, $k \in K$, on va étudier des modes de **décomposition qui considèrent ce vecteur Sum(f) comme objet principal** du problème, à savoir :

- un procédé de décomposition *Maître/Esclave* **DME** (Benders, Sect. 3.1), qui considère f comme l'objet Maître et exploite les propriétés des flots entiers. Ce procédé va faire apparaître un problème auxiliaire **P-aux** de "*Multiflot de Coût Entier Minimal*", qui sera approfondi en section 4.1 ;
- un procédé de relaxation lagrangienne **DRCOUP** de la contrainte couplante (C3), (Sect. 3.2), et un procédé de relaxation lagrangienne **DR-FLOT** de la contrainte de flot sur F, (Sect. 3.3), qui induiront des résultats similaires et l'appel au traitement d'une instance du problème auxiliaire **P-aux**.

L'analyse à laquelle nous nous livrerons tendra à suggérer que les comportements de ces trois procédés sont assez voisins et qualitativement satisfaisants, et que les cas pour lesquels ils débouchent sur une valeur optimale sont les mêmes (Ths. 1 et 2). Ceci justifiera (Sects. 4.1 et 4.2) que nous cherchions à nous appuyer sur ces procédés, et notamment sur le mode de décomposition Maître/Esclave **DME**, qui présente l'avantage d'opérer sur des solutions réalisables de **CFEMF**, de façon à en extraire une heuristique opérationnelle.

3.1. Schéma DME : décomposition maître-esclave

Soit donc (F, f) une solution réalisable de **CFEMF**, choisie telle que F soit solution optimale du problème restreint **CFEMF**$_f$. Correspond alors à ce couple (F, f) une solution optimale ($\mu = (\mu_x, x \in X)$, $\alpha = (\alpha_e, e \in E) \geq 0$, $\lambda = (\lambda_e$, $e \in A) \geq 0$) du dual **DU**$_f$ de **CFEMF**$_f$. Améliorer le couple (F, f) signifie modifier f de façon à le maintenir positif, compatible avec les familles $C\min$ et $C\max$, et à diminuer la quantité :

$$\lambda.\lceil \mathrm{Sum}(f) \rceil + p.\mathrm{Sum}(f) - \alpha.MAX. \tag{E.1}$$

La réciproque est bien entendu fausse. Cette remarque nous conduit à introduire un problème auxiliaire :

> ***Problème P-aux(***λ***) :*** $\{$*Le vecteur* λ*,* ≥ 0 *étant donné, indexé sur E et nul sur E-A, chercher un multiflot* $f \geq 0$*, compatible avec* $C\min$ *et* $C\max$ *et minimisant la quantité :* $\lambda.\lceil \mathrm{Sum}(f) \rceil + p.\mathrm{Sum}(f)\}$

et à proposer le schéma algorithmique suivant :

> ***Schéma algorithmique DME (Décomposition Maître/Esclave)***
> **Initialiser** f, réalisable pour le problème **CFEMF** ; Not Stop ; $\Delta := + \propto$;
> **Tant que** $\neg$ Stop **faire**
> > **Résoudre CFEMF**$_f$ et **extraire** la composante primale F et la composante duale λ associées ;
> > $\Delta :=$ c.F $+ p.\mathrm{Sum}(f)$; Mettre à jour Stop (en cas de dégradation ou de *surplace prolongé* de Δ) ;

Perturber f de façon à diminuer la quantité $\lambda.\lceil \mathrm{Sum}(f)\rceil + p.\mathrm{Sum}(f)$ tout en maintenant f compatible avec $C\min$ et $C\max$; (I1)
Si l' instruction (I1) échoue **alors** Stop ;

Surplace Prolongé signifie ici que l'on n'enregistre plus d'amélioration de la quantité $c.F + p.\mathrm{Sum}(f)$, supérieure à un seuil ε, depuis N d'itérations de la boucle principale : N et ε sont alors des paramètres du processus.

Nous reviendrons sur le problème **P-Aux** en sections 4 et 5.

Remarque. Ce schéma **DME** est **heuristique**, puisque l'évolution de f au cours d'une itération est susceptible d'aller en sens contraire des perturbations appliquées au cours des itérations précédentes. Il serait bien sûr possible de réécrire **DME** sous forme d'un processus de génération incrémentale de coupes de la forme :

$$\lambda.\lceil \mathrm{Sum}(f)\rceil + p.\mathrm{Sum}(f) \geq S, \qquad (\alpha)$$

qui serait alors exact (modulo les problèmes liés à la vitesse de convergence du processus). Mettre en œuvre ce processus impliquerait toutefois de savoir gérer simultanément un grand nombre de coupes de type (α) ci-dessus.

3.2. Schéma DRCOUP : relaxation de la contrainte couplante

Relaxer la contrainte couplante (C3) signifie introduire un vecteur de Lagrange $\lambda \geq 0$, indexé sur E et nul sur $E - A$, et poser :

Problème P-Coup$_\lambda$: $\{$*Chercher sur G un flot entier* $F \geq 0$ *et un multiflot f* $=$ *{f(k), k dans K}* ≥ 0*, compatibles avec leurs capacités, et qui minimisent la quantité c.F + p.*$\mathrm{Sum}(f) - \lambda.(F - [\mathrm{Sum}(f)])\}$.

On note B_λ la valeur optimale de **P-Coup**$_\lambda$ et on pose : $B = \mathrm{Sup}_{(0\leq\lambda)} B_\lambda$. Le problème **P-Coup**$_\lambda$ se sépare en une instance du problème **P-aux**(λ), et en un problème de flot entier de coût minimum. On peut se limiter à λ tel que $B_\lambda > -\infty$, soit tel qu'il n'existe pas de circuit négatif pour les coûts $c - \lambda$ dans le réseau $G^\infty = (X, E^\infty)$.

Quand $E^\infty = E$, on peut donc associer à λ un vecteur potentiel Π, indexé sur X et tel que : $\Pi.M(G) \leq c - \lambda$. La correspondance qui à λ associe B_λ étant alors croissante en λ, on obtient :

$$B = \mathrm{Sup}_{\Pi \text{ tels que } \Pi.M(G)\leq c} \mathrm{Inf}_{f \text{ tels } C\min\leq f\leq C\max}(p.\mathrm{Sum}(f) + (c - \Pi \mathrm{uM}(G))_A.\lceil \mathrm{Sum}(f)_A\rceil). \quad (\mathrm{E2})$$

Dans le cas inverse, ou peut supposer que le vecteur MAX est borné, quitte à introduire des capacités fictives suffisamment grandes, et B se calcule alors selon le schéma suivant :

Initialisation de λ ; Not Stop ;
Tant que Not Stop **faire**
Résoudre la composante en F de **P-Coup**$_\lambda$; soit F_λ la solution obtenue ;

Résoudre la composante en f de **P-Coup**$_\lambda$; soit f_λ la solution ;
Mettre à jour Stop ;
Si Not Stop alors
Calculer une quantité *Pas* ; $\lambda := \text{Sup}\ (0,\ \lambda - Pas.\ (F_\lambda - \lceil \text{Sum}(f_\lambda)\rceil)_A$;

Dans tous les cas, il nous est suggéré le schéma de résolution **DRCOUP** suivant pour le problème **CFEMF** :

phase 1 : **calcul** de B ;
phase 2 : extraction par projection d'une solution de **CFEMF**.
Soit λ_o tel que $B_{\lambda o}$ peut être pris comme approximation de B ;
soit f_o la solution de la composante en f du problème **P-Coup**$_\lambda$;
résoudre le problème **CFEMF**$_{fo}$; soit F_o la solution obtenue ;
prendre le couple (F_o, f_o) comme solution approchée du problème **CFEMF**.

3.3. Schéma DRFLOT : relaxation de la contrainte de flot sur F

Relaxer la contrainte de flot sur F signifie poser, pour tout vecteur de Lagrange μ indexé sur les sommets de G :

Problème P-Flot$_\mu$: $\{$*Chercher sur G un vecteur entier $F \geq 0$ et un multiflot $f = \{f(k),\ k \in K\} \geq 0$, compatibles avec leurs capacités, tels que* $\text{Sum}(f)_A \leq F_A$, *et minimisant $c.F + p.Sum(f)\ \text{-}\mu.\ M(G).F$*$\}$.

On note D_μ la valeur optimale de **P-Flot**$_\mu$ et on pose : $D = \text{Sup}_\mu D_\mu$.

On peut se limiter aux vecteurs μ assurant la finitude de D_μ, soit tels que pour tout arc e dans E^∞, on ait $(c - \mu(MG))_e \leq 0$. Pour tout vecteur μ, la solution optimale $(F_\mu,\ f_\mu)$ de **P-Flot**$_\mu$ est alors telle que :

pour tout arc e dans A, tel que $0 \leq (c - \mu M(G)_e$, on a $F_e = \lceil \text{Sum}(f)_e \rceil$;
pour tout arc e dans $E - A$, tel que $0 \leq (c - \mu M(G))_e$, on a $F_e = 0$;
pour tout arc e dans E, tel que $(c - \mu M(G))_e \leq 0$, on a $F_e = MAX_e$.

Le problème **P-Flot**$_\mu$ se réécrit alors : (E3)

$\{$*Trouver un multiflot $f \geq 0$, compatible avec* $C\min$ *et* $C\max$*, tel que $Sum(f)_A \leq MAX_A$, et qui minimise :* $p.\text{Sum}(f) - \Sigma_{\ e \text{ dans } A/0 \leq (c-\mu M(G))e}\ (c - \mu M(G))_e.\ \lceil \text{Sum}(f)_e \rceil + \Sigma_{\ e \text{ dans } E/(c-\mu M(G))e \leq 0}\ (c - \mu M(G))_e.\ MAX_e\}$.

On reconnaît là une instance du problème **P-aux**, enrichie d'une contrainte globale sur le vecteur $\text{Sum}(f)$.

Faisant l'hypothèse que nous savons traiter **P-aux**, nous pouvons alors proposer, pour traiter **CFEMF**, un processus **DRFLOT** similaire à celui de la section 2.2. Ce processus comporte en première phase un calcul (éventuellement approché) de D, suivi de l'extraction d'une solution réalisable de **CFEMF** au moyen d'un algorithme de flot entier de coût minimum.

3.4. Un exemple : le TSP Triangulaire

Soit K_n le réseau complet construit sur un ensemble X_n à n sommets, et soit c un vecteur *Coût* défini sur les arcs de K_n et satisfaisant *l'Inégalité Triangulaire.* On suppose que A est l'ensemble de tous les arcs de K_n, que K est l'ensemble de tous les couples de sommets de K_n, que $p = 0$, et que les capacités MAX, $C\min$ et $C\max$ sont respectivement infinies, nulles et infinies. On suppose enfin que chaque flot $f(x, y)$, x, y dans X_n, doit exprimer l'acheminement d'une quantité $1/n^2$ de flux depuis x vers y. Pour une telle instance de **CFEMF-OD**, on a :

- V est la longueur pour c d'une plus courte tournée du ***Voyageur de Commerce*** dans K_n ;
- $B = D = V$ = valeur optimale de **CFEMF-OD** ;
- V^* est **seulement** égale à la valeur moyenne du coût d'un arc.

3.5. Comparaison théorique des 3 schémas DRCOUP, DRFLOT et DME

Théorème 1. *On a toujours $B \geq V^*$ et $D \geq V^*$. Si de plus $MAX = +\infty$, alors on a $B = D$.*

Démonstration. Pour tout vecteur $\lambda \geq 0$, indexé sur A, notons V_λ la valeur optimale du programme **P*-Coup**$_\lambda$ obtenu à partir de **CFEMF*** par relaxation lagrangienne de la contrainte de couplage (C3). Nous savons (théorie de la dualité) que $V = \mathrm{Sup}_{\lambda \geq 0} V_\lambda$. De l'inégalité $V_\lambda \leq B_\lambda$, nous déduisons $V \leq B$. Le même raisonnement fournit $V \leq D$.

Supposons à présent $MAX = +\infty$. Dans ce cas, nous savons (Eq. (E2)) que $B = \mathrm{Sup}_{\lambda \geq 0} B_\lambda$ s'écrit aussi :

$$B = \mathrm{Sup}_{\Pi \text{ tels que } \Pi.M(G)_{\leq c}} Inf_{f \text{ tels que } C\min \leq f \leq C\max}(p.\mathrm{Sum}(f) + (c - \Pi.M(G)_A.\lceil \mathrm{Sum}(f)_A \rceil).$$

On a aussi (Eq. (E3)) : $D = \mathrm{Sup}_{\ \mu \text{ tels que } c - \mu M_{\geq 0}} \mathrm{Inf}_{\ f \geq 0 \text{ compatibles } C\min\ et\ C\max} ((c - \mu \mathrm{M})_A.\ \lceil \mathrm{Sum}(f)_A \rceil + p.\mathrm{Sum}(f))$. On voit dès lors que B et D coincident. $\square$

Nous allons à présent montrer que les cas où les trois schémas de résolution **DME**, **DRCOUP** et **DRFLOT** produisent un résultat optimal sont les mêmes. Nous introduisons pour ce faire les définitions suivantes :

> ***Configuration d'arrêt pour le schéma DME.***
> Soit (F^*, f^*) une solution réalisable de P, telle que F^* soit optimal pour **CFEMF**$_{f*}$. Soient μ^*, τ^*, λ^* trois vecteurs respectivement indexés sur X, sur E et sur A, et optimaux pour le Dual **DU**$_{f^*}$ de **CFEMF**$_{f^*}$.
> Nous disons que (F^*, f^*) et $(\mu^*, \tau^*, \lambda^*)$ sont une *configuration d'arrêt* du schéma **DME** appliqué à **CFEMF** si f^* est solution optimale du problème **P-aux**(λ^*) :
>
> {*Trouver un multiflot f, compatible avec* $C\min$ *et* $C\max$*, et qui minimise* $p.\mathrm{Sum}(f) + \lambda^*.\ \lceil \mathrm{Sum}(f)_A \rceil$}.

Configuration d'arrêt pour le schéma DRCOUP.
Soient $\lambda^* \geq 0$ un vecteur indexé sur E et nul sur A, et $(F^*,\ f^*)$ une solution optimale de **P-Coup**$_{\lambda*}$. Nous disons que $(F^*,\ f^*)$ et λ^* sont une *configuration d'arrêt* du schéma **DRCOUP** appliqué à **CFEMF** si, pour tout λ dans $\Lambda = \{\lambda \geq 0$, indexés sur E, nuls sur $E-A$, et tels qu'il n'existe pas de circuit négatif pour le coût $(c-\lambda)$ dans le réseau $(X,E^\infty)\}$, on a : $(\lambda - \lambda^*).(F^* - \lceil \mathrm{Sum}(f^*)\rceil \geq 0$. (E4)

Configuration d'arrêt pour le schéma DRFLOT.
Soient μ^* un vecteur indexé sur X et (F^*, f^*) une solution optimale du problème **P-Flot**$_{\mu^*}$. $(F^*,\ f^*)$ et μ^* sont une *configuration d'arrêt* du schéma **DRFLOT** appliqué à **CFEMF** si, pour tout μ dans $\Omega = \{\mu$ indexés sur X tels que $(c - \mu.M(G))_{E\infty 8} \geq 0\}$, on a l'inégalité : $(\mu - \mu^*).M(G))_A.\lceil \mathrm{Sum}(f^*)_A\rceil \geq 0$. (E5)

Théorème 2. *Les énoncés suivants sont équivalents :*

1. *il existe une configuration d'arrêt pour le schéma* ***DME****;*
2. *il existe une configuration d'arrêt pour le schéma* ***DRCOUP****;*
3. *il existe une configuration d'arrêt pour le schéma* ***DRFLOT****;*
4. $V = B$*;*
5. $V = D$*.*

Démonstration. De (1) déduisons (2), (3), (4) et (5). Soit (notations ci-dessus) une configuration d'arrêt $(F^*,\ f^*)$, $(\mu^*, \tau*, \lambda^*)$ du schéma **DME**. Nous pouvons supposer que λ^* est en fait indexé sur E et que $\lambda^*_{E-A} = 0$. Par dualité, il vient : $(c-\lambda*).F^* + \lambda^*.\lceil \mathrm{Sum}(f^*)\rceil + p.\mathrm{Sum}(f^*) = V_{f^*}$.

F^* est donc solution optimale du programme :

$\{$*Chercher un flot* F*, entier, tel que* $0 \leq F \leq MAX$ *et minimisant* $c.F - \lambda^*.F\}$. (E6)

Pour tout λ dans R^{+E} tel que $\lambda_{E-A} = 0$, on a : $(c-\lambda^*).F^* + \lambda^*.\lceil \mathrm{Sum}(f^*)\rceil \geq (c-\lambda).F^* + \lambda.\lceil \mathrm{Sum}(f^*\rceil$. (E7)

Le couple (F^*, f^*) étant solution réalisable de **P-Coup**$_{\lambda*}$, il découle de (E6) et du fait que f^* est solution optimale de **P-aux**(λ^*), que (F^*, f^*) sont une solution optimale de **P-Coup**$_{\lambda*}$. De $V_{f^*} \geq V \geq B \geq B_{\lambda^*}$, on déduit alors que $V = B$. Et de (E7) on déduit que (F^*, f^*) et λ^* sont une configuration d'arrêt de **DRCOUP** (*Points (4) et (2)*).

De même, on voit que $(c - \mu^*.M(G)).F^* + p.\mathrm{Sum}(f^*) = V_{f^*}$. Le vecteur F^* est donc solution optimale du programme : $\{$*Chercher* F*, entier, tel que* $F_A \geq \mathrm{Sum}(f^*)_A$*, et* $F \leq \mathrm{MAX}$*, et minimisant* $c.F - \mu^*.M(G).F\}$. (E8).

On voit aussi que pour tout vecteur μ, indexé sur X, on a : $\mu^*.M(G).F^* \leq \mu.M(G).F^*$. (E9)

De (E9) on déduit $M(G).F^* = 0$ et de (E8) on déduit :

- si e est dans A et $(c - \mu^*.M(G))_e \geq 0$, alors $F^*_e = [\mathrm{Sum}(f^*)_e]$;
- si e n'est pas dans A et si $(c - \mu^*.M(G))_e \geq 0$, alors $F^*_e = 0$;
- si $(c - \mu^*.M(G))_e < 0$ alors $F^*_e = MAX_e$.

On en déduit que $(c - \mu^*.M(G)).F^* + p.\mathrm{Sum}(f^*) = p.\mathrm{Sum}(f^*) + (c - \mu * .M(G))^+_A.\lceil \mathrm{Sum}(f^*)_A \rceil + (c - \mu^*.M(G))^-.MAX$.

Le triplet $(\mu*, \tau*, \lambda^*)$ vérifie aussi $\mu^*.M(G) - \tau^* + \lambda^* \leq c$, et maximise, sous cette condition, la quantité $\tau^*.MAX + \lambda^*.\lceil \mathrm{Sum}(f^*)] \rceil$. On déduit $(c - \mu^*.M(G)).F^* + p.\mathrm{Sum}(f^*) = p.\mathrm{Sum}(f^*) + \lambda^*.[\mathrm{Sum}(f^*)_A] + \tau^*.MAX$, et, puisque f^* est solution optimale de **P-aux**(λ^*), que le couple $(F^*,\ f^*)$, qui est déjà solution réalisable de **P-Flot**$_{\mu^*}$, en est aussi une solution optimale. De $V_{f*} \geq V \geq D \geq D_{\mu^*}$, on déduit alors $V = D$. Et de (E9) on déduit que (F^*, f^*) et λ^* définissent une configuration d'arrêt du schéma **DRFLOT** (*points (5) et (3))*.
De (3) déduisons (5).

Soit une configuration d'arrêt $(F^*,\ f^*)$ et μ^* pour le schéma **DRFLOT**. F^* et f^* sont donc une solution optimale de **P-Flot**$_{\mu^*}$ et μ^* est solution optimale du programme H :
$\{$*Trouver* μ*, indexé sur X, tel que (c-*μ*.M(G))*$_{E\infty} \geq$ *0, et qui minimise* $\mu.M(G).F^*\}$.
Le dual H^* de H s'écrit :
$\{$*Trouver* $z \geq 0$*, indexé sur E, nul sur* $E - E^\infty$*, tel que* $M(G).z + M(G).F^* = 0$*, et minimisant c.z*$\}$;
et admet une solution optimale entière z^*. Le flot $F1 = F^* + z^*$ est entier, tel que $MAX \geq F$ et que $F1_A \geq [\mathrm{Sum}(f^*)_A]$. On a aussi $c.F1 + p.\mathrm{Sum}(f^*) = c.F^* + c.z* + p.\mathrm{Sum}(f^*) = (c - \mu^*.M(G)).F^* + p.\mathrm{Sum}(f^*) = D_{\mu^*}$. $F1$ et f^* forment donc une solution réalisable de **CFEMF**, optimale puisque $D_{\mu*} \leq D \leq V$. On déduit $D = V$ (*point (5)*).
De (2) déduisons (4).
Soit une configuration d'arrêt $(F^*,\ f^*)$, λ^* pour le schéma **DRCOUP**. F^* et f^* sont donc une solution optimale du programme **P-Coup**$_{\lambda*}$ et λ^* est solution optimale du programme K suivant :
$\{$*Trouver* $\lambda \geq 0$*, indexé sur E et nul E-A, et* μ*, indexé sur X, tels que (*μ*.M(G) +* $\lambda)_{E\infty} \leq c_e$*, et qui maximisent* $\lambda.(\lceil \mathrm{Sum}(f^*)\rceil - F^*)\}$.

Le dual K^* de K s'écrit : $\{$*Trouver un flot* $u \geq 0$*, indexé sur E et dont la trace sur* $E - E^\infty$ *est nulle, tel que :*
$u_e \geq (\lceil \mathrm{Sum}(f^*)\lceil - F^*)_e$ *pour tout arc e dans* $A \cap E^\infty; (E10)$
et qui minimise la quantité c.u$\}$.
Soit u^* une solution optimale, que l'on peut choisir entière, de K^*. Le flot $F1* = F^* + u^*$ est entier, compatible avec le vecteur MAX. Si $e \in A \cap E^\infty$, alors on déduit de (E10) que $F1_e \geq \lceil \mathrm{Sum}(f^*)_e \rceil$. Dans le cas où $e \in A \cap E^\infty$, alors le programme K n'est borné (et n'admet donc λ^* comme solution optimale) que si $F^*_e \geq \lceil \mathrm{Sum}(f^*)_e \rceil$.

Le couple $(F1,\ f^*)$ est donc une solution réalisable de **CFEMF**. On a enfin que :

$$\begin{aligned} c.F1 + p.\mathrm{Sum}(f^*) = c.F^* + c.u * + p.\mathrm{Sum}(f^*) = (\text{Dualité}) c.F^* - \lambda^*.F^* \\ + \lambda^*.\mathrm{Sum}(f^*) + p.\mathrm{Sum}(f^*) = B_{\lambda^*} \leq V. \end{aligned}$$

Il s'ensuit que $B_{\lambda^*} = B = V$, c'est-à-dire le point (4).

De (4), déduisons (1).
Supposons donc $B = V$. L'ensemble des couples (F, f) susceptibles d'être solution optimale d'un problème **P-Coup**$_\lambda$ peut être considéré comme réduit à l'ensemble des sommets d'un certain polyèdre et donc comme fini. Il en découle l'existence d'un vecteur $\lambda^* \geq 0$, qui est indexé sur E et de trace nulle sur $E - A$, et d'un couple (F^*, f^*), solution optimale de **CFEMF**, tels que (F^*, f^*) est aussi solution optimale du programme **P-coup**$_{\lambda^*}$. On a alors $\lambda^*.(F^* - \lceil \mathrm{Sum}(f^*) \rceil) = 0$. Soient μ^* un vecteur indexé sur X, et $\tau^* \geq 0$ un vecteur indexé sur E qui constituent une solution duale optimale du programme :
{*Trouver un flot F (entier)* $\geq$ *0, tel que F* $\leq$ *MAX et qui minimise la quantité (c-*λ^**).F*}.
Le triplet $(\mu^*, \tau^*, \lambda^*)$ est alors solution réalisable du dual **DU**$_{f^*}$ de **CFEMF**$_{f^*}$, telle que :

$$c.F^* - (\lambda^*(\lceil \mathrm{Sum}(f^*) \rceil) - \tau^*.MAX) = (c - \lambda^*).\ F^* + \tau^*.MAX = 0,$$

et donc en fait solution optimale de **DU**$_{f*}$. Le fait que f^* soit alors solution optimale du programme :
{*Trouver un multiflot* $f \geq 0$, *tel que* $C\min \leq f \leq C\max$ *et minimisant* $p.\mathrm{Sum}(f) + \lambda^*.\mathrm{Sum}(f)$},
induit alors que (F^*, f^*) et $(\mu*, \tau*, \lambda*)$ sont une configuration d'arrêt pour le schéma **DME** (*point (1)*).
De (5) déduisons (1). L'égalité $D = V$ implique l'existence de μ^*, tel que **P-Flot**$_{\mu^*}$ admet une même solution optimale (F^*, f^*) que **CFEMF**. On a alors : $\mu * .M(G).F^* = 0$. Soit le programme :
{*Trouver F* $\geq$ *0, indexé sur E, tel que* $F_A \geq \lceil \mathrm{Sum}(f^*)_A \rceil$ *et F* $\leq$ *MAX et minimisant (c -* $\mu^*.M(G)$*). F*}.
Ce programme admet F^* comme solution optimale à laquelle correspond donc une solution duale optimale associée (τ^*, λ^*). Les vecteurs $\tau^* \geq 0$ et $\lambda^* \geq 0$ sont indexés sur E et tels que λ^*_{E-A} est nulle, et μ^*, τ^* et λ^* définissent alors une solution optimale du dual **DU**$_{f*}$ de **CFEMF**$_{f*}$. On peut en fait choisir τ^* et λ^* tels que :

$$\tau* = (\mu^*.M(G) - c)^+ \text{ et } (\lambda^*)_A = (c - \mu^*.M(G))^+_A.$$

Donc, si f est tel que $C\min \leq f \leq C\max$ et si $F(f)$ est solution optimale du programme $(\textbf{P-Flot}_{\mu^*})_f$, alors la quantité $c.F(f) + p.\mathrm{Sum}(f) - \mu^*.M.F(f)$ s'écrit $\lambda^*.\ \mathrm{Sum}(f) + p.\mathrm{Sum}(f) - \tau^*.MAX$. Il s'ensuit que f^* est solution optimale du problème **P-aux**(λ^*) et donc que (F^*, f^*) et $(\mu*, \tau^*, \lambda^*)$ sont une configuration d'arrêt de **DME** (*point (1)*). □

3.6. Une illustration des théorèmes 1 et 2

Considérons le réseau $G = (X, E)$ suivant :

$$X = \{A, B, C, D\};$$

$$E = \{[A, B], [B, A], [B, C], [C, A],$$

$$[C, D], [D, C], [D, A], [A, D], [A, C], [B, D]\}; A = E.$$

Chaque arc e dans E est pourvu d'un coût $p_e = 0{,}5$, ainsi que d'un coût c_e défini par le tableau suivant :

	$[A, B]$	$[B, A]$	$[B, C]$	$[C, B]$	$[C, D]$	$[D, C]$	$[D, A]$	$[A, D]$	$[A, C]$	$[B, D]$
c	1	0	1	0	1	0,2	0,1	4	2,5	2,5

Nous cherchons sur ce réseau un flot entier $F \geq 0$ et un multiflot $f = \{f(k), k = 1...2\} \geq 0$ tels que :

- *le flot f(1) exprime l'acheminement d'une quantité de flot 0,4 de A vers C ;*
- *le flot f(2) exprime l'acheminement d'une quantité de flot 0,6 de B vers D ;*
- *pour tout arc e dans A, on a $F_e \geq \mathrm{Sum}(f)_e$;*

et qui minimise la quantité c.F + p.(f(1) + f(2)).

Nous voyons qu'une solution optimale du problème ainsi posé est définie par :

- un flot F^* valant 1 sur les arcs du circuit (A, B, C, D, A) et 0 ailleurs ;
- un flot $f^*(1)$ valant 0,4 sur les arcs du chemin (A, B, C) ;
- un flot $f^*(2)$ valant 0,6 sur les arcs du chemin (B, C, D) ;

et que la valeur optimale associée est 3,1 + 2 = 5,1.

Le couple $(F1* = 0, f1* = f^*)$ et le vecteur λ^* ci-dessous sont une configuration d'arrêt du schéma **DRCOUP**.

	[A,B]	[B,A]	[B,C]	[C,B]	[C,D]	[D,C]	[D,A]	[A,D]	[A,C]	[B,D]
λ^*	1	0	1	0	1,1	0,1	0	4,1	2,5	2,6

La valeur B alors obtenue est elle aussi égale à 5,1. Nous sommes ici dans les conditions du théorème 1, et donc $D = 5{,}1$. Le couple (F1*, f1*) et le vecteur μ^* ci-dessous sont une configuration d'arrêt pour le schéma **DRFLOT**.

	A	B	C	D
μ^*	0	0	0	–0,1

Testons à présent le comportement sur cet exemple du schéma **DME**. À l'intérieur des tableaux ci-dessous, λ et μ désignent les composantes du vecteur dual, pour le problème **CFEMF**$_f$,qui sont respectivement indexées sur E et sur X. Les flots $f(1)$ et $f(2)$ sont initialisés de façon à minimiser la quantité $p.(f(1) + f(2))$, et sont remis à jour à chaque itération par minimisation de la quantité $\lambda.[(f(1)+f(2))]+p.(f(1)+f(2))$. Nous obtenons :

1re itération :

$f(1)$ est associé au chemin (A, C) ; $f(2)$ est associé au chemin (B, D) ; F est associé au circuit (A, C, B, D, C, D, A) ; μ est défini par le tableau :

A	B	C	D
0	1	–1,1	–0,1

λ est défini par le tableau :

[A,B]	[A,C]	[A,D]	[B,A]	[B,C]	[B,D]	[C,B]	[C,D]	[D,A]	[D,C]
0	3,6	4,1	1	3,1	3,6	0	0	0	1,2

2e itération :

$f(1)$ est associé au chemin (A, B, D, C) ; $f(2)$ est associé au chemin (B, D) ; F est associé au circuit (A, B, D, C, C, B, A) ; μ est défini par le tableau :

A	B	C	D
0	0	0	1

λ est défini par le tableau :

[A,B]	[A,C]	[A,D]	[B,A]	[B,C]	[B,D]	[C,B]	[C,D]	[D,A]	[D,C]
1	2,5	3	0	1	1,5	0	0	1,1	1,2

3e itération :

$f(1)$ est associé au chemin (A, B, C) ; $f(2)$ est associé au chemin (B, C, D) ; F est associé au circuit (A, B, C, D) ; μ est défini le tableau :

A	B	C	D
0	0,1	0,1	–0,1

λ est défini par le tableau :

[A,B]	[A,C]	[A,D]	[B,A]	[B,C]	[B,D]	[C,B]	[C,D]	[D,A]	[D, C]
0,9	2,4	4,1	–0,1	1	2,6	0	1,2	0	0

4e itération : (F, f) et (μ, λ) définissent une configuration d'arrêt.

4. Comment traiter le problème auxiliaire P-aux ?

Il s'agit ici de spécifier l'instruction (I1) du schéma **DME**, et décrire quel traitement apporter au problème **P-aux** :

P-**aux**(λ) : {*Étant donnés le réseau* $G = (X, E)$*, les familles* $C\min$ *et* $C\max$*, les vecteurs* $\lambda \geq 0$ *et* $p \geq 0$*, on cherche un multiflot* $f = \{f(k),\ k \in K\}$*, tel que* $C\min \leq f \leq C\max$ *et minimisant* $\lambda.\lceil \mathrm{Sum}(f) \rceil + p.\mathrm{Sum}(f)\}$.

Ce problème peut bien sûr être reformulé en introduisant un vecteur auxiliaire z, entier et dominant $\mathrm{Sum}(f)$, et en imposant la minimisation de $\lambda.z + p.\mathrm{Sum}(f)$. Ce faisant, on va toutefois en sens inverse du processus amorcé, qui vise à traiter $\mathrm{Sum}(f)$ comme objet principal et à le gérer en tirant parti de la structure des espaces de flots.

Afin de poursuivre dans le sens suggéré par les sections précédentes, nous devons procéder en deux temps :

- nous devons d'abord disposer (Sect. 4.1), d'un outil nous permettant de gérer la quantité $\lambda.\lceil \mathrm{Sum}(f)\rceil + p.\mathrm{Sum}(f)$, en agissant directement sur $\mathrm{Sum}(f)$, ou, si l'ensemble K est réduit, sur un des composants de f ;
- nous devons ensuite (Sect. 4.2), gérer l'articulation entre $\mathrm{Sum}(f)$ et les composants de f. Le but est ici d'éviter de n'agir sur $\mathrm{Sum}(f)$ qu'au travers de ses composantes prises de façon séparée. Dans le cas où K est grand et où les valeurs de chaque composante $f(k)_e$ ont vocation à être petites par rapport à l'unité, (ce qui est le cas pour les problèmes de la forme **CFEMF-OD** et pour les problèmes exemples de la Sect. 2), alors une perturbation de f, localisée sur une seule composante, risque en effet de n'induire rapidement aucune modification du vecteur $\lceil \mathrm{Sum}(f)\rceil$, et donc aucune amélioration de la quantité $\lambda.\lceil \mathrm{Sum}(f)\rceil + p.\mathrm{Sum}(f)\}$.

4.1. La gestion de la quantité $\lambda.\lceil \mathrm{Sum}(f)\rceil + p.\mathrm{Sum}(f)$: le problème FCEM

Il s'agit donc du problème **P-aux** considéré dans le cas où le multiflot f est en fait un flot :

> ***Problème de Flot de Coût Entier Minimum (FCEM) :*** $\{$*Étant donnés le réseau G = (X,E), 2 vecteurs capacités* $C\min$ *et* $C\max$*, deux vecteurs coûts* λ *et p, positifs ou nuls, tous indexés sur E, un vecteur flot u, on cherche alors un flot* $f \geq 0$*, tel que* $C\min \leq f \leq C\max$*, et qui minimise : p.Sum(f)b +* $\lambda.\lceil (f+u)\rceil\}$.

Proposition 2. *Le problème* ***FCEM*** *est NP-Difficile.*

Démonstration. Considérons le problème **PLO** dit de *Localisation*, connu pour être NP-complet (voir KHU72) :

> ***Problème P-LO de Localisation*** : $\{$*Étant donnés un ensemble Z, un vecteur "Coût de Construction"* α *indexé sur Z, un vecteur "Coût de Distribution"* β*, indexé sur X.X, un nombre "Seuil" S, on cherche un sous-ensemble A de Z, tel que* $\Sigma_{\ x\,\mathrm{dans}\,A}\ \alpha_x + \Sigma_{\ y\,\mathrm{dans}\,X}\ Inf_{\ x\,\mathrm{dans}\,A}\ \beta_{x,y} \leq S\}$.

À une instance (Z, S, α, β) de ce problème, associons le réseau $G = (X, E)$ suivant :

- $X = \{\mathrm{DEBUT, FIN}\} \cup Z \cup Z'$, où Z' est une copie de Z et où DEBUT et FIN sont deux sommets additionnels. Pour chaque élément x de Z, nous notons x' son homologue dans Z' ;
- $E = \{[\mathrm{FIN, DEBUT}]\} \cup \{[\mathrm{DEBUT}, x'], x' \in Z'\} \cup \{[x', y], x'$ dans Z', $y \in Z\} \cup \{(x, \mathrm{FIN}), x \in X\}$.

Résoudre **P-LO** sur l'instance (Z, S, α, β) revient à trouver sur G un flot $f \geq 0$, tel que $f([x,\text{FIN}]) = 1/\text{Card(Z)}$ pour tout x dans Z, et tel que : $\Sigma_{x \text{ dans } Z}\alpha_x$. $f([\text{DEBUT}, x']) + \Sigma_{x,y \text{ dans } Z}\beta_{x,y}$. $f([x', y]) \leq S$. Le résultat en découle. $\square$

Cycles généralisés améliorants pour le problème FCEM

Considérons à présent une instance du problème **FCEM**, conformément aux notations ci-dessus. Soient f une solution réalisable de **FCEM**, γ un cycle de G et $q > 0$ un nombre. Nous posons :

$E(\gamma,+)$ = ensemble des arcs de γ orientés dans le même sens que γ ;

$E(\gamma,-)$ = ensemble des arcs de γ orientés dans le sens opposé au sens de γ ;

et, pour tout arc e :

$$D_f(e,q) = q.p_e + \lambda_e.[\lceil (f+u)_e + q\rceil - \lceil (f+u)_e \rceil] \ ;$$
$$D^*_f(e,q) = -q.p_e - \lambda_e.[\lceil (f+u)_e\rceil - \lceil (f+u)_e - q\rceil].$$

Le couple (γ, q), dit *Cycle Généralisé*, sera *améliorant*, au sens du problème **FCEM**, pour le flot courant f, si :

- pour tout e dans $E(\gamma,+)$, $f_e + q \leq C\max_e$; pour tout e dans $E(\gamma,-)$, $f_e - q \geq C\min_e$;
- longueur$(\gamma) \geq 3$; (*Contrainte de Longueur*)
- $\Sigma_{e \text{ dans } E(\gamma,+)}\ D_f(e,q) + \Sigma_{e \text{ dans } E(\gamma,-)}\ D^*_f(e,q) < 0$.

Clairement, l'existence d'un *Cycle Généralisé Améliorant* permet de remplacer la solution courante f par $f + q.\phi_\gamma$, et d'améliorer f sans violer de contrainte. La recherche d'un tel cycle généralisé améliorant constituera donc l'épine dorsale de la procédure de traitement de **FCEM** que nous allons proposer ici. Dans cette perspective, le sens de la contrainte *de Longueur* ci-dessus est qu'un arc e de G peut être tel que $D_f(e,q) + D^*_f(e,q) < 0$, mais qu'il n'est évidemment pas possible de simultanément augmenter et diminuer la valeur de f sur un tel arc.

Remarque. L'absence d'un *Cycle Généralisé Améliorant* (γ, q) ne signifie pas que le flot courant f soit une solution optimale pour le problème **FCEM** associé. Considérons en effet l'exemple suivant :

$X = \{a,b,c,d\}; E = \{[a,b],[b,c],[c,a],[b,d],[d,a]\}$;

$p([a,b]) = 0$; $p(e) = 1$ pour tout arc e autre que $[a,b]$;

$u = 0$; $\lambda([a,b]) = -10$; $\lambda(e) = 0$ pour tout arc e autre que $[a,b]$;

$C\min = 0$; $C\max([a,b]) = 4$; $C\max(e) = 1{,}6$ pour tout arc e autre que $[a,b]$;

$f([a,b]) = 2{,}2$; $f(e) = 1{,}1$ pour tout arc e autre que $[a,b]$.

Ce flot f n'est pas optimal pour le problème **FCEM**, alors que n'existe aucun cycle généralisé améliorant f.

La recherche du couple (γ, q) peut se reformuler en termes de circuits négatifs dans un réseau auxiliaire $G^*_f = (X, E^*)$, muni d'une fonction coût C_f, dite de *Coût Généralisé*, définis comme suit :

- $E^* = \{e,\ e^{-1},\ e \text{ dans } E\}$; (si e est un arc $[x,y]$ dans E, e^{-1} désigne l'arc symétrique $[y,x]$).

- Pour e dans E, et $q > 0$, on pose :
  - $s(f,e) = C\max_e - f_e$; $s(f,e^{-1}) = f_e - C\min_e$;
  - $t(f,e) = \lceil f_e + u_e \rceil - (f_e + u_e); t(f,e^{-1}) = (f_e + u_e) - \lceil f_e + u_e \rceil$;
  - $C_f(q,e) = q.p_e + \lceil (q - t(e)) \rceil.\alpha_e$, si $q \leq s(e)$ et $C_f(q,e) = +\infty$ (un grand nombre) sinon ;
  - $C_f(q,e^{-1}) = -q.p_e - \lfloor 1 + (q - t(e^{-1})) \rfloor.\alpha_e$, si $q \leq s(e^{-1})$ et $C_f(q,e^{-1}) = +\infty$ sinon.

La recherche du couple (γ,q) devient alors celle d'un circuit élémentaire Γ dans $G{*}_f$, de longueur ≥ 3, et d'un nombre $q > 0$, tels que : $L_f(\Gamma,q) = \Sigma_{e* \text{ dans } \Gamma}$ $C_f(q,e^*) < 0$. Un tel couple (Γ,q) sera un *Circuit Négatif Généralisé.*

Recherche d'un Circuit Négatif Généralisé dans le réseau G*

Nous avons d'abord besoin d'un outil de filtrage nous permettant d'identifier les valeurs q qui sont susceptibles de faire partie d'un *Circuit Négatif Généralisé* (Γ,q) de $G{*}_f$. Afin de disposer d'un tel outil, nous associons, à chaque valeur $q \geq 0$, 2 valeurs symboliques q^+ et q^-, et étendons à ces valeurs symboliques la fonction **Coût Généralisé** C_f définie plus haut, en posant :

> $C_f(q^+,e)$ = limite de $C_f(q+\delta)$, pour $\delta- > 0^+$; $C_f(q^-,e)$ = limite de $C_f(q+\delta)$, pour $\delta- > 0^-$.

Nous obtenons alors :

Théorème 3. *Si un Circuit Généralisé Négatif (Γ,q) existe pour le réseau $G^*{}_f$ et pour la fonction C_f, alors q peut être pris dans l'Ensemble Significatif $T_f = \{t^+, t^-$ pour $t \in [0,1] \cap (\{0, 1\} \cup \{s(f,e^*), t(f,e^*), e^* \in E^*\})\}$.*

Démonstration. Si Γ est un circuit de $G{*}_f$ et si $q > 0$, t et $t' \in \{0, 1\} \cup \{s(f,e^*), t(f,e^*), e^* \in E^*\}$, sont tels que $t < q < t'$, alors $L_f(\Gamma,q)$ dépend linéairement de q entre t et t'. Si donc $L_f(\Gamma,q) < 0$, alors l'une des valeurs $L_f(G,t^+)$, $L_f(G,t')$ est négative. Si par ailleurs q est strictement plus grand que 1, alors $L_f(\Gamma,q) = L_f(\Gamma,1) + L_f(\Gamma,q-1)$. Le résultat s'en déduit. □

De ce résultat, nous pouvons déduire une procédure de recherche d'un circuit généralisé négatif, nommée **CYCLE-SEARCH**, qui parcourt T_f par valeurs croissantes et cherche, pour chaque valeur q dans T_f, un circuit négatif de G^* pour les coûts $C_f(q,e^*)$, $e^* \in E^*$. Cet algorithme, qui n'est qu'heuristique, du fait de la Contrainte de Longueur, est une adaptation de l'algorithme des plus courts chemins de Ford. Nous faisons en sorte que $G* = (X,E^*)$ soit fortement connexe, considérons un sommet de référence x_o, et faisons, pour q donné dans T_f, évoluer une antiarborescence Λ de racine x_o, et couvrant les sommets de G, comme suit :

> ***Procédure CYCLE-SEARCH***
>
> *f est donné, compatible avec les capacités $C\min$ et $C\max$, ainsi qu'une antiarborescence courante Λ de racine x_o dans le réseau $G{*}_f$;*
>
> Construire l'Ensemble Significatif T_f associé à f ; $q = 0+$; Not Stop ;
>
> Tant que (Not Stop) et ($q \leq 1$) faire
>
> > Calculer les coûts $C_f(q,e)$, $e \in E^*$;
> >
> > Pour x dans X, Calculer, au sens des coûts $C_f(q,e)$, $e \in E^*$, la distance $D_f(q,x)$ de x à x_o dans Λ ;

Laux := Nil ; Not Stop1 ; (* *Laux est une liste d'arcs qui gère les circuits négatifs de longueur* 2*)
Tant que Not Stop1 faire
Tant qu' il existe $[x, y]$ dans $E^* - \Lambda$ tel que
$D_f(q, x) > D_f(q, y) + C(q, [x, y])$ ET $[y, x] \notin \Lambda$ ET Not Stop1
Faire
Si (remplacer dans Λ l'arc sortant de x par $[x, y]$ crée un circuit négatif Γ_q)
Alors Stop1 ; Construire le cycle γ_q associé à Γ_q dans G ; Stop ; (*)
Sinon
Retirer de Λ l'arc sortant de x et Insérer l'arc $[x, y]$ dans Λ ;
Recalculer les quantités $D_f(q, x),\ x \in X$;
Si Not Stop1 ET (il existe $[y, x]$ dans $\Lambda-$ Laux et $[x, y]$ dans $E^* - \Lambda$ formant un circuit négatif)
Alors
Insérer $[y, x]$ dans Laux ; Soit e l'arc sortant de x dans Λ ;
Choisir un arc $[y, z]$ de poids C_f minimal et tel que $\Lambda - \{e,\ [y, x]\} + \{[x, y],\ [y, z]\}$ reste une antiarborescence
Sinon Stop1 ;
Si Stop alors le résultat est fourni par l'instruction (*) si Stop, et sinon Echec ;

Commentaire : le rôle de la liste auxiliaire Laux est ici de garantir la terminaison de l'algorithme, et d'éviter l'occurence d'une boucle infinie de génération des circuits négatifs de longueur 2.

Un algorithme pour le problème FCEM

Nous déduisons de ce qui précède un algorithme simple et naturel, de nature **heuristique**, pour le problème **FCEM**, que nous nommons **CYGEN** (*Cycles Généralisés*) :

Algorithme CYGEN (Résolution du problème FCEM)
Initialiser f, compatible avec les capacités $C\min$ et $C\max$; Not Stop ;
Tant que Not Stop faire
Soit Λ une antiarborescence de racine x_o dans le réseau G_f^* ; Not Stop ;
Appliquer la procédure **CYCLE-SEARCH** à f et Λ ;
Si **CYCLE-SEARCH** produit un cycle généralisé améliorant γ, q alors $f := f + q.\phi_\gamma$
Sinon Stop (le flot courant f devient le résultat fourni par le processus).

4.2. Un mécanisme d'agrégation/désagrégation pour le problème CFEMF-OD

Comme il a été dit tant en section 1 qu'au début de la section 4, nous voulons pouvoir, dans le cas ou K est grand, traiter **P-aux** en appliquant directement la procédure **CYGEN** ci-dessus au flot agrégé Sum(f). En effet, dans le cas où le multiflot f comporte un grand nombre de composants flots, dont les valeurs sont petites par rapport à l'unité, traiter séparément chaque flot $f(k)$ tend rapidement à ne plus altérer $\lceil$Sum$(f)\rceil$, c'est-à-dire à ne plus créer de regroupement autour de la ressource associée à F. Le but de cette section est donc d'appliquer directement **CYGEN** au flot agrégé Sum(f), de telle sorte que celui-ci reste décomposable comme somme des composants $f(k)$, $k \in K$, d'un multiflot f compatible avec les capacités $C\min$ et $C\max$.

Pour plus de simplicité, nous nous limitons au cas particulier du problème **CFEMF-OD**, qui est assez représentatif des situations que nous cherchons à gérer à l'aide de d'un tel mécanisme d'agrégation :

> ***Problème CFEMF-OD*** :
> {**Input** : *le réseau G = (X,E), un sous-ensemble A de E, des vecteurs coûts* $c \geq 0$ *et* $p \geq 0$*, indexés sur les arcs de G, une famille OD =* $\{(o_k, d_k), k \in K\}$ *de couples de sommets origine/destination, et une famille D =* $\{D_k$*, k dans K*$\}$ *de demandes associées.*
> ***Objectif*** *: trouver un flot* $F \geq 0$*, entier, et un multiflot f =* $\{f(k), k \in K\} \geq 0$*, tels que :*
> – *pour tout k dans K, f(k) achemine la quantité* D_k *de* o_k*, vers* d_k*, ; (f achemine D vers OD),*
> – $F_A \geq Sum(f)_A$
> *et qui minimisent la quantité c.F + p.Sum(f)*}.

Nous disons alors qu'un flot $g \geq 0$ défini sur G est *décomposable par rapport au couple (OD, D),* s'il existe un multiflot $f = \{f(k), k \text{ dans } K\} \geq 0$, qui soit *d'Acheminement* de D vers OD, et tel que Sum$(f) = g$. Il est dit *chemin-décomposable* s'il est possible de faire en sorte que chaque flot $f(k)$ soit un multiple d'un flot chemin.
Le problème **CFEMF-OD** se réécrit donc sous la forme :

> *trouver un flot* $F \geq 0$*, entier, et un flot g décomposable par rapport à D et OD, tel que* $F_A \geq g_A$*,*
> *et qui minimisent la quantité c.F + p.g*},

et le problème auxiliaire **P-aux**, qui lui est associé selon la section 3, prend la forme **FACEM (Flot Agrégé de Coût Entier Minimal)** suivante :

> {***Input*** *:* $\lambda \geq 0$ *est un vecteur coût indexé sur E et tel que* $\lambda_A = 0$.
> ***Objectif*** : *trouver un flot g décomposable par rapport à D et OD, et minimisant* $\lambda.\lceil g\rceil + p.g\}$.

Nous simplifions ces deux problèmes en faisant *l'Hypothèse de Non Bifurcation*, c'est-à-dire en imposant que le flot g soit *chemin-décomposable* par rapport à D et OD. Cette hypothèse, nous permettra de gérer la contrainte de décomposabilité à l'aide de procédures simples de recherche de chemins dans des graphes. Testée en [9], elle s'avère peu restrictive quand OD est assez grand, ce que nous pouvons confirmer grâce au résultat suivant :

Théorème 4. *Supposons G fortement connexe. On peut alors trouver une solution optimale* (F^*, f^*) *de* ***CFEMF-OD*** *, telle que au plus* Card*(E)* – Card*(X) +1 composants de f ne soient pas des flot-chemins.*

Démonstration. Considérons une solution optimale (F^*, f^*) du problème **CFEMF-OD**, et supposons que N composants de f^*, numérotés $f^*(1)...f^*(N)$, ne sont pas des flots-chemins, avec $N > \mathrm{Card}(E) - \mathrm{Card}(X) + 1$. Supposons aussi que F^* et f^* ont été choisis de telle sorte que N soit minimal et que pour cette valeur N, la somme $U = \Sigma_{n=1...N} U_n$ soit minimale, U_n étant le nombre d'arcs e de G tels que $f(n)_e \# 0$. Il existe dès lors, pour chaque $i = 1...N$, un cycle élémentaire γ_n formé d'arcs dans U_n, et les flots-cycles $\phi_{\gamma n}$, $n = 1...N$, qui sont associés à ces cycles sont linéairement dépendants. Supposons qu'on puisse écrire $\Sigma_{=1...N} t_n.\phi_{\gamma n} = 0$, où les coefficients t_n sont non tous nuls, et considérons un petit nombre $\lambda \geq 0$. On peut remplacer, pour tout $n = 1...N$, $f(n)$ par $f(n) + \lambda.t_n.\phi_{\gamma n}$. Ce faisant, on ne change Sum(f). Il suffit alors de choisir λ le plus grand possible et tel que chaque flot $f(n)$ demeure ≥ 0, pour déduire une contradiction sur la minimalité de N et de la quantité U. □

Cette contrainte *de Décomposabilité* est malheureusement difficile à gérer. Nous pouvons observer qu'une condition nécessaire de décomposabilité est fournie par la famille de contraintes dites *Inégalités de Coupes*, qui constituent un cas particulier des *Coupes Métriques* (voir [9, 21, 26]), et qui se formulent ainsi :

pour toute partie de Z de X, $\Sigma_{e \,\mathrm{sort\,de}\, Z}\, g_e \geq \Sigma_{\,k\,\mathrm{dans}\, OD(Z)} D_k$, *où OD(Z)* $= \{k \in$ *K tels que* $o_k \in$ *Z et* $d_k \in$ *X-Z*$\}$.

Cette remarque nous conduit à un traitement itératif de **FACEM**, selon lequel à chaque étape :

- un cycle généralisé améliorant (γ, q) est recherché (adaptation de la procédure **CYCLE-SEARCH**), pour le flot courant g, de telle sorte que le flot résultant $g + q.\phi_{\gamma q}$ ne viole aucune des contraintes présentes au sein d'un ensemble Δ d'*inégalités de coupes courantes* ;
- en cas de succès, une décomposition de g ainsi perturbé en flots-chemins compatibles avec D et OD est alors recherchée. En cas d'échec de cette dernière recherche, nous procédons à :
  - la génération d'inégalités de coupes insatisfaites par g, qui sont lors injectées dans la famille Δ ;
  - une correction de g de façon à ce qu'il demeure chemin-décomposable par rapport D et OD.

Ce traitement se synthétise comme suit :

Algorithme CYGEN-AGREG (traitement du problème FACEM).

1re phase : **Initialisation de** g ;
Ordonner les éléments de K par quantités D_k décroissantes ; $g := 0$; $\Delta :=$ Nil ;

Pour k dans K ainsi ordonné faire :

Calculer un plus court $(o_k - d_k)$-chemin Γ_k dans G, pour les coûts $\alpha_e = D_k.p_e + \lambda_e.(\lceil g_e + D_k \rceil - \lceil g_e \rceil)$, $e \in E$;

Ajouter à g le flot réalisant le transport de D_k depuis o_k vers d_k le long de Γ_k ;

Ajouter à Δ les coupes associées à la construction de Γ_k selon l'algorithme de Dijskstra ;

Poser $\Sigma := \{\Gamma_k,\, k \in K\}$;

2e phase : **Amélioration de g par itérations successives** ; (* SEUIL est ici un paramètre de contrôle*)

Not Stop ; Compteur := 0 ;

Tant que Not Stop faire :

soit Λ une antiarborescence de racine x_o dans le réseau G^*_g associé à g ;

appliquer la procédure **CYCLE-SEARCH** à f et Λ, en l'adaptant de façon à ne générer que des cycles généralisés (γ, q) tels que $g + \phi_\gamma$ satisfait les coupes de Δ ;

si **CYCLE-SEARCH** produit un tel cycle généralisé améliorant γ, q alors

perturbation de g et incrémentation de Δ (génération de coupes) :

$h := g + q.\phi_\gamma$; Not Stop1 ;

Tant que Not Stop1 ET $g <> h$ faire

$\delta :=$ Sup $_{e\in E}$ $\lceil (g-h)_e \rceil$;

Chercher $k \in K$ et un chemin Γ de o_k vers d_k, tels que remplacer Γ_k par Γ dans la définition de g induit une diminution stricte de la quantité Sup $_{e\in E}$ $\lceil (g-h)_e \rceil$;

Si k et Γ existent alors remplacer Γ_k par Γ dans Σ et modifier g en conséquence

Sinon

Stop1 ; Déduire, pour chaque $k \in K$, une partition Z_k, $X - Z_k$ de X associée à l'échec de la construction de Γ, (recherche de chemin dans un graphe orienté), et intégrer dans Δ l'inégalité de coupe associée à Z_k ;

Si la quantité $\lambda.\lceil g \rceil + p.g$ n'a pas diminué alors

Compteur := Compteur +1 ; Si Compteur > SEUIL alors Stop ;

Sinon Stop (**le flot courant g est le résultat fourni par le processus**) ;

4.3. Traitementde l'Instruction (I1) du Schéma DME de la section 3

Cette instruction commande, pour un multiflot courant f et un vecteur dual λ, l'exécution d'une itération d'un processus de résolution du problème **P-aux**, et ce en vue de diminuer la valeur de la quantité $\lambda.\lceil \mathrm{Sum}(f) \rceil + p.\mathrm{Sum}(f)\}$. Dans le cas du Problème **CFEM-OD**, nous pouvons la réaliser en procédant comme suit :

> **Instruction (I1)**, **Schéma DME, Problème CFEM-OD** : appliquer, à partir de $g = \text{Sum}(f)$ courant, une itération de la boucle principale de la 2^e^ phase de la procédure **CYGEN-AGREG**.

5. Tests numériques

Nous présentons ici quatre classes de tests, effectués sur un IBM Power PC 604o, 100 Mhz, 1 Go mémoire vive, à l'aide de la bibliothèque CPLEX (version déjà ancienne 6.0, *Callable Library* utilisée depuis un compilateur C, gcc 2.95.2) :

- la première évalue les schémas de résolution **DRCOUP**, **DRFLOT** et **DME** présentés en section 3 ;
- la deuxième vise à analyser le comportement de l'algorithme **CYGEN** de la section 4.1 ;
- la troisième concerne les performances de l'algorithme **CYGEN-AGREG** de la section 4.2 ;
- la quatrième vise à permettre l'analyse du comportement de la procédure heuristique **DME-CYGEN**, qui constitue l'heuristique résultant de l'intégration, au niveau de l'instruction (I1) du schéma **DME** de la section 3.1, des méthodes présentées en section 4.

Nous utiliserons d'une façon générale des graphes proches de ceux que l'on peut rencontrer dans les systèmes de transports, c'est-à-dire faiblement denses (faible rapport $\text{Card}(E)/\text{Card}(X)$), cette dernière caractéristique ne semblant avoir que peu d'impact sur le comportement des algorithmes, d'abord déterminés par les nombres d'arcs, de composants du multiflot, et les caractéristiques des vecteurs coûts et demandes.

5.1. *1^re^ catégorie de tests* : comparaison des schémas DRCOUP, DRFLOT et DME

Cette première expérimentation évalue donc les schémas **DRCOUP**, **DRFLOT** et **DME** présentés en section 3 et à les comparer. Elle ne concerne que le cas où le multiflot f est en fait un flot ($\text{Card}(K) = 1$). Les schémas **DRCOUP** et **DRFLOT** sont entièrement traités selon une séquence d'appels de la bibliothèque CPLEX sur les programmes mixtes induits par les relaxations lagrangiennes associées. Le schéma **DME** est implémenté sous la forme proposée en section 3.1, l'instruction portant sur la perturbation du multiflot f étant alors traitée à l'aide de l'algorithme **CYGEN** de la section 4 (schéma **DME-CYGEN**), et les appels à CPLEX ne concernant que la résolution du sous-problème relatif au flot entier F.

Elle vise d'abord à comparer entre elles les valeurs produites par ces schémas de résolution. Elle est réalisée sur des instances du problème **CFEMF** de petites tailles, c'est-à-dire n'induisant pas la présence de plus d'une centaine de variables entières, de façon à permettre l'obtention de résultats exacts par CPLEX,

TABLEAU 1. Descriptif des tests.

Test	CX	CE	CA	Cc	CpA	CpA^*
1	10	30	10	0,5	0,4	1,2
2	10	30	10	1	0,2	1,4
3	10	30	10	2	0,2	1,6
4	20	60	15	0,5	0,4	1,2
5	20	60	15	1	0,2	1,4
6	20	60	15	2	0,2	1,6
7	15	50	25	0,5	0,4	1,2
8	15	50	25	1	0,2	1,4
9	15	50	25	2	0,2	1,6

et présentant les caractéristiques suivantes : (rappelons que le flot entier F est indexé sur l'ensemble E des arcs de G)

$CX = \mathrm{Card}(X) \in \{10...20\}$; $CE = \mathrm{Card}(E) \in \{30...50\}$;
$\mathrm{Card}(A)/\mathrm{Card}(E) \in [0{,}25,\ 0{,}5]$; nous posons $CA = \mathrm{Card}(A)$;
$0 \leq C\min \leq 5$; $C\min \leq MAX \leq C\min + 5$;
les composantes de p et de c sont générées entre 0 et 10 : Cc est la moyenne des coûts de c, CpA est la moyenne des coûts de p sur les arcs de A, et CpA^* est la moyenne des coûts de p sur les arcs hors de A.

Les instances considérées n'ont pas de sémantique particulière.

Les quantités que nous cherchons alors à observer sont, pour chaque test :

V^* = valeur optimale de la relaxation fractionnaire du problème **CFEMF** considéré ;
V = valeur optimale du problème **CFEMF** considéré ;
V-DME = valeur produite par le schéma **DME-CYGEN** ;
V-COUP = valeur produite par le schéma **DRCOUP** ;
B = valeur produite par la phase 1 du schéma **DRCOUP** ;
V-FLOT = valeur produite par le schéma **DRFLOT** ;
D = Valeur produite par la phase 1 du schéma **DRFLOT**.

Nous ne fournissons pas non plus ici de temps CPU, dans la mesure où les valeurs associées aux schémas **DRCOUP** et **DRFLOT** n'ont pas fait l'objet d'un travail algorithmique particulier. Les autres expérimentations, qui auront pour objet de mieux tester les algorithmes présentés en section 4, contiendront ce type d'information.

Nous obtenons : (**rappel** : le nombre K de composants du multiflot f est ici égal à 1).

Commentaire : nous constatons que les valeurs B, D et V-DME sont ici en général très proches de V, ce qui signifie que les différents schémas **DRFLOT**, **DRCOUP** et **DME** constituent de bonnes approximations pour le traitement des problèmes **CFEMF** considérés : erreurs moyennes respectivement égales à 3 % et 4 %. Ceci est à pondérer toutefois avec le fait que, dans le cadre des instances traitées ici, la relaxation de la contrainte d'intégrité a fourni elle aussi des résultats

TABLEAU 2. Résultats associés.

Test	V^*	V	V-DME	V-COUP	B	V-FLOT	D
1	57,5	61,2	61,8	64,2	60,5	66,7	59,9
2	59,1	65,5	68,7	74,0	63,2	71,6	63,1
3	66,1	72,3	75,4	78,7	70,5	75,6	71,5
4	105,8	125,0	127,2	127,3	124,6	133,7	124,5
5	117,2	134,7	148,6	152,1	125,5	158,3	129,6
6	135,9	151,6	154,0	154,0	149,8	168,7	149,4
7	105,4	121,3	123,6	129,7	119,8	126,0	119,1
8	112,2	132,6	144,4	147,3	125,0	147,6	125,0
9	121,9	145,8	147,9	151,8	141,4	155,2	141,4

relativement proches des optimae (erreur moyenne de 13 %). Les phases de projection associées à **DRFLOT** et **DRCOUP** fournissent des résultats contrastés (erreurs moyennes respectives de 7 % et 9 %), sensiblement moins bons que ceux fournis par **DME-CYGEN**, ce qui explique que pour les tests de la section 5.4, qui concernent des problèmes de type **CFEM-MOD** dans lequel l'objet f est un véritable multiflot, nous nous soyons limités à l'étude du schéma **DME-CYGEN**.

Les valeurs de Cc, CpA et CpA^* n'influent par ailleurs que peu sur les écarts entre les valeurs B, D et V-DME. Elles influent par contre sur la structure des solutions et sur les coûts respectifs des flots F et f. En dehors des plages induites par ces tableaux pour les coefficients Cc, CpA et CpA^*, on constate qu'il y a soit recours systématique pour le flot f aux arcs de A (dans le cas ou le coût moyen Cc pèse sensiblement moins que la différence $CpA^* - CpA$), soit au contraire, évitement systématique des arcs de A par le flot f (dans le cas ou Cc domine sensiblement $CpA^* - CpA$), ce qui induit alors que F n'existe quasiment pas. De fait, ces plages correspondent aux valeurs des paramètres c et p qui font que le problème présente vraiment une difficulté.

5.2. 2^{e} catégorie de tests : l'algorithme CYGEN et le problème FCEM

Cette catégorie de tests concerne le problème **FCEM** défini en section 4.1. Les réseaux considérés sont générés aléatoirement de façon à présenter les caractéristiques suivantes :

$CX = \mathrm{Card}(X) \in \{20, 60\}$;
$CE = \mathrm{Card}(E) \in \{60, 400\}$;
Card(A)/Card(E) $\in [1/4, 1/2]$; nous posons $CA = \mathrm{Card}(A)$; (rappelons que le vecteur λ est nul sur $E - A$) ;
les composantes de p et λ sont générées entre 0 et 5 : $C\lambda$ est la moyenne des coûts de λ, CpA est la moyenne des coûts de p sur les arcs de A et CpA^* est la moyenne des coûts de p sur les arcs hors de A.

CYGEN considère l'ensemble significatif T_f, comme étant l'ensemble des multiples de 0,05, ce qui nous permet d'en limiter la taille, et ne semble pas avoir d'impact excessif sur le fonctionnement de l'algorithme.

TABLEAU 3. Descriptif des tests.

Test	CX	CE	CA	$C\lambda$	CpA	CpA^*
1	20	60	30	0,5	0,4	1,2
2	20	60	30	1	0,2	1,4
3	20	60	30	2	0,2	1,6
4	40	150	50	0,5	0,4	1,2
5	40	150	50	1	0,2	1,4
6	40	150	50	2	0,2	1,6
7	60	400	100	0,5	0,4	1,2
8	60	400	100	1	0,2	1,4
9	60	400	100	2	0,2	1,6

TABLEAU 4. Résultats associés.

Test	*RE*	*T-RE*	*OPT*	*T-OPT*	*V-CY*	*T-CY*	*ITER-2*
1	60,6	0,1	68,2	13	68,2	1	3,8
2	80,3	0,2	86,7	98	88,0	5	6,7
3	89,4	0,1	95, 3	48	96,3	4	4,3
4	175,0	0,5	190,2	92	194,6	9	3,4
5	208,7	1	222,7	941	224,8	48	5,0
6	237,8	0,8	246,3	680	256,3	31	6,7
7	358,3	2	404,8	851	404,8	76	3,2
8	461,7	7	492,0	4874	500,6	147	5,2
9	524,6	4	549,6	3304	576,3	120	4,0

Les quantités que nous cherchons alors à observer sont, pour chaque test :

RE = valeur optimale de la relaxation fractionnaire du problème **FCEM** (calculé avec la bibliothèque CPLEX) ;
T-RE = temps CPU associé ;
OPT = valeur optimale du problème **FCEM** (calculée directement avec la bibliothèque CPLEX) = 1 ;
T-OPT = temps CPU associé ;
V-CY = valeur trouvée par l'algorithme **CYGEN** ;
T-CY = temps CPU associé ;
ITER-2 : nombre moyen de valeurs t dans T_f testées à chaque itération de la boucle principale du programme CYGEN (nombre d'itérations de la boucle interne).

Nous obtenons les tableaux 3 et 4 (rappelons que le nombre de variables entières dans le programme linéaire associé au problème traité **FCEM** est fourni ici par CA).

Commentaires : l'algorithme **CYGEN** fournit ici en général des résultats très proches de l'optimum (erreur moyenne autour de 2 %). Les temps relativement longs pour l'exécution de **CYGEN** sont dûs au fait que son implémentation n'est pas optimisée. Par ailleurs, les remarques faites à l'issue de la section précédente à propos du rôle joué par les coefficients $C\lambda$, CpA et CpA^* valent sensiblement de la même façon ici. Deux tests seulement sur les neuf présentés ici voient **CYGEN** fournir le résultat optimum. Cette proportion n'est pas très élevée, ce qui s'explique d'une part par le fait que l'absence d'un cycle généralisé améliorant n'est qu'une condition nécessaire d'optimalité pour le problème que nous traitons ici, et d'autre part par le fait que le procédé de recherche d'un tel cycle que nous utilisons ici n'est qu'heuristique. Nous n'avons pu distinguer la part qui revient ici à chacune de ces deux causes possibles de non optimalité, car il nous aurait fallu pouvoir nous appuyer, afin de tester ce point, sur une procédure exacte de recherche de cycle généralisé améliorant, ce dont nous ne disposons pas.

5.3. 3^e^ catégorie de tests : l'algorithme CYGEN-AGREG et le problème FACEM

Cette catégorie de tests concerne le problème **FACEM** défini en section 4.2. Les réseaux considérés satisfont :

CX = Card(X) $\in \{10, 60\}$;
CE = Card(E) $\in \{30, 400\}$;
K = Card(OD) $\in \{10, 200\}$;
Card(A)/Card(E) $\in [1/4, 1/2]$; nous posons CA = Card(A) ; (rappelons que le vecteur λ est nul sur $E - A$) ;
les composantes de p et λ sont générées entre 0 et 5 : $C\lambda$ est la moyenne des coûts de λ, CpA est la moyenne des coûts de p sur les arcs de A et CpA^* est la moyenne des coûts de p sur les arcs hors de A.
Les composantes de D sont générées entre 0 et 1 : nous notons CD la moyenne des coefficients D_k, $k \in K$.

Les quantités que nous cherchons alors à observer sont, pour chaque test :

RE = valeur optimale de la relaxation fractionnaire du problème **FACEM** (calculée avec CPLEX) ;
T-RE = temps CPU associé ;
OPT = valeur optimale du problème **FACEM** (calculée avec la bibliothèque CPLEX) ;
T-OPT = temps CPU associé ;
V-ACY = valeur trouvée par l'algorithme **CYGEN-AGREG** ;
T-ACY = temps CPU associé ;
N-CO = nombres de coupes métriques gérées par le processus ;
NA = nombre d'arcs e de A portant des valeurs F_e non nulles selon **CYGEN-AGREG.**

Nous obtenons les tableaux 5 et 6 (le nombre de variables entières dans le programme linéaire associé à **FACEM** est égal à CA).

Commentaire : l'erreur induite par **CYGEN-AGREG** se situe ici autour de 5 %. La gestion de l'agrégation n'entraine pas ici un surplus d'erreur considérable par rapport au cas testé en section antérieure 5.3. Les temps d'exécution associés

TABLEAU 5. Descriptif des tests.

Test	CX	CE	CA	$C\lambda$	CpA	CpA^*	K	CD
1	20	60	30	0,5	0,4	1,2	10	0,5
2	20	60	30	1	0,2	1,4	10	0,5
3	20	60	30	2	0,2	1,6	10	0,5
4	40	150	50	0,5	0,4	1,2	50	0,2
5	40	150	50	1	0,2	1,4	50	0,2
6	40	150	50	2	0,2	1,6	50	0,2
7	60	400	100	0,5	0,4	1,2	200	0,05
8	60	400	100	1	0,2	1,4	200	0,05
9	60	400	100	2	0,2	1,6	200	0,05

TABLEAU 6. Résultats associés.

Test	RE	T-RE	OPT	T-OPT	V-ACY	T-ACY	NCO	NA
1	48,6	2	57,5	17	58,6	14	39	22
2	51,7	3	66,3	140	71,3	21	102	20
3	54,4	2	70,4	108	72,8	22	108	14
4	132,5	8	185,8	165	188,0	73	252	25
5	144,0	12	204,7	2408	220,6	165	288	24
6	149,3	11	210,9	927	222,9	102	267	17
7	267,2	64	396,0	2825	402,6	309	431	37
8	294,8	108	*Echec*	Echec	482,4	1255	602	34
9	306,9	99	455,7	9762	485,3	947	686	23

à ces tests sont assez importants, partie car nos algorithmes n'ont pas été optimisés. On remarque que, pour aucun des tests présentés ici, la valeur fournie par **CYGEN-AGREG** ne coincide avec l'optimum. Cet algorithme fonctionne en effet de façon heuristique à la fois dans sa gestion du flot agrégé (algorithme **CYGEN**), et dans sa gestion du processus d'agrégation/désagrégation. Les paramètres de coût (Cλ, CpA, CpA^*) et de demande (K, CD), exercent enfin ici une double influence :

- le traitement des instances est en général plus coûteux en temps et moins précis quand les valeurs CD sont faibles et quand les coûts $C\lambda$ équilibrent le différentiel $CpA - CpA^*$;
- le nombre d'arcs de A portant des valeurs de flots non nulles décroit sensiblement au fur et à mesure que le coût $C\lambda$ croit, et le ratio NA/CA décroit en même temps que croit CA/CE, c'est-à-dire en même temps que la part de G qui est occupée par les arcs de A.

5.4. 4e catégorie de tests : l'algorithme DME-CYGEN et le problème CFEMF-OD

Cette catégorie de tests concerne le problème **CFEM-OD**. Les réseaux considérés tels que :

> CX = Card(X) $\in \{10...60\}$; Card(E) $\in \{30...400\}$; nous posons CE = Card(E) ;
> K = Card(OD) $\in \{10...200\}$;
> Card(A)/Card(E) $\in [1/4, 2/3]$; nous posons CA = Card(A) ;
> les composantes de p et c sont générées entre 0 et 5 : nous notons Cc la moyenne des coûts de c, par CpA la moyenne des coûts de p sur les arcs de A et par CpA^* la moyenne des coûts de p sur les arcs hors de A.
> Les composantes de D sont générées entre 0 et 1 : nous notons CD la moyenne des coefficients D_k, $k \in K$.

Est principalement testée la procédure heuristique **DME-CYGEN**, qui constitue l'heuristique résultant de l'intégration, au niveau de l'instruction (I1) du schéma **DME** de la section 3.1, des méthodes **CYGEN** et **CYGEN-AGREG** présentées en section 4. Cette procédure appelle CPLEX au niveau de chaque résolution du sous-problème **CFEMF-OD**$_f$, qui porte sur le flot entier F, le multiflot f étant fixé.
Les quantités que nous cherchons alors à observer sont, pour chaque test :

> *RE* = valeur optimale de la relaxation fractionnaire du problème **CFEMF-OD** (calculée avec la bibliothèque CPLEX) ;
> *T-RE* = temps CPU associé ;
> OPT = valeur optimale du problème **CFEMF-OD** (calculée avec la bibliothèque CPLEX) ;
> *T-*OPT = temps CPU associé ;
> *V-*ACY = valeur trouvée par l'algorithme **DME-CYGEN** ;
> *T-*ACY = temps CPU associé ;
> *ITER* = nombre d'itétations de la boucle principale du schéma **DME** ;
> NA = nombre d'arcs e de A tels que F_e est non nul selon **DME-CYGEN**.

Nous obtenons les tableaux 7 et 8 (NB : le nombre de variables entières du programme linéaire associé est ici égal à CE)
Commentaires : le problème **FCEMF-OD** est structurellement assez proche du problème **FACEM**. Les temps d'éxécution de l'algorithme **DME-CYGEN** sur le problème **CFEMF-OD** sont sont d'un ordre de grandeur assez comparable, pour les mêmes instances, à ceux enregistrés pour l'algorithme **CYGEN-AGREG** sur le problème **FACEM**. Les deux algorithmes opèrent en effet sur les mêmes objets et le traitement, à chaque itération de la boucle principale de **DME-CYGEN**, de la restriction au flot entier F problème **CFEMF-OD**, ne constitue pas une surcharge considérable. À l'opposé, les temps nécessaires à l'obtention d'une valeur exacte par CPLEX pour le problème **CFEM-OD**, augmentent très sensiblement, du fait bien sûr de la contrainte de flot sur F, mais aussi du fait que le flot entier F est ici indexé sur l'ensemble des arcs du réseau G, ce qui augmente nettement le

TABLEAU 7. Descriptif des tests.

Test	CX	CE	CA	Cc	CpA	CpA^*	K	CD
1	10	30	20	0,5	0,4	1,2	10	0,5
2	10	30	20	1	0,2	1,4	10	0,5
3	10	30	20	2	0,2	1,6	10	0,5
4	20	60	30	0,5	0,4	1,2	20	0,3
5	20	60	30	1	0,2	1,4	20	0,3
6	20	60	30	2	0,2	1,6	20	0,3
7	30	90	50	0,5	0,4	1,2	40	0,2
8	30	90	50	1	0,2	1,4	40	0,2
9	30	90	50	2	0,2	1,6	40	0,2
10	60	400	100	0,5	0,4	1,2	200	0,05
11	60	400	100	1	0,2	1,4	200	0,05
12	60	400	100	2	0,2	1,6	200	0,05

TABLEAU 8. Résultats associés.

Test	*RE*	*T-RE*	*OPT*	*T-OPT*	*V-ACY*	*T-ACY*	*ITER*	*NA*
1	32,4	2	*40,3*	99	41,2	35	13	14
2	38,6	7	*48,1*	282	52,7	92	38	13
3	42,8	5	*51,9*	208	55,0	81	35	10
4	62,5	18	*79,4*	1239	81,4	122	24	18
5	71,7	29	*92*	3410	101,9	345	62	17
6	75,9,	19	*97,8*	1980	105,3	303	60	13
7	95,3	38	*117,5*	9243	121,1	157	29	24
8	113,8	257	*Echec*	Echec	143,6	906	94	21
9	122,6	89	*150,8*	17525	159,2	437	48	14
10	289,8	419	*Echec*	Echec	451,5	912	128	35
11	355,2	2337	*Echec*	Echec	545,8	3588	249	27
12	428,6	955	*Echec*	Echec	550,9	2854	301	19

nombre de variables entières apparaissant dans la formulation PLNE du problème. L'erreur moyenne commise par **DME-CYGEN** se situe ici un peu au-delà de 6 %, soit un taux sensiblement un peu plus élevé que celui enregistré pour les tests de la section précédente 5.3, ainsi qu'à à celui enregistré en section 5.1 relativement au comportement du schéma **DME**. Les remarques faites en section 5.3 à propos du rôle joué par les coefficients Cc, CpA, CpA^*, CD et K valent ici. On constate enfin une diminution du taux des arcs porteurs d'infrastructures et mesurés par

NA, qui est due à l'augmentation des coûts associés au flot F, dès lors que celui-ci est susceptible de prendre des valeurs non nulles en dehors de A.

5.5. 5e catégorie de tests : une expérimentation sur un problème réel : le routage des charges convoyées par une flotte de camions

Nous avons enfin expérimenté nos algorithmes sur une version particulière du problème dit du « *Less than Truck Load* » (voir [49]), et en adaptant un modèle initialement conçu au département de Génie Industriel de l'Université d'Oklahoma, dans le cadre d'un partenariat avec un transporteur nord-américain, opérant dans les états du sud des USA.

Le projet à l'intérieur duquel s'insère cette modélisation est vaste et complexe. Pris dans sa globalité, il porte sur le pilotage d'une flotte de camions et de remorques, qui convoient, entre différentes plate-formes logistiques fonctionnant en tant qu'interfaces entre des boucles locales et un réseau central (backbone), des chargements assemblés et désassemblés au niveau de ces plate-formes. Une fois constitués, ces chargements sont faits de remorques entières ou de containers portés par ces remorques .

À intervalles réguliers (quelques heures), l'opérateur doit, en fonction des demandes en attente et des demandes prévisionnelles, constituer les assemblages de remorques et containers qui vont être raccrochés aux camions, et router ceux-ci entre les plate-formes. Les cheminements suivis peuvent transiter par des nœuds « Relais », sur lesquels les camions peuvent échanger tout ou partie de leurs chargements.

Tel quel, le problème ainsi résumé combine de l'affectation de charges, du routage, de l'ordonnancement, et de la prise en compte d'aléas et d'incertitudes sur les demandes. Il n'a fait l'objet que de peu d'études dans la littérature, chaque transporteur nord-américain disposant de ses propres outils logiciels « *ad-hoc* ». Afin de simplifier le traitement de ce problème, et compte tenu du fait que 80 % de l'activité du transporteur dans le type de produits considérés (approvisionnements industriels) est récurrente, il a été défini un modèle agrégé qui vise à spécifier, pour une période donnée, les parcours suivis par les flux transportés et la façon dont ces parcours sont « mutualisés ». Ce modèle fait l'impasse sur les horaires et sur les mécanismes de synchronisation auxquels ces parcours vont donner lieu. Les solutions issues de ce modèle agrégé sont par la suite utilisées comme entrées pour d'autres modules qui tendent à dimensionner l'ensemble de véhicules utilisés, et à planifier de façon effective les parcours ainsi envisagés, ainsi que les échanges auxquels ils donnent lieu.

C'est donc sur ce modèle agrégé, qui s'intègre dans notre formalisme flot/multiflot avec contraintes de capacités et contraintes d'intégrité, et sur les données induites par le contexte d'application, que nous avons testé nos méthodes. Ce modèle se présente comme suit :

P désigne l'ensemble des plate-formes logistiques et R l'ensemble des nœuds *relais*. On travaille de fait dès lors sur un réseau $G = (X, E)$, tel que $X = P \cup R$, et qui présente des caractéristiques proches de celles des réseaux symétriques planaires. Pour tout arc $e = [x, y]$ de G, on connaît :

– le coût $p_e$de circulation d'un camion vide sur e (sensiblement proportionnel au temps de parcours prévisible) ;
– le coût q_e^i induit par la circulation sur l'arc e d'un container ($i = 3$) ou d'une remorque ($i = 1$) : ces derniers coûts sont des approximations, qui reflètent des surcoûts de consommation d'énergie et de qualité de service.

Pour tout couple x, y dans X, on connaît la demande $D_{x,y}$ qui doit être acheminée de x vers y durant la période considérée. Cette demande est faite de containers, qui occupent le plus souvent 1/3 de remorque), et de remorques entières, sachant qu'un camion ne peut tirer plus de 2 remorques.
Du fait de considérations sur la stabilité du système (aspect récurrent de la demande), on considère que la distribution des camions dans l'espace doit être stable au fil des périodes. Nous choisissons de modéliser ce point en assimilant la circulation des camions durant la période considérée à un flot entier $F \geq 0$, défini sur le réseau G.
L'acheminement de chaque demande $D_{x,y}$ est assimilée à un flot $f_{x,y} \geq 0$, fractionnaire, qui est un flot d'acheminement de la quantité $D_{x,y}$ de x vers y. Ce flot doit pouvoir s'écrire comme une combinaison linéaire entière de 2 flots $f_{x,y}^i$, $i \in \{1, 3\}$ dont les valeurs sont respectivement des multiples des nombres 1/2, et 1/6. L'ensemble des flots ainsi définis constitue un multiflot noté $f = (f_{x,y}^i,\ x, y \in X,\ i \in \{1, 3\})$.
Le flot F doit dominer, sur chaque arc e, le flot agrégé $\mathrm{Sum}(f) = \Sigma_{x,y \in X} f_{x,y}$. F et f doivent être choisis de telle sorte que le coût $p.F + q.\mathrm{Sum}(f)$ soit le plus petit possible.
Pratiquement, sur une période considérée, les valeurs prises par F ont vocation à être petites, ce qui rend mal adaptée l'approche qui consisterait à relaxer les contraintes d'intégrité sur F et f.

Une formulation du problème est donc la suivante : (modèle **LTL-MF** : ***Less than Truck Load Multicommodity flow***)

{Trouver, sur le réseau G_X, un flot F entier $\geq$ 0, et un multiflot $f = (f_{x,y}^i,\ i \in \{1,3\},\ x,\ y \in X) \geq 0$, tel que :

– *pour tout couple x,y $\in X$, $f^{\,i}_{\,x,y}$ achemine la demande $D_{x,y}^i$ de x vers y ;*
– *pour tout $i \in \{1,3\}$, et tout x,y $\in X$, le flot $2i.f_{\ x,y}$ est entier ; (C4)*
– *pour tout arc e, $F(e) \geq \Sigma_{x,y \in X, i \in \{1,3\},} f_{x,y}^i(e)$;*

et qui minimise la quantité $p.F + q.f.$}.

Il est important d'avoir conscience que ce modèle se contente de proposer des parcours pour les remorques et les containers, et de décrire la façon dont ces parcours peuvent être mutualisés. Il fait l'impasse sur les parcours effectifs des camions, et sur les aspects temporels, qui sont traités en aval du modèle **LTL-MF** de façon heuristique.

Le problème ainsi formulé n'est pas exactement le problème **CFEMF-OD**, puisque le multiflot f n'est pas simplement fractionnaire, mais doit satisfaire les contraintes (C4), qui peuvent être considérées comme des contraintes d'intégrité. Toutefois, il devient exactement le problème **CFEMF-OD** dès lors que l'on fait l'hypothèse du chemin unique pour le multiflot f, et l'algorithme **DME-CYGEN** s'applique dès lors complètement.

Les tests réalisés concernent un réseau réel, prêté par le Département de Génie Industriel de l'Université d'Oklahoma, qui comporte 25 sites, 73 nœuds relais et 640 arcs ($\text{Card}(X) = 25 + 73 = 98$, $\text{Card}(E) = 640$). Le nombre K de couples origines-destination actifs, potentiellement égal à 1200, est ici égal à 392. Les jeux de demandes et de coûts sont quant à eux générés aléatoirement (chaque demande $D^i_{x,y}$ se situe entre 0 et 3), les demandes et coûts réels ne pouvant être prêtés. Les quantités calculées sont alors les suivantes :

- *T-DME* = temps CPU induit par l'exécution de l'algorithme **DME-CYGEN** ;
- *T-RE-LTL* = temps CPU induit par l'application de CPLEX au modèle **LTL-MF**, relaxé de la contrainte d'intégrité sur F et f ;
- *GAP* : écart, en pourcentage, entre les valeurs fournies par **DME-CYGEN** et par l'application de CPLEX à la relaxation fractionnaire du modèle **LTL-MF**.

Nous n'avons pas été en mesure d'obtenir la valeur optimale du modèle **LTL-MF**.

Trois résultats de tests sont présentés ici, numérotés de 1 à 3. Les coûts p_e, $e \in E$, sont les mêmes pour chaque test, les coûts q_e étant nuls pour le test 1.

Test	*GAP*	*T-DME*	*T-RE-LTL*
1	10,6 %	42 min	1 h 18 min
2	9,8 %	48 min	1 h 15 min
3	8,9 %	46 min	1 h 20 min

Commentaires : cette séquence de tests avait pour objectif principal de confronter les méthodes et modèles décrits dans les sections antérieures à des problèmes de tailles réelles, et d'évaluer notamment l'impact de ce facteur de taille sur les temps d'exécution. Dans le cas présent, on constate que cet impact est de fait assez important, les temps d'exécution associés aux tests ci-dessus étant trop élevés pour une intégration du modèle **LTL-MF** à l'intérieur d'un système de décision fonctionnant en temps continu. Par contre, on peut considérer que l'utilisation du modèle **LTL-MF**, à base de flots et de multiflots, et des méthodes que nous avons décrites ici, est viable dans la perspective d'une planification stratégique des activités du système de transport, effectuée sur une base de stationnarité de ses activités. Le traitement des routages et des charges effectives en temps réel (c'est-à-dire, ici, avec des délais pour la prise de décision de l'ordre du quart d'heure), doit alors être envisagé de façon complémentaire à l'aide de processus moins lourds. Pour le reste, il ne nous était pas possible de comparer nos résultats avec ceux obtenus par les chercheurs de l'Université d'Oklahoma, car ceux-ci n'avaient pas inclus la contrainte de flot portant sur F dans leur modèle. On remarquera toutefois que le gap entre les résultats fournis respectivement par la relaxation fractionnaire et par l'algorithme **DME-CYGEN**, s'est avéré ici sensiblement moindre que ceux enregistrés lors des tests de la section 5.4, et que ce gap tend à décroitre dès lors que l'on fait croître les coûts q_e, $e \in E$. Ceci tient au fait que les valeurs prises par le multiflot f sont beaucoup plus grandes ici que dans les tests 5.4. On remarquera enfin que la méthode **DME-CYGEN**, qui est construite sur une hypothèse de routage mono-chemin, présente ici le grand avantage de prendre en compte de façon transparente les contraintes (C4) qui pèsent ici sur le multiflot f.

6. Conclusion

Nous avons présenté ici un formalisme général permettant de modéliser certains problèmes de couverture de flux par des infrastructures de transport à l'aide d'un flot « *véhicule* » entier et d'un multiflot « *usager* » fractionnaire, couplés par une contrainte de capacité. Nous avons d'abord proposé pour ces problèmes, divers schémas de décomposition et de relaxation, puis isolé un sous-problème de *Flot de Coût Entier Minimum*, et enfin introduit un mécanisme d'agrégation, qui nous ont permis de traiter le problème en considérant l'objet défini par le multiflot « *usager* » comme objet principal, et en tirant parti de la structure particulière des espaces de flots.

Ce faisant, nous avons fait apparaitre différents problèmes auxiliaires, tels que celui de la recherche d'un circuit généralisé de coût négatif ou celui de la prise en compte, dans le cadre de la recherche d'un circuit négatif, de contraintes additionnelles associées à des coupes. Ces problèmes sont complexes et nous ne les avons traités que de façon heuristique. Il serait donc intéressant de les analyser de façon approfondie. De même il serait intéressant de reprendre le problème **CFEMF** en tenant compte d'élasticités des demandes d'acheminement de flux aux coûts individuels de parcours (les coûts $p(k)$, $k \in K$), de façon à se rapprocher des réalités de certains problèmes de transport. Nous travaillerons enfin à l'avenir sur les possibilités d'applications des modèles et des techniques que nous venons d'introduire à des problèmes réels de transport et d'organisation de la production.

Références

[1] R.K. Ahuja, J.B. Orlin and D. Sharma, Multiexchance neighbourhood structures for the capacitated minimum spanning tree problem. *Math Programming* **91** (2001) 71–97.

[2] R.K. Ahuja, T.L. Magnanti, J.B. Orlin and M.R. Reddy, Applications of network optimization. Chapter 1 of *Network Models, Handbook of Operation Research and Management Science* **7** (1995)1–83.

[3] R.V. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows: Theory, Algorithms and Applications*. Prentice hall, Englewood Cliffs, N.J. (1993).

[4] J.E. Aronson, A survey on dynamic network flows. *Ann. Oper. Res.* **20** (1989) 1–66.

[5] A. Assad, Multicommodity networks flows: a survey. *Networks* **8** (1978) 37–91.

[6] A. Balakrishnan, T. Magnanti and P. Mirchandani, Designing hierarchical survivable networks. *Oper. Res.* **46** (1998) 116–130.

[7] A. Balakrishnan, T. Magnanti and P. Mirchandani, A dual based algorithm for multi level network design. *Manage. Sci.* **40** (1994) 567–580.

[8] M. Balinski, Fixed cost transportation problems. *Nov. Res. Log. Quart* **8** (1961) 41–54.
[9] F. Barahona, Network design using cut inequalities. *SIAM J. Optim.* **6** (1995) 822–837.
[10] W. Ben Ameur, Constrained length connectivity and survivable networks. *Networks* **36** (2000) 1.
[11] A. Benchakroun, J. Ferland and V. Gascon, Benders decomposition for network design problems with underlying tree structure. *Investigacion operativa* **6** (1997) 165–180.
[12] J.F. Benders, Partitionning procedures for solving mixed variable programming problems. *Num. Math.* **4** (1962) 238–252.
[13] D.Bertsimas and S. Stock Patterson, The air traffic flow problem with en route capacities. *Oper. Res.* **46-3** (1998) 406–422.
[14] D. Bienstock and O. Unluk, Capacited network design: polyedral structure and computation. *Inform. J. Comput.* **8** (1996) 243–259.
[15] T. Boffey and A. Hinxman, Solving for optimal network problem. *EJOR* **3** (1979) 386–393.
[16] A. Caminada, J.K. Hao, J.L. Lutton and V. Martin, L'optimisation des réseaux de télécommunications. *Recherche Opérationnelle et Réseaux : Méthodes d'Analyse Spatiale.* Collection IGAT, Hermes, Chap. 7 (2002) 191–236.
[17] S.G. Chang and B. Gavish, Telecommunication network topological design and capacity expansion: formulations and algorithms. *Telecom. Syst.* **1** (1993) 99–131.
[18] P. Chardaire, M.C. Costa and A. Sutter, Solving the dynamic facility location problem. *Networks* **28** (1996) 117–124.
[19] J. Chifflet, A. Lisser and P. Tolla, Interior point methods for multicommodity netflow problems. *Perquisa Operacional* **15** (1994) 1.
[20] S. Chopra and M. Rao, The Steiner tree problem I : formulations, composition and extension of facets. *Math Programming* **64** (1994) 209–229.
[21] N. Christophides, C.A. Whitlock, Network synthesis with connectivity constraint: a survey. *Oper. Res.* (1981) 705–723.
[22] J.P. Cordeau, P. Toth and D. Vigo, A survey of optimization models for train routing and scheduling. *Transportation Science* **32** (1998) 380–404.
[23] T. Crainic, M. Gendreau and J.M. Farvolden, A simplex based tabu search method for capacitated network design. *Inform. J. Comput.* **12** (2000) 223–236.
[24] T. Crainic and J.M. Rousseau, Multicommodity, multimode freight transportation: a general modeling and algorithmic framework for the service network design problem. *Transport Research B* **20** (1988) 290–297.
[25] T. Crainic, A. Frangioni and B. Gendron, Bundle based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Appl. Math.* **112** (2001) 73–99.
[26] G. Dahl and M. Stoer, A polyedral approach to multicommodity survivable network design. *Numer. Math.* **68** (1994) 149–167.
[27] P. Dejax and T. Crainic, A review of empty flow and fleet management models in freight transportation. *Transportation Science* **21** (1987) 227–247.
[28] D. De Wolf and Y. Smeers, Optimal dimensionning of pipe networks with application to gas transmission networks. *Oper. Res.* **44-4** (1996) 596–608.
[29] A. Dionne and M. Florian, Exact and approximate algorithms for optimal network design. *Networks* **9** (1979) 37–59.
[30] Z. Drezner and T. Drezner, Applied location theory models, in *Modern Methods for Business Research*, edited by Lawrence Erlbaum Mahwa, N.J. (1998) 79–120.
[31] H.A. Eiselt, G. Laporte and J.F. Thisse, Competitive location models: a framework and bibliography. *Transportation Sciences* **27** (1993) 44–54.
[32] V. Ferreira Filho and J. Galvao, A survey of computer network design problems. *Investigacion Operativa* **4** (1994) 183–211.
[33] M. Florian, *An introduction to network models used in transportation planning*, in *Transp. Plann. Models*, edited by M. Florian, North Holland, Amsterdam (1984) 137–152.
[34] J.M. Forvalden, W.B. Powell and I. Lustig, A primal partitionning solution for the arc chain formulation of a multicommodity network flow problem. *Oper. Res.* **41** (1993) 669–693.
[35] G. Gallo, Lower planes for the network design problem. *Networks* **13** (1983) 411–426.

[36] B. Gavish, Topological design of telecommunication networks: local access design methods. *Ann. Oper. Res.* **33** (1991) 17–71.
[37] A. Geoffrion, Lagrangean relaxation and its uses in integer programming. *Math. Prog. Study* **2** (1974) 82–114.
[38] A. Geoffrion, Generalized Benders decomposition. *J. Optim. Theory Appl.* **10** (1972) 237–260.
[39] A. Girard and B. Liau, Dimensioning of adaptatively routed networks. *IEE/ACM Trans. Networking* **1-4** (1993) 460–468.
[40] A. Golberg and R. Tarjan, Finding minimum cost circulation by cancelling negative cycles, *JACM* **36** (1989) 873–886.
[41] H. Hoang, Topological optimization of networks: a non linear mixed model employing generalized Benders decomposition. *IEEE Trans. Automat. Control.* **AC-27** (1982) 164–169.
[42] P.A. Hossein, D.P. Bertsekas and P. Tseng, Relaxation methods for network problems with convex arc costs. *SIAM J. Control Optim.* **5** (1987) 1219–1243.
[43] F.K. Hwang, D.S. Richards and P. Winter, The Steiner Tree Problem. North Holland (1992).
[44] P. Jaillet, G. Song and G. Yu, Airline network design and hub location problems. *Location Science* **4** (1997) 195–212.
[45] D. Johnson, J. Lenstra, A. RInnoy-Khan, The complexity of the Network Design Problem. *Networks* **8** (1978) 279–285.
[46] K.L. Jones, I.J. Lustig, J.M. Farvolden and W.B. Powel, Multicommodity network flows: the impact of formulation on decomposition. *Math. Programming* **62** (1993) 95–117.
[47] J.L. Kennington and R.V. Helgason, *Algorithms for network programming.* Wiley N.Y. (1980).
[48] B.M. Khumalala, Warehouse location problem efficient branch and bound algorithm. *Manage. Sciences B* **18** (1972) 718–731.
[49] J.G. Klincewicz and M.B. Rosenwein, Planning and consolidating shipments from a warehouse. *J. Operat. Res. Soc.* **48** (1997) 241–246.
[50] L.J. Leblanc, An algorithm for discrete network design. *Trans. Sci.* **9** (1975) 283–287.
[51] P.J. Lederer and R.S. Nambimadom, Airline network design. *Oper. Res.* **46-6** (1998) 785–804.
[52] J. Mac Gregor Smith and P. Winter, Topological Network Design. *Ann. Oper. Res.* **33** (1991).
[53] T. Magnanti, Combinatorial optimization and vehicle fleet planning: perspectives and prospects. *Networks* **11** (1981) 179–214.
[54] T. Magnanti and R.T. Wong, Network design and transportation planning models and algorithms. *Trans. Sci.* **18** (1984) 1–5.
[55] P. Mahey, A. Benchakroun and F. Boyer, Capacity and flow assignment of data networks by generalized Benders decomposition. *J. Global Optim.* **20** (2001) 173–193.
[56] A. Marin and J. Salmeron, Tactical design of rail freight networks: part 1- exact and heuristic methods. *EJOR* **90** (1996) 26–44.
[57] M. Minoux, Network synthesis and optimum network design problems: models, solution methods and application. *Networks* **19** (1989) 313–360.
[58] M. Minoux, *Optimum synthesis of a network with non simultaneous multicommodity flow requirements*, *Studies on graphs and Discrete Programming*, *Annals of Disc. Math.*, edited by P. Hansen, North Holland **11** (1981) 269–277.
[59] P.B. Mirchandani and L.R. Francis, *Discrete Location Theory.* John Wiley Eds, N.Y. (1990).
[60] I. Norenkov, Y. Yevstifeyev and V. Manichev, A method for an accelerated analysis of multiperiod electronic circuits. *Telecom Radio Engin.* **42** (1987) 123–126.
[61] A. Ouorou, P. Mahey and J.P. Vial, A survey of algorithms for convex multicommodity flow problems. *Manage. Science* **46** (2000) 126–147.
[62] P.M. Pardalos and D.Z. Du, Network design: connectivity and facility location. *DIMACS Series* **40**, N.Y., American Math Society (1998).
[63] R. Rebai, *Optimisation de réseaux de télécommunications avec sécurisation.* Thèse Paris-Dauphine (2000).

[64] T. Scott and E. Read, Modelling hydroreservoir operation in a deregulated electricity market. *ITOR* **3** (1996) 243–253.
[65] P.A. Steenbrink, *Optimization of transport networks*. Wiley, N.Y. (1974).
[66] P. Tseng, Dual Ascent methods for problems with strictly convex costs and linear constraints: a unified approach. *SIAM J. Control Optim.* **28** (1990) 214–242.
[67] R. Vijay, A. Kanda and P. Vrat, Multiperiod capacity expansion of road networks: formulation and algorithms. *Oper. Res.* **30** (1993) 117–140.
[68] R.T. Wong, Worst case analysis of network design problem heuristics. *SIAM J. Alg. Disc. Meth.* **1** (1980) 51–63.

To access this journal online:
www.edpsciences.org