

Université de Montréal

Méthodes de montée duale pour le problème de conception
de réseaux multiproduits avec coûts fixes et capacités

par

Dominique Tourillon

Département d'informatique et de recherche opérationnelle

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures

en vue de l'obtention du grade de

Maître ès sciences (M.Sc.)

en informatique option recherche opérationnelle

août 2002

©Dominique Tourillon, 2002



QA

76

U54

2002

v.037

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé:

Méthodes de montée duale pour le problème de conception de réseaux multiproduits avec coûts fixes et capacités

présenté par:

Dominique Tourillon

a été évalué par un jury composé des personnes suivantes:

Patrice Marcotte

(président-rapporteur)

Bernard Gendron

(directeur de recherche)

Jacques Ferland

(membre du jury)

Mémoire accepté le:

5 septembre 2002

Sommaire

Le problème de conception de réseaux multiproduits avec coûts fixes et capacités fait partie de la classe des problèmes NP-difficiles et aucun logiciel commercial de programmation en nombres entiers n'est efficace pour résoudre des instances de grande taille de ce problème. Dans ce mémoire, nous développons une heuristique de décomposition par montée duale dont le but est de trouver rapidement une borne inférieure de qualité à la relaxation continue de ce problème. Trois sous-problèmes, conception de réseaux sans capacité, sac à dos continu et multiflot, sont résolus l'un à la suite de l'autre de façon à toujours augmenter la valeur de l'objectif du dual.

Quand l'heuristique de décomposition par montée duale ne réussit plus à faire augmenter l'objectif du dual, nous utilisons une des deux approches suivantes, une méthode de faisceaux ou une approche primale-duale que nous avons développée, afin de modifier les variables duales et ainsi tenter de débloquer l'heuristique de décomposition par montée duale.

Les résultats obtenus par les différentes versions de notre approche de résolution sont comparés aux résultats obtenus par le logiciel commercial CPLEX et par une méthode de faisceaux sur environ deux cent instances du problème.

Mots clés : problème de conception de réseaux multiproduits avec coûts fixes et capacités, heuristique de décomposition par montée duale, approche primale-duale, méthode des faisceaux.

Abstract

The fixed-charged multicommodity capacitated network design problem is NP-hard and state-of-the-art mixed integer optimization software packages cannot solve large instances efficiently. In this thesis, we develop a dual ascent decomposition heuristic which attempts to quickly compute a good lower bound on the linear relaxation of the problem. Three subproblems, uncapacitated network design, continuous knapsack and multicommodity flow, are solved one after the other in order to always improve the objective function value of the dual.

When the dual ascent decomposition heuristic can no longer increase the objective function value of the dual, we either use a bundle method or a primal-dual heuristic that we have developed to alter the dual variables in order to restart the dual ascent decomposition heuristic.

Results of the different versions of our method are compared with results obtained by CPLEX optimization software and by a bundle method on about two hundred instances of the problem.

Keywords : fixed-charged multicommodity capacitated network design problem, dual ascent decomposition heuristic, primal-dual heuristic, bundle method.

Table des matières

Sommaire	iii
Abstract	iv
Table des matières	v
Liste des figures	viii
Liste des tableaux	ix
Remerciements	xi
Chapitre 1	
Introduction	1
1.1 Problématique et revue de littérature	1
1.2 Modèle	3
1.3 Méthodes de relaxation lagrangienne	6
1.3.1 Relaxation lagrangienne	6
1.3.2 Méthode des faisceaux	8
1.4 Plan du mémoire	10
Chapitre 2	
Méthodologie	12
2.1 Décomposition par montée duale	12
2.2 Montée duale pure	15
2.2.1 Initialisation	16

2.2.2	Sous-problème de conception de réseaux sans capacité	17
2.2.3	Sous-problème de sac à dos continu	22
2.2.4	Sous-problème de multiflot	34
2.3	Combinaison avec l'approche primale-duale	37
2.3.1	Approche primale-duale avec primal restreint	39
2.3.2	Approche primale-duale avec primal non restreint	44
2.4	Combinaison avec la méthode des faisceaux	48
2.5	Synthèse des approches de résolution	52

Chapitre 3

Analyse et résultats		54
3.1	Cadre expérimental	54
3.1.1	Environnement	54
3.1.2	Données	54
3.1.3	Plan d'expérimentation	56
3.1.4	Mesures de performance	57
3.2	Calibrage de la méthode de décomposition par montée duale	58
3.2.1	Paramètre d'initialisation τ	59
3.2.2	Ordre des sous-problèmes	60
3.2.3	Nombre d'itérations de la montée duale et du sous-problème de multiflot	62
3.2.4	Amélioration minimale de l'objectif entre deux itérations de la montée duale	66
3.2.5	Résultats de la phase de calibrage	69
3.3	Approche primale-duale	69
3.4	Méthode des faisceaux	72
3.5	Comparaison des méthodes retenues avec CPLEX et la méthode des faisceaux	75
3.5.1	Analyse des résultats en moyenne	76

3.5.2	Analyse des résultats moyens en fonction des caractéristiques des problèmes	82
3.5.3	Analyse de l'efficacité des sous-problèmes	84

Chapitre 4

Conclusion	87
-------------------	-----------

Bibliographie	89
----------------------	-----------

Annexe A : Résultats complets des tests comparatifs	xii
--	------------

Liste des figures

- 3.1 Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème c25_100_30_F_L résolu avec la méthode UPD libre. 78
- 3.2 Proportion du temps pris par chacun des trois sous-problèmes de la méthode de décomposition par montée duale sur les 196 problèmes. . . . 79
- 3.3 Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème c25_100_30_F_L résolu avec la méthode DA \Rightarrow KB. 81
- 3.4 Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème c25_100_30_F_L résolu avec la méthode DA (P_1). 84
- 3.5 Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème c30_700_400_V_L résolu avec la méthode DA (P_1). 85
- 3.6 Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème r20_220_200_0.01_2.0 résolu avec la méthode DA (P_1). 86

Liste des tableaux

3.1	Classification des problèmes quant à leur dimension.	56
3.2	Problèmes de la classe C choisis pour les tests préliminaires.	57
3.3	Problèmes de la classe R choisis pour les tests préliminaires.	57
3.4	Choix du paramètre d'initialisation τ	59
3.5	Choix de l'ordre des sous-problèmes.	61
3.6	Choix des paramètres DAiter et MFiter (stratégie A).	64
3.7	Choix des paramètres DAiter et MFiter (stratégie B).	66
3.8	Choix de l'amélioration minimale de l'objectif entre deux itérations de la montée duale (première partie).	67
3.9	Choix de l'amélioration minimale de l'objectif entre deux itérations de la montée duale (deuxième partie).	68
3.10	Choix de l'approche primale-duale.	71
3.11	Choix de la combinaison de la méthode des faisceaux et de la méthode de décomposition par montée duale (configuration P_1).	73
3.12	Choix de la combinaison de la méthode des faisceaux et de la méthode de décomposition par montée duale (configuration P_2).	74
3.13	Résultats moyens des méthodes retenues sur les 196 problèmes.	76
3.14	Résultats moyens des méthodes retenues sur les 153 problèmes de la classe R en fonction des caractéristiques des problèmes.	83

5.1	Résultats obtenus par les méthodes retenues sur les problèmes de la série C.	xvii
5.2	Résultats obtenus par les méthodes retenues sur les problèmes r10_35_10.	xviii
5.3	Résultats obtenus par les méthodes retenues sur les problèmes r10_35_25.	xix
5.4	Résultats obtenus par les méthodes retenues sur les problèmes r10_35_50.	xx
5.5	Résultats obtenus par les méthodes retenues sur les problèmes r10_60_10.	xxi
5.6	Résultats obtenus par les méthodes retenues sur les problèmes r10_60_25.	xxii
5.7	Résultats obtenus par les méthodes retenues sur les problèmes r10_60_50.	xxiii
5.8	Résultats obtenus par les méthodes retenues sur les problèmes r10_85_10.	xxiv
5.9	Résultats obtenus par les méthodes retenues sur les problèmes r10_85_25.	xxv
5.10	Résultats obtenus par les méthodes retenues sur les problèmes r10_85_50.	xxvi
5.11	Résultats obtenus par les méthodes retenues sur les problèmes r20_120_40.	xxvii
5.12	Résultats obtenus par les méthodes retenues sur les problèmes r20_120_100.	xxviii
5.13	Résultats obtenus par les méthodes retenues sur les problèmes r20_120_200.	xxix
5.14	Résultats obtenus par les méthodes retenues sur les problèmes r20_220_40.	xxx
5.15	Résultats obtenus par les méthodes retenues sur les problèmes r20_220_100.	xxxi
5.16	Résultats obtenus par les méthodes retenues sur les problèmes r20_220_200.	xxxii
5.17	Résultats obtenus par les méthodes retenues sur les problèmes r20_320_40.	xxxiii
5.18	Résultats obtenus par les méthodes retenues sur les problèmes r20_320_100.	xxxiv
5.19	Résultats obtenus par les méthodes retenues sur les problèmes r20_320_200.	xxxv

Remerciements

Je veux tout d'abord remercier mon directeur de maîtrise, le professeur Bernard Gendron, pour l'intérêt porté à mon développement académique depuis mes années au baccalauréat jusqu'à la maîtrise. Ses nombreuses idées pour améliorer mon projet et son intérêt contagieux pour la recherche m'ont grandement facilité la tâche, en plus d'augmenter et de parfaire mes connaissances en recherche opérationnelle.

Je remercie Geneviève Hernu et le professeur Antonio Frangioni pour leurs précieux conseils à propos de la méthode des faisceaux, ainsi que Renée Touzin pour ses connaissances en \LaTeX .

Je tiens aussi à remercier les membres du personnel du Département d'informatique et de recherche opérationnelle (DIRO) et du Centre de recherche sur les transports pour les cinq années enrichissantes que j'y ai passées.

Un gros merci aux organismes suivants pour leur soutien financier qui m'a permis de poursuivre sans soucis mes études au deuxième cycle :

- le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) ;
- le Fonds pour la formation de chercheurs et l'aide à la recherche (FCAR) ;
- la compagnie Giro Inc. ;
- le Centre de recherche sur les transports de l'Université de Montréal ;
- la firme d'avocats Davies, Ward, Phillips et Vineberg ;
- la Banque Nationale du Canada.

Finalement, je remercie tous mes amis du DIRO ainsi que les membres de ma famille pour leur constant soutien moral.

Chapitre 1

Introduction

1.1 Problématique et revue de littérature

L'intérêt des problèmes de conception de réseaux multiproduits avec coûts fixes réside dans leurs nombreuses applications dans les domaines du transport et des télécommunications (voir Magnanti et Wong [25], Minoux [26], Crainic [6]). La version sans capacité de ce problème a été étudiée depuis longtemps et plusieurs méthodes efficaces ont déjà été développées (voir Balakrishnan, Magnanti et Wong [1], Holmberg et Hellstrand [21], Magnanti, Mireault et Wong [24]). Parmi celles-ci, notons la méthode de montée duale développée par Balakrishnan, Magnanti et Wong [1], qui sera décrite en détail au chapitre 2.

La version avec capacités du problème de conception de réseaux multiproduits avec coûts fixes (PCMC) est en général plus représentative de la réalité, mais pose par contre d'importants défis de recherche notamment parce que les contraintes de capacité lient les flots des différents produits sur chaque arc. Ce problème fait partie de la classe des problèmes NP-difficiles ; on ne connaît donc pas à ce jour de méthode exacte qui puisse le résoudre en un temps polynomial (c'est-à-dire proportionnel à un polynôme en fonction de la taille d'une instance quelconque) et il est peu probable qu'il en existe une, à moins que $P = NP$ (voir Brassard et Bratley [3]). De plus, les logiciels commerciaux d'optimisation, peu adaptés à la structure particulière du PCMC, sont généralement inefficaces pour résoudre des instances de grande taille de ce problème (voir Chouman, Crainic et Gendron [4]). Pour être en mesure de résoudre ces dernières efficacement, il s'avère nécessaire de développer des méthodes spécialisées afin de calculer rapidement

de bonnes approximations de leurs valeurs optimales.

Les résultats impressionnants obtenus par l’heuristique de montée duale de Balakrishnan, Magnanti et Wong [1] pour la version sans capacité du problème nous ont poussé à utiliser une approche similaire pour le PCMC. Le choix d’une telle approche a aussi été motivé par de nombreuses autres méthodes de montée duale ayant obtenu de bons résultats sur différents problèmes apparentés au PCMC : les problèmes de localisation (Gendron et Crainic [13], Sridharan [28], Van Roy et Erlenkotter [30], Erlenkotter [9], Guignard et Spielberg [19], Guignard [17], Guignard et Opaswongkarn [18], Van Roy [29]), le problème du voyageur de commerce (Fischetti et Toth [10]), le problème de flot multiproduits à coût minimum (Barnhart [2]), les problèmes de conception de réseaux de télécommunications (Murthy et Seo [27]). La méthode de *décomposition par montée duale*, que nous développons dans ce mémoire, tente de calculer *rapidement* des bornes inférieures de qualité en résolvant une suite de sous-problèmes tirés du dual de la relaxation linéaire du problème. Chacun de ces sous-problèmes est plus simple à résoudre que la relaxation linéaire puisqu’il traite un sous-ensemble strict des variables duales. La méthode génère une suite non décroissante de bornes inférieures, mais peut néanmoins ne pas converger vers la valeur optimale de la relaxation linéaire. Afin de pallier à ce problème, nous proposons de combiner notre méthode à d’autres techniques (approche primale-duale ou méthode des faisceaux), lorsqu’elle ne réussit plus à faire augmenter l’objectif du dual.

Notons que plusieurs travaux ont porté sur des méthodes de calcul de bornes inférieures pour le PCMC, notamment des approches basées sur la relaxation lagrangienne, qui peuvent être utilisées dans des méthodes d’énumération de type “branch-and-bound” : Gendron et Crainic [12, 14], Gendron, Crainic et Frangioni [15], Crainic, Frangioni et Gendron [7], Holmberg et Yuan [22]. Notre méthode de décomposition par montée duale s’inscrit dans cette ligne de recherche, bien qu’elle propose une nouvelle approche pour calculer des bornes inférieures. Chouman, Crainic et Gendron [4] ont aussi réalisé une étude polyédrale du PCMC afin de trouver des familles d’inégalités valides qui améliorent les relaxations du problème.

Étant donné la difficulté du PCMC, plusieurs chercheurs ont développé des approches heuristiques pour trouver de bonnes solutions réalisables au problème. En particulier, Ghamlouche, Crainic et Gendreau [16] et Crainic, Gendreau et Farvolden [8] ont utilisé la recherche avec tabous, tandis que Hernu [20] a proposé une heuristique basée sur des méthodes de programmation mathématique.

Dans la suite du chapitre 1, nous introduisons le modèle utilisé par notre approche de résolution du PCMC, ainsi que certaines méthodes de programmation mathématique évoquées dans la suite du mémoire. Un plan du mémoire décrit finalement le contenu des chapitres subséquents.

1.2 Modèle

Étant donné un graphe orienté $G = (N, A)$, où N est l'ensemble des noeuds et A est l'ensemble des arcs, et un ensemble K de produits qui représente les demandes à satisfaire. Chaque produit $k \in K$ correspond à une paire de noeuds origine-destination. L'objectif consiste à satisfaire la demande à coût minimum, étant donné deux types de coûts non-négatifs :

- un *coût de transport* c_{ij}^k , qui doit être payé pour chaque unité de flot du produit k circulant sur chaque arc (i, j) ;
- un *coût fixe* f_{ij} , qui doit être payé pour chaque arc (i, j) où le flot est non-nul.

On définit $d^k > 0$ comme la demande à satisfaire entre l'origine $O(k)$ et la destination $D(k)$ pour chaque produit k . Sur chaque arc (i, j) , on impose une capacité $u_{ij} > 0$ et une borne supérieure $b_{ij}^k = \min\{d^k, u_{ij}\}$ sur la quantité de flot du produit k . Notons que $\sum_{k \in K} b_{ij}^k \geq u_{ij}, \forall (i, j) \in A$.

Pour formuler le modèle du PCMC, on définit les variables x_{ij}^k , représentant la quantité de flot acheminée sur chaque arc (i, j) pour chaque produit k , et les variables y_{ij} , qui prennent la valeur 0 si l'arc (i, j) est fermé (inutilisé) et la valeur 1 si l'arc (i, j)

est ouvert (utilisé). Le modèle est alors formulé ainsi :

$$Z = \min \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij},$$

Sujet à :

$$\sum_{j \in N \mid (j,i) \in A} x_{ji}^k - \sum_{j \in N \mid (i,j) \in A} x_{ij}^k = \begin{cases} -d^k, & i = O(k), \\ d^k, & i = D(k), \\ 0, & i \neq O(k), D(k), \end{cases} \quad \begin{matrix} \forall i \in N, \\ k \in K, \end{matrix} \quad (1.1)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i,j) \in A, \quad (1.2)$$

$$x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall (i,j) \in A, k \in K, \quad (1.3)$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, k \in K, \quad (1.4)$$

$$y_{ij} \in \{0, 1\}, \quad \forall (i,j) \in A. \quad (1.5)$$

Les contraintes de *conservation de flot* (1.1) assurent que la demande soit satisfaite pour chacune des paires origine-destination et que le flot entrant soit égal au flot sortant pour chacun des noeuds intermédiaires. Les contraintes de *capacité mutuelle* (1.2) assurent que le flot soit nul sur chaque arc (i,j) fermé, et inférieur à la capacité de l'arc sur chaque arc (i,j) ouvert. Les contraintes de *capacité individuelle* (1.3) assurent que le flot soit nul sur chaque arc (i,j) fermé, et inférieur à la capacité pour chaque produit k sur chaque arc (i,j) ouvert. Les contraintes (1.3) sont redondantes, mais peuvent être fort utiles pour améliorer les bornes inférieures obtenues par des méthodes de relaxation. Notons qu'en enlevant les contraintes (1.3), nous obtenons la formulation dite *faible* du PCMCC. Les variables y_{ij} sont des variables entières 0-1 tandis que les variables x_{ij}^k sont continues et non-négatives.

Relaxation continue

L'algorithme que nous présentons résout la relaxation continue du modèle, c'est-à-dire que les contraintes (1.5) sont remplacées par

$$y_{ij} \geq 0 \quad \text{et} \quad y_{ij} \leq 1, \quad \forall (i,j) \in A.$$

De plus, nous ajoutons les contraintes de *capacité pure* $\sum_{k \in K} x_{ij}^k \leq u_{ij}, \forall (i, j) \in A$, qui sont pourtant redondantes puisque les contraintes de borne $y_{ij} \leq 1, \forall (i, j) \in A$, sont présentes. Les contraintes de capacité pure sont néanmoins ajoutées pour pouvoir traiter des sous-problèmes différents où nous activerons à tour de rôle un de ces deux types de contraintes, selon la structure des sous-problèmes à résoudre.

Le modèle associé à la relaxation continue du PCMC peut donc être formulé ainsi :

$$Z_p = \min \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij}, \quad (1.6)$$

Sujet à :

$$\sum_{j \in N | (j,i) \in A} x_{ji}^k - \sum_{j \in N | (i,j) \in A} x_{ij}^k = \begin{cases} -d^k, & i = O(k), \\ d^k, & i = D(k), \\ 0, & i \neq O(k), D(k), \end{cases} \quad \begin{matrix} \forall i \in N, \\ k \in K, \end{matrix} \quad (\pi_i^k), \quad (1.7)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i, j) \in A, \quad (\alpha_{ij} \geq 0), \quad (1.8)$$

$$x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall (i, j) \in A, k \in K, \quad (\beta_{ij}^k \geq 0), \quad (1.9)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij}, \quad \forall (i, j) \in A, \quad (\gamma_{ij} \geq 0), \quad (1.10)$$

$$y_{ij} \leq 1, \quad \forall (i, j) \in A, \quad (\lambda_{ij} \geq 0), \quad (1.11)$$

$$x_{ij}^k \geq 0, \quad \forall (i, j) \in A, k \in K,$$

$$y_{ij} \geq 0, \quad \forall (i, j) \in A.$$

Aux contraintes de conservation de flot (1.7), de capacité mutuelle (1.8), de capacité individuelle (1.9), de capacité pure (1.10) et de borne sur les variables de conception (1.11), nous associons respectivement les variables duales π_i^k (non restreintes), $\alpha_{ij} \geq 0$, $\beta_{ij}^k \geq 0$, $\gamma_{ij} \geq 0$ et $\lambda_{ij} \geq 0$. Notons également que pour chaque produit k , une des contraintes de conservation de flot (1.7) est redondante ; nous avons donc arbitrairement fixé les variables duales $\pi_{O(k)}^k$ à zéro.

Le dual de ce problème, sur lequel se base notre algorithme de montée duale, peut

être formulé ainsi :

$$Z_d = \max \sum_{k \in K} d^k \pi_{D(k)}^k - \sum_{(i,j) \in A} u_{ij} \gamma_{ij} - \sum_{(i,j) \in A} \lambda_{ij}, \quad (1.12)$$

Sujet à :

$$\pi_j^k - \pi_i^k - \alpha_{ij} - \beta_{ij}^k - \gamma_{ij} \leq c_{ij}^k, \quad \forall (i,j) \in A, k \in K, \quad (x_{ij}^k \geq 0), \quad (1.13)$$

$$-\lambda_{ij} + u_{ij} \alpha_{ij} + \sum_{k \in K} b_{ij}^k \beta_{ij}^k \leq f_{ij}, \quad \forall (i,j) \in A, \quad (y_{ij} \geq 0), \quad (1.14)$$

$$\alpha_{ij}, \gamma_{ij}, \lambda_{ij} \geq 0, \quad \forall (i,j) \in A,$$

$$\beta_{ij}^k \geq 0, \quad \forall (i,j) \in A, k \in K.$$

Le premier type de contraintes (1.13) est associé aux variables primales $x_{ij}^k \geq 0$, tandis que le deuxième type de contraintes (1.14) est associé aux variables primales $y_{ij} \geq 0$.

1.3 Méthodes de relaxation lagrangienne

Cette section vise à introduire deux méthodes de programmation mathématique, la relaxation lagrangienne et la méthode des faisceaux, qui seront utilisées dans la suite du mémoire.

1.3.1 Relaxation lagrangienne

Soit le problème suivant :

$$(PE) \quad Z_{PE} = \min \quad cx,$$

Sujet à :

$$A^1 x = b^1, \quad (1.15)$$

$$A^2 x \leq b^2, \quad (1.16)$$

$$x \in X,$$

où A^1 et A^2 sont des matrices $m^1 \times n$ et $m^2 \times n$, b^1 et b^2 sont des vecteurs de dimensions m^1 et m^2 et c et x sont des vecteurs de dimension n . Nous allons présenter la méthode

de relaxation lagrangienne associée à chacun des deux ensembles de contraintes (1.15) et (1.16).

Relaxation des contraintes d'égalité

En relaxant les contraintes d'égalité (1.15), nous transférons ces contraintes dans l'objectif en leur associant un vecteur de multiplicateurs $\lambda^1 \in \mathbb{R}^{m^1}$. Nous obtenons alors le sous-problème lagrangien suivant :

$$(RL_1(\lambda^1)) \quad Z_{RL_1}(\lambda^1) = \min cx + \lambda^1(A^1x - b^1),$$

Sujet à :

$$A^2x \leq b^2,$$

$$x \in X.$$

L'objectif $Z_{RL_1}(\lambda^1)$ fournit une borne inférieure sur Z_{PE} puisque :

$$\begin{aligned} Z_{RL_1}(\lambda^1) &= \min\{cx + \lambda^1(A^1x - b^1) \mid A^2x \leq b^2, x \in X\} \\ &\leq \min\{cx + \lambda^1(A^1x - b^1) \mid A^2x \leq b^2, A^1x = b^1, x \in X\} \\ &= \min\{cx \mid A^2x \leq b^2, A^1x = b^1, x \in X\} \\ &= Z_{PE}. \end{aligned}$$

Nous définissons le dual lagrangien relativement aux contraintes (1.15) comme étant le problème qui consiste à trouver la meilleure (plus grande) borne inférieure :

$$\max_{\lambda^1} Z_{RL_1}(\lambda^1).$$

La fonction $Z_{RL_1}(\lambda^1)$ est concave, mais non différentiable. On peut donc utiliser une méthode de sous-gradients ou de faisceaux (ou toute autre méthode d'optimisation non différentiable, voir LeMaréchal [23]) pour résoudre le dual lagrangien. Mentionnons que ces méthodes utilisent la propriété suivante : si \bar{x} est une solution optimale du sous-problème lagrangien $(RL_1(\bar{\lambda}^1))$, alors $A^1\bar{x} - b^1$ est un sous-gradient de $Z_{RL_1}(\lambda^1)$ en $\lambda^1 = \bar{\lambda}^1$.

Relaxation des contraintes d'inégalité

En relaxant les contraintes d'inégalité (1.16), nous transférons ces contraintes dans l'objectif en leur associant un vecteur de multiplicateurs $\lambda^2 \in \mathbb{R}^{m^2}$, $\lambda^2 \geq 0$. Nous obtenons alors le sous-problème lagrangien suivant :

$$(RL_2(\lambda^2)) \quad Z_{RL_2}(\lambda^2) = \min cx + \lambda^2(A^2x - b^2),$$

Sujet à :

$$A^1x = b^1,$$

$$x \in X.$$

L'objectif $Z_{RL_2}(\lambda^2)$ fournit lui aussi une borne inférieure sur Z_{PE} puisque :

$$\begin{aligned} Z_{RL_2}(\lambda^2) &= \min\{cx + \lambda^2(A^2x - b^2) \mid A^1x = b^1, x \in X\} \\ &\leq \min\{cx + \lambda^2(A^2x - b^2) \mid A^1x = b^1, A^2x \leq b^2, x \in X\} \\ &\leq \min\{cx \mid A^1x = b^1, A^2x \leq b^2, x \in X\} \\ &= Z_{PE}. \end{aligned}$$

Le dual lagrangien relativement aux contraintes (1.16) est le problème consistant à trouver la meilleure (plus grande) borne inférieure :

$$\max_{\lambda^2 \geq 0} Z_{RL_2}(\lambda^2).$$

La fonction $Z_{RL_2}(\lambda^2)$ est concave, mais non différentiable. On peut donc ici aussi utiliser une méthode de sous-gradients ou de faisceaux pour résoudre ce dual lagrangien. De même, si \bar{x} est une solution optimale du sous-problème lagrangien $(RL_2(\bar{\lambda}^2))$, alors $A^2\bar{x} - b^2$ est un sous-gradient de $Z_{RL_2}(\lambda^2)$ en $\lambda^2 = \bar{\lambda}^2$.

1.3.2 Méthode des faisceaux

Dans le cadre de ce mémoire, nous utilisons la méthode des faisceaux afin de résoudre les deux lagrangiens issus soit de la relaxation lagrangienne des contraintes de conser-

vation de flot (1.7), soit de la relaxation lagrangienne des contraintes de capacité mutuelle (1.8) et individuelle (1.9).

Dans les deux cas, les duaux lagrangiens ont la forme suivante :

$$\max_{\omega \in \Omega} f(\omega),$$

où $f(\omega)$ est une fonction concave, non différentiable et finie sur Ω .

Soit $\bar{\omega}$ la valeur courante de ω ; la méthode des faisceaux répète les cinq étapes suivantes à chaque itération :

1. trouver une direction de montée potentielle d à l'aide du faisceau, construit à partir de l'ensemble des sous-gradients obtenus lors des itérations précédentes ;
2. choisir un pas t ;
3. déterminer un sous-gradient de f en $\bar{\omega} + td$ à ajouter au faisceau ;
4. changer le point courant $\bar{\omega}$ pour $\bar{\omega} + td$ si $f(\bar{\omega} + td) > f(\bar{\omega})$;
5. vérifier un critère d'arrêt.

Le faisceau de sous-gradients vise à fournir une approximation la plus précise possible de la fonction f de façon à obtenir une direction de montée à l'étape 1. Comme le problème résolu par la méthode des faisceaux est un dual lagrangien, il est facile de déterminer un sous-gradient à l'étape 3, tel que mentionné dans la section précédente.

Si la sélection de la direction de montée potentielle obéit à certaines conditions, on peut montrer qu'il y a convergence finie de la méthode vers une solution optimale. De plus, selon Crainic, Frangioni et Gendron [7], la méthode des faisceaux est plus robuste et converge plus rapidement qu'une méthode de sous-gradients simple lorsqu'on l'applique aux relaxations lagrangiennes de notre modèle. Voilà pourquoi nous avons opté pour l'utilisation de cette méthode, dont une description plus détaillée peut être trouvée dans la thèse de doctorat de Frangioni [11].

1.4 Plan du mémoire

Dans le chapitre 2, nous présentons les concepts méthodologiques qui ont mené à l'implantation de notre approche de résolution. Dans un premier temps, nous présentons l'approche générale de décomposition par montée duale, avant de nous attarder sur l'implantation spécifique de cette méthode pour résoudre le PCMC. Puis, nous présentons deux méthodes pouvant être combinées à notre méthode de montée duale afin de redémarrer celle-ci lorsqu'elle ne réussit plus à faire augmenter l'objectif du dual : l'approche primale-duale et la méthode des faisceaux.

Dans le chapitre 3, nous présentons et analysons les différents tests effectués sur 44 instances du problème de conception de réseaux multiproduits avec coûts fixes et capacités afin de déterminer les meilleures valeurs pour chacun des paramètres de notre méthode de résolution. Nous comparons par la suite les résultats de notre méthode de résolution (avec les meilleurs paramètres identifiés dans la première série de tests) avec ceux obtenus par d'autres méthodes de résolution (CPLEX [5] et la méthode des faisceaux [11]) sur un ensemble d'environ deux cent instances du problème.

Finalement, nous concluons dans le chapitre 4 par un bref rappel des points importants discutés dans le cadre de ce mémoire et nous suggérons quelques avenues de recherche future.

Les contributions principales de ce mémoire sont d'ordre méthodologique. Bien que plusieurs auteurs aient déjà développé des méthodes de montée duale pour résoudre des problèmes apparentés au PCMC, notre méthode de décomposition par montée duale, présentée à la section 2.1, constitue à notre connaissance la première présentation d'un cadre général d'utilisation de ce genre de méthode. De plus, l'application de la méthode de décomposition par montée duale au PCMC constitue la première utilisation d'une méthode de *montée duale* pour résoudre ce problème particulier. Les relaxations du PCMC que nous résolvons dans les sous-problèmes de la méthode de décomposition par montée duale sont déjà connues dans la littérature scientifique ; par contre, il s'agit de la première utilisation simultanée de ces trois relaxations dans le cadre d'une même

méthode de résolution. L'utilisation d'une approche primale-duale est généralement suggérée dans la littérature pour redémarrer les méthodes de montée duale appliquées aux problèmes apparentés au PCMC; il était donc naturel d'explorer cette voie dans le cadre de notre méthode de résolution. Nous avons toutefois développé deux versions de l'approche primale-duale (la première étant très similaire à ce que l'on retrouve dans la littérature et la deuxième ayant été adaptée pour tenir compte des particularités du PCMC, notamment les contraintes de capacité). Finalement, notons que la combinaison d'une méthode de montée duale avec une méthode de faisceaux constitue une autre contribution de ce mémoire.

Chapitre 2

Méthodologie

L'algorithme que nous proposons pour résoudre la relaxation continue du PCMC se décompose en deux parties. Dans un premier temps, une méthode itérative de montée duale tente d'augmenter l'objectif du dual à chaque itération tout en maintenant la réalisabilité duale. Dans un deuxième temps, pour tenter de redémarrer la montée duale pure lorsque celle-ci ne réussit plus à faire augmenter l'objectif du dual, nous avons recours à l'une des deux méthodes suivantes : la *méthode des faisceaux* ou l'*approche primale-duale*.

Dans la première section de ce chapitre, nous présentons le cadre général de notre méthode de décomposition par montée duale. Notre algorithme de montée duale pure, qui utilise ce cadre général pour résoudre le PCMC, est présenté dans la section suivante. Nous présentons ensuite, dans les deux dernières sections du chapitre, comment l'approche primale-duale et la méthode des faisceaux ont été combinées à notre algorithme de montée duale pure afin de le faire redémarrer.

2.1 Décomposition par montée duale

Soit le problème suivant et son dual :

$$\begin{array}{l|l} (P) & Z_p = \min \quad cx, \\ & \text{Sujet à : } A^l x \geq b^l, \quad \forall l \in L, \\ & x \geq 0. \\ \hline (D) & Z_d = \max \sum_{l \in L} \lambda^l b^l, \\ & \text{Sujet à : } \sum_{l \in L} \lambda^l A^l \leq c, \\ & \lambda^l \geq 0, \quad \forall l \in L. \end{array}$$

La stratégie générale de la décomposition par montée duale consiste à traiter, dans chacun des sous-problèmes d'une suite de k sous-problèmes, un sous-ensemble des variables λ du modèle (D) de façon à faire augmenter l'objectif du dual. Ainsi, la résolution de chacun des sous-problèmes est moins complexe que la résolution du problème général, puisqu'elle traite des problèmes dont la structure est simplifiée par le fait que certaines variables duales soient fixées. Les variables duales λ sont donc injectées dans les sous-problèmes, où elles sont, à tour de rôle, soit fixées, soit modifiables.

Soit $\tilde{\lambda}$ la solution duale courante du problème (D), \tilde{Z}_d la valeur de l'objectif du dual pour cette solution et $\tilde{L} \subset L$ un sous-ensemble strict des variables duales du problème (D). On construit les sous-problèmes en fixant les variables duales $\lambda^l, l \in \tilde{L}$, à leurs valeurs respectives dans la solution réalisable $\tilde{\lambda}$:

$$\begin{array}{l|l}
 (P(\tilde{L}, \tilde{\lambda})) & (D(\tilde{L}, \tilde{\lambda})) \\
 Z_p(\tilde{L}, \tilde{\lambda}) = \sum_{l \in \tilde{L}} \tilde{\lambda}^l b^l + \min(c - \sum_{l \in \tilde{L}} \tilde{\lambda}^l A^l)x, & Z_d(\tilde{L}, \tilde{\lambda}) = \sum_{l \in \tilde{L}} \tilde{\lambda}^l b^l + \max \sum_{l \in L \setminus \tilde{L}} \lambda^l b^l, \\
 \text{Sujet à :} & \text{Sujet à :} \\
 A^l x \geq b^l, \quad \forall l \in L \setminus \tilde{L}, & \sum_{l \in L \setminus \tilde{L}} \lambda^l A^l \leq c - \sum_{l \in \tilde{L}} \tilde{\lambda}^l A^l, \\
 x \geq 0. & \lambda^l \geq 0, \quad \forall l \in L \setminus \tilde{L}.
 \end{array}$$

Remarquons que toute solution réalisable du sous-problème ($D(\tilde{L}, \tilde{\lambda})$) est aussi une solution réalisable du problème (D) puisque ($D(\tilde{L}, \tilde{\lambda})$) est une restriction de (D). De plus, on remarque que les valeurs des objectifs des problèmes ($D(\tilde{L}, \tilde{\lambda})$) et (D) sont identiques pour un même vecteur λ donné. Par conséquent, pour effectuer une *montée duale* dans le problème général (D), il suffit de trouver une solution réalisable du sous-problème ($D(\tilde{L}, \tilde{\lambda})$) qui a une valeur d'objectif $\tilde{Z}_d(\tilde{L}, \tilde{\lambda})$ supérieure ou égale à celle de l'objectif de la solution courante, \tilde{Z}_d .

Heuristiques de montée duale

Remarquons d'abord que la solution réalisable du problème (D), $\tilde{\lambda}$, est aussi une solution réalisable du sous-problème ($D(\tilde{L}, \tilde{\lambda})$) puisqu'on forme ($D(\tilde{L}, \tilde{\lambda})$) avec $\tilde{\lambda}$ et que

la seule restriction supplémentaire du sous-problème $(D(\tilde{L}, \tilde{\lambda}))$ par rapport au problème (D) consiste à garder constantes les variables $\lambda^l, l \in \tilde{L}$.

La solution $\tilde{\lambda}$ peut donc servir de solution initiale pour une heuristique de montée duale appliquée au sous-problème $(D(\tilde{L}, \tilde{\lambda}))$. Cette heuristique ne pourra, par construction, que faire augmenter l'objectif du dual ou le laisser inchangé par rapport à sa valeur initiale. Or, comme les valeurs des objectifs des problèmes (D) et $(D(\tilde{L}, \tilde{\lambda}))$ sont identiques pour un même λ , l'objectif de $(D(\tilde{L}, \tilde{\lambda}))$ est initialisé à la valeur courante \tilde{Z}_d de l'objectif de (D) lorsqu'on prend $\tilde{\lambda}$ comme solution initiale ; il ne peut donc que dépasser ou égaler \tilde{Z}_d à la fin de l'heuristique de montée duale.

Les heuristiques de montée duale nous permettront donc de trouver une nouvelle solution réalisable du problème (D) ayant un objectif de valeur supérieure ou égale à la valeur de l'objectif de la solution courante $\tilde{\lambda}$.

Méthodes exactes

Si on résout le sous-problème $(D(\tilde{L}, \tilde{\lambda}))$ à l'aide d'une méthode exacte plutôt qu'avec une heuristique de montée duale, alors on obtient la valeur optimale $Z_d(\tilde{L}, \tilde{\lambda})$, qui ne peut donc qu'égaliser ou dépasser la valeur courante \tilde{Z}_d , correspondant à la solution $\tilde{\lambda}$, réalisable pour $(D(\tilde{L}, \tilde{\lambda}))$. En effet, nous avons :

$$\begin{aligned} Z_d(\tilde{L}, \tilde{\lambda}) &= \left\{ \sum_{l \in \tilde{L}} \tilde{\lambda}^l b^l + \max \sum_{l \in L \setminus \tilde{L}} \lambda^l b^l \mid \sum_{l \in L \setminus \tilde{L}} \lambda^l A^l \leq c - \sum_{l \in \tilde{L}} \tilde{\lambda}^l A^l; \quad \lambda^l \geq 0, \forall l \in L \setminus \tilde{L} \right\} \\ &\geq \sum_{l \in \tilde{L}} \tilde{\lambda}^l b^l + \sum_{l \in L \setminus \tilde{L}} \tilde{\lambda}^l b^l \\ &= \sum_{l \in L} \tilde{\lambda}^l b^l. \end{aligned}$$

Les méthodes exactes nous permettront donc, comme les heuristiques de montée duale, de trouver une nouvelle solution réalisable du problème (D) qui améliorera la valeur courante de l'objectif du dual ou la laissera inchangée si la solution courante $\tilde{\lambda}$ était déjà une solution optimale du sous-problème $(D(\tilde{L}, \tilde{\lambda}))$.

Algorithme de décomposition par montée duale

La méthode de décomposition par montée duale consiste donc à résoudre, à l'aide de méthodes exactes ou d'heuristiques de montée duale, k sous-problèmes obtenus du problème original en fixant certaines variables duales à leurs valeurs respectives dans la solution du sous-problème précédent. Peu importe l'heuristique de montée duale ou la méthode exacte utilisée pour résoudre les sous-problèmes, l'objectif du dual ne peut qu'augmenter ou rester constant. L'algorithme se termine lorsqu'un critère d'arrêt prédéterminé est atteint.

Notons \tilde{L}_j l'ensemble des indices des variables duales fixées lors de la résolution du sous-problème j , $1 \leq j \leq NbSProb$. Alors, l'algorithme de décomposition par montée duale peut être décrit ainsi :

Déterminer $\tilde{\lambda}$, une solution réalisable de (D) ;
Tant que (le critère d'arrêt n'est pas atteint)
 Incréments le nombre d'itérations ;
Pour $j := 1$ à $NbSProb$
 Résoudre le sous-problème j , $(D(\tilde{L}_j, \tilde{\lambda}))$, pour obtenir une solution $\bar{\lambda}$;
 $\tilde{\lambda} := \bar{\lambda}$.

2.2 Montée duale pure

Notre algorithme de montée duale pure utilise le cadre général de la décomposition par montée duale décrit dans la sous-section précédente avec $NbSProb = 3$. Les trois sous-problèmes résolus à tour de rôle sont : un sous-problème de *conception de réseaux sans capacité*, un sous-problème de *sac à dos continu* et un sous-problème de *multiflot* ; nous allons présenter en détail ces trois sous-problèmes dans les sections qui suivent.

L'algorithme s'arrête lorsqu'un des trois critères d'arrêt suivants est atteint :

- aucune variable duale n'a été modifiée lors de la dernière itération ;
- un nombre maximum prédéterminé d'itérations a été atteint ;

- l'objectif du dual ne s'est pas suffisamment amélioré depuis la dernière itération (ce qui sera mesuré par un paramètre).

L'ordre des sous-problèmes, ainsi que le choix des paramètres à utiliser dans les critères d'arrêt seront testés et analysés au chapitre 3.

2.2.1 Initialisation

Nous devons, avant de commencer la résolution du problème, donner des valeurs initiales réalisables aux variables duales. Nous définissons aussi deux autres variables \bar{c}_{ij}^k et \bar{f}_{ij} , représentant respectivement les variables d'écart des contraintes duales (1.13) et (1.14) (c'est-à-dire les coûts réduits dans le contexte primal) :

$$\begin{aligned}\bar{c}_{ij}^k &= c_{ij}^k + \pi_i^k - \pi_j^k + \beta_{ij}^k + \alpha_{ij} + \gamma_{ij} \geq 0, \\ \bar{f}_{ij} &= f_{ij} - u_{ij}\alpha_{ij} - \sum_{k \in K} b_{ij}^k \beta_{ij}^k + \lambda_{ij} \geq 0.\end{aligned}$$

Nous proposons la stratégie d'initialisation suivante :

$$\begin{aligned}\alpha_{ij} &:= \tau \frac{f_{ij}}{u_{ij}}, \quad \forall (i, j) \in A, \\ \lambda_{ij}, \gamma_{ij} &:= 0, \quad \forall (i, j) \in A, \\ \pi_i^k &:= 0, \quad \forall i \in N, k \in K, \\ \beta_{ij}^k &:= 0, \quad \forall (i, j) \in A, k \in K, \\ \bar{c}_{ij}^k &:= c_{ij}^k + \alpha_{ij}, \quad \forall (i, j) \in A, k \in K, \\ \bar{f}_{ij} &:= f_{ij} - u_{ij}\alpha_{ij}, \quad \forall (i, j) \in A, \\ \bar{Z}_d &:= 0,\end{aligned}$$

où \bar{Z}_d est la valeur courante de l'objectif du dual et $0 \leq \tau \leq 1$ est un paramètre qui sera testé et analysé au chapitre 3.

Si $\tau = 0$, alors toutes les variables duales sont nulles, ce qui ne semble pas très prometteur. Si $\tau = 1$, alors le modèle obtenu en fixant à leurs valeurs initiales uniquement $\alpha_{ij}, \forall (i, j) \in A$, et $\beta_{ij}^k, \forall (i, j) \in A, k \in K$, correspond au dual de la relaxation linéaire de la formulation faible. En effet, cette relaxation est obtenue en supprimant

les contraintes (redundantes) de capacité individuelle (1.9) du modèle général. Sans ces contraintes, les variables de conception y_{ij} ne sont présentes que dans les contraintes de capacité mutuelle (1.8) et peuvent par conséquent être fixées à $y_{ij} = \frac{\sum_{k \in K} x_{ij}^k}{u_{ij}}$ pour chaque arc (i, j) . Si on remplace y_{ij} par cette valeur dans l'objectif, nous obtenons des coûts de transport modifiés $\hat{c}_{ij}^k = c_{ij}^k + \frac{f_{ij}}{u_{ij}}$ qui sont égaux à \bar{c}_{ij}^k lorsque $\tau = 1$. Par conséquent, on s'attend à ce qu'un paramètre τ plus près de 1 donne de meilleurs résultats qu'un paramètre τ plus près de 0.

Nous avons aussi envisagé l'initialisation des variables π_i^k aux plus courts chemins du noeud origine de chaque produit k au noeud i , mais comme nous pouvons obtenir le même résultat en résolvant le sous-problème de conception de réseaux sans capacité immédiatement après l'initialisation, nous avons jugé inutile de fixer initialement la valeur de ces variables.

2.2.2 Sous-problème de conception de réseaux sans capacité

Dans le premier type de sous-problème, nous fixons les variables duales α_{ij} , λ_{ij} et γ_{ij} . Nous tentons donc de modifier les autres variables duales (π_i^k et β_{ij}^k) de façon à augmenter l'objectif du dual. Le dual correspondant $D(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma})$ est alors :

$$Z_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma}) = \max \sum_{k \in K} d^k \pi_{D(k)}^k - \sum_{(i,j) \in A} u_{ij} \tilde{\gamma}_{ij} - \sum_{(i,j) \in A} \tilde{\lambda}_{ij}, \quad (2.1)$$

Sujet à :

$$\pi_j^k - \pi_i^k - \beta_{ij}^k \leq c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\gamma}_{ij}, \quad \forall (i, j) \in A, k \in K, \quad (2.2)$$

$$\sum_{k \in K} b_{ij}^k \beta_{ij}^k \leq f_{ij} - u_{ij} \tilde{\alpha}_{ij} + \tilde{\lambda}_{ij}, \quad \forall (i, j) \in A, \quad (2.3)$$

$$\beta_{ij}^k \geq 0, \quad \forall (i, j) \in A, k \in K.$$

Le primal de ce sous-problème, $P(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma})$, est en fait la relaxation continue du

problème de conception de réseaux sans capacité :

$$Z_p(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma}) = \min \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\gamma}_{ij}) x_{ij}^k + \sum_{(i,j) \in A} (f_{ij} - u_{ij} \tilde{\alpha}_{ij} + \tilde{\lambda}_{ij}) y_{ij} - \sum_{(i,j) \in A} u_{ij} \tilde{\gamma}_{ij} - \sum_{(i,j) \in A} \tilde{\lambda}_{ij},$$

Sujet à :

$$\sum_{j \in N | (j,i) \in A} x_{ji}^k - \sum_{j \in N | (i,j) \in A} x_{ij}^k = \begin{cases} -d^k, & i = O(k), \\ d^k, & i = D(k), \\ 0, & i \neq O(k), D(k), \end{cases} \quad \begin{matrix} \forall i \in N, \\ k \in K, \end{matrix}$$

$$x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall (i,j) \in A, k \in K,$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, k \in K,$$

$$y_{ij} \geq 0, \quad \forall (i,j) \in A.$$

Nous résolvons ce sous-problème par une heuristique de montée duale développée par Balakrishnan, Magnanti et Wong [1]. Notons que dans la formulation présentée par ces auteurs, $b_{ij}^k = d^k, \forall (i,j) \in A, k \in K$. Pour tenir compte de ce changement dans le modèle, nous avons légèrement modifié la méthode qui consiste en un algorithme de marquage pouvant être vu comme une généralisation de l'algorithme de plus court chemin de Dijkstra.

Stratégie générale

La stratégie utilisée consiste à modifier, à chaque itération, les variables β_{ij}^k et π_i^k afin d'augmenter l'objectif du dual de façon monotone. Notons que, pour tout vecteur $\bar{\beta}$ qui satisfait les contraintes (2.3), les meilleures valeurs de π_i^k sont obtenues en résolvant le problème défini par (2.1)–(2.2) avec $\beta = \bar{\beta}$. Ce problème, décomposé par produit k , engendre le problème $D^k(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma}, \bar{\beta})$:

$$Z_d^k(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma}, \bar{\beta}) = \max d^k \pi_{D(k)}^k - \sum_{(i,j) \in A} u_{ij} \tilde{\gamma}_{ij} - \sum_{(i,j) \in A} \tilde{\lambda}_{ij},$$

Sujet à :

$$\pi_j^k - \pi_i^k \leq c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\gamma}_{ij} + \bar{\beta}_{ij}^k, \quad \forall (i,j) \in A.$$

Or, $D^k(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma}, \bar{\beta})$ n'est autre que le dual d'un problème de plus court chemin de $O(k)$ à $D(k)$ avec des coûts modifiés : $\hat{c}_{ij}^k = c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\gamma}_{ij} + \bar{\beta}_{ij}^k$. Une solution optimale de $D^k(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma}, \bar{\beta})$ peut donc être trouvée en donnant aux variables π_i^k la valeur du plus court chemin de l'origine $O(k)$ au noeud i , en utilisant les coûts modifiés \hat{c}_{ij}^k comme longueur des arcs.

Notons qu'il est possible d'augmenter la valeur de l'objectif de $D(\tilde{\alpha}, \tilde{\lambda}, \tilde{\gamma})$ en augmentant la longueur du plus court chemin entre $O(k)$ et $D(k)$, pour un ou plusieurs produits k à la fois, grâce à des augmentations des β_{ij}^k et, par le fait même, des \hat{c}_{ij}^k . À chaque itération, la méthode tente donc d'augmenter les β_{ij}^k de façon à ce que les contraintes (2.3) ne soient pas violées et que la longueur du plus court chemin de $O(k)$ à $D(k)$, $\pi_{D(k)}^k$, augmente pour au moins un produit k . Pour que les contraintes (2.3) ne soient pas violées par l'augmentation des β_{ij}^k , on n'augmente que les β_{ij}^k qui correspondent à des contraintes (2.3) qui ont un écart positif, c'est-à-dire qui ne sont pas saturées (ou de façon équivalente, lorsque $\bar{f}_{ij} > 0$).

Méthode d'étiquetage

Définissons d'abord, pour chaque produit k , $N_u(k) \subseteq N$ qui est l'ensemble des noeuds non étiquetés pour ce produit ; $N - N_u(k)$ représente donc l'ensemble des noeuds étiquetés pour le produit k . Au départ, tous les noeuds sont non étiquetés, sauf le noeud destination de chaque produit. De plus, nous définissons l'ensemble $A(N_u(k)) = \{(i, j) \in A \mid i \in N_u(k), j \in N - N_u(k)\}$.

À chaque itération, la méthode augmente plusieurs variables β_{ij}^k correspondant au même produit k . Ayant choisi un produit k , il s'agit donc ensuite d'identifier les arcs (i, j) pour lesquels on peut augmenter la valeur de β_{ij}^k afin de permettre d'augmenter le plus court chemin de $O(k)$ à $D(k)$, $\pi_{D(k)}^k$. Clairement, en augmentant les β_{ij}^k pour tous les arcs $(i, j) \in A(N_u(k))$, les plus courts chemins de $O(k)$ à tous les noeuds de $N - N_u(k)$, incluant le noeud destination $D(k)$, augmenteront aussi. Par contre, certains de ces β_{ij}^k n'ont pas besoin d'être augmentés pour augmenter $\pi_{D(k)}^k$. En effet, si un arc $(i, j) \in A(N_u(k))$ n'appartient à aucun plus court chemin de $O(k)$ à $D(k)$, alors on peut

ne pas modifier le β_{ij}^k correspondant. Pour ne pas augmenter les β_{ij}^k correspondant à de tels arcs, notons qu'un arc (i, j) utilisé dans un plus court chemin de $O(k)$ à $D(k)$ satisfait, à cause des écarts complémentaires (puisque son flot sera non nul), l'équation suivante :

$$\pi_j^k - \pi_i^k - \beta_{ij}^k = c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\gamma}_{ij}. \quad (2.4)$$

Notons $A'(N_u(k))$ l'ensemble des arcs $(i, j) \in A(N_u(k))$ qui satisfont cette équation et dont, par conséquent, les β_{ij}^k doivent être augmentés pour faire augmenter un ou plusieurs $\pi_{D(k)}^k$. Pour chaque arc $(i, j) \in A'(N_u(k))$, l'écart courant sur les contraintes (2.3), \bar{f}_{ij} , déterminera la quantité maximale par laquelle on peut augmenter les β_{ij}^k tout en maintenant la réalisabilité du dual :

$$\delta_1 = \min\{\bar{f}_{ij}/b_{ij}^k : (i, j) \in A'(N_u(k))\}.$$

À mesure que la méthode augmente les β_{ij}^k pour les arcs $(i, j) \in A'(N_u(k))$, les plus courts chemins de l'origine à chaque noeud de $N - N_u(k)$ augmentent aussi. Un ou plusieurs arcs qui n'appartenaient à aucun plus court chemin pour le produit k peuvent maintenant appartenir à des plus courts chemins. Pour de tels arcs, l'équation (2.4) doit être satisfaite. Les prochains arcs de $A(N_u(k)) - A'(N_u(k))$ à "entrer" dans un plus court chemin restreindront donc aussi l'augmentation des β_{ij}^k :

$$\delta_2 = \min\{\bar{c}_{ij}^k : (i, j) \in A(N_u(k)) - A'(N_u(k)), \bar{c}_{ij}^k > 0\}.$$

Augmenter les β_{ij}^k de $\delta = \min\{\delta_1, \delta_2\}$ pour tous les arcs $(i, j) \in A'(N_u(k))$ cause une augmentation des plus courts chemins π_i^k , pour tous les noeuds $i \in N - N_u(k)$, de δ . Cette procédure "montée $\pi - \beta$ " a donc pour effet d'augmenter l'objectif du dual. Notons que si $\delta = \delta_1$, l'écart \bar{f}_{ij} d'au moins un arc $(i, j) \in A'(N_u(k))$ s'annule, tandis que si $\delta = \delta_2$, un arc de $A(N_u(k)) - A'(N_u(k))$ "entre" dans un plus court chemin.

À chaque itération, cet algorithme s'attaque aux produits k pour lesquels $O(k) \notin N - N_u(k)$, un à la fois. Pour chacun de ces produits, il exécute la procédure "montée $\pi - \beta$ " une fois et si \bar{f}_{ij} s'annule pour un certain arc (i, j) , alors le noeud i devient étiqueté, passant de l'ensemble $N_u(k)$ à l'ensemble $N - N_u(k)$; c'est pourquoi on parle d'un

algorithme d'étiquetage ressemblant à l'algorithme de plus court chemin de Dijkstra. L'algorithme s'arrête quand l'origine $O(k)$ est marquée pour tous les produits $k \in K$.

Algorithme "Conception de réseaux sans capacité"

Initialisation

$N_u(k) \leftarrow N - \{D(k)\}, \forall k \in K$: ensemble des noeuds non étiquetés ;

$K_u \leftarrow K$: ensemble des produits candidats ;

$A(N_u(k)) \leftarrow \{(i, j) \in A \mid i \in N_u(k), j \in N - N_u(k)\}, \quad \forall k \in K.$

Étape 1

Choisir $k \in K_u$.

Étape 2 : mises à jour (montée $\pi - \beta$)

$\delta_1 \leftarrow \min\{\bar{f}_{ij}/b_{ij}^k \mid \bar{c}_{ij}^k = 0, \forall (i, j) \in A(N_u(k))\}$;

$\delta_2 \leftarrow \min\{\bar{c}_{ij}^k \mid \bar{c}_{ij}^k > 0, \forall (i, j) \in A(N_u(k))\}$;

$\delta \leftarrow \min\{\delta_1, \delta_2\}$;

$\beta_{ij}^k \leftarrow \beta_{ij}^k + \delta, \quad \forall (i, j) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k = 0\}$;

$\bar{f}_{ij} \leftarrow \bar{f}_{ij} - b_{ij}^k \delta, \quad \forall (i, j) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k = 0\}$;

$\bar{c}_{ij}^k \leftarrow \bar{c}_{ij}^k - \delta, \quad \forall (i, j) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k > 0\}$;

$\bar{c}_{ij}^k \leftarrow \bar{c}_{ij}^k + \delta, \quad \forall (i, j) \in \{(i, j) \in A \mid i \in N - N_u(k), j \in N_u(k)\}$;

$\pi_i^k \leftarrow \pi_i^k + \delta, \quad \forall i \in N - N_u(k)$;

$\bar{Z}_d \leftarrow \bar{Z}_d + d^k \delta.$

Étape 3 : étiquetage

Si $\delta = \delta_1$, choisir un arc $(i^*, j^*) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k = 0\}$

tel que $\bar{f}_{i^*j^*} = 0$ et faire $N_u(k) \leftarrow N_u(k) - \{i^*\}$.

Étape 4

$K_u \leftarrow K_u - \{k\}$;

si $K_u \neq \emptyset$, aller à l'étape 1.

Étape 5 : critère d'arrêt

$K_u \leftarrow \{k \mid O(k) \in N_u(k)\}$;

si $K_u = \emptyset$, STOP ;

sinon, aller à l'étape 1.

Remarques :

- À l'étape 1, nous avons choisi de traiter, à chaque itération, les produits k qui ont la même origine un après l'autre, tel que suggéré par Balakrishnan, Magnanti et Wong [1]. En effet, les résultats préliminaires de Balakrishnan, Magnanti et Wong [1] ont permis aux auteurs de conclure que des stratégies plus sophistiquées pour ordonner les produits $k \in K$ ne donnent pas systématiquement de meilleurs résultats.
- À chaque itération (étapes 1 à 4), l'algorithme traite tous les produits $k \in K$ dont l'origine n'est pas étiquetée, faisant potentiellement passer des noeuds de l'ensemble des noeuds non étiquetés à l'ensemble des noeuds étiquetés. Puis, ces itérations sont répétées tant que le noeud origine d'au moins un produit demeure non marqué (étape 5).

2.2.3 Sous-problème de sac à dos continu

Dans le deuxième type de sous-problème, nous fixons les variables π_i^k et γ_{ij} . Nous tentons donc de modifier les autres variables duales (α_{ij} , λ_{ij} et β_{ij}^k) de façon à augmenter l'objectif du dual. Le dual correspondant $D(\tilde{\pi}, \tilde{\gamma})$ est alors :

$$Z_d(\tilde{\pi}, \tilde{\gamma}) = \max - \sum_{(i,j) \in A} \lambda_{ij} - \sum_{(i,j) \in A} u_{ij} \tilde{\gamma}_{ij} + \sum_{k \in K} d^k \tilde{\pi}_{D(k)}^k,$$

Sujet à :

$$\begin{aligned} -\alpha_{ij} - \beta_{ij}^k &\leq c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}, & \forall (i, j) \in A, k \in K, \\ -\lambda_{ij} + u_{ij} \alpha_{ij} + \sum_{k \in K} b_{ij}^k \beta_{ij}^k &\leq f_{ij}, & \forall (i, j) \in A, \\ \alpha_{ij}, \lambda_{ij} &\geq 0, & \forall (i, j) \in A, \\ \beta_{ij}^k &\geq 0, & \forall (i, j) \in A, k \in K. \end{aligned}$$

Puisque les variables π_i^k sont fixées, on peut maintenant décomposer ce sous-problème $D(\tilde{\pi}, \tilde{\gamma})$ par arc, en plus d'enlever le terme constant de l'objectif, pour obtenir des sous-

problèmes, $D^{ij}(\tilde{\pi}, \tilde{\gamma})$, de la forme suivante :

$$Z_d^{ij}(\tilde{\pi}, \tilde{\gamma}) = \max \quad - \lambda_{ij},$$

Sujet à :

$$\begin{aligned} -\alpha_{ij} - \beta_{ij}^k &\leq c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}, & \forall k \in K, \\ -\lambda_{ij} + u_{ij}\alpha_{ij} + \sum_{k \in K} b_{ij}^k \beta_{ij}^k &\leq f_{ij}, \\ \alpha_{ij}, \lambda_{ij} &\geq 0, \\ \beta_{ij}^k &\geq 0, & \forall k \in K. \end{aligned}$$

Le primal de ce sous-problème, $P^{ij}(\tilde{\pi}, \tilde{\gamma})$, est donné par :

$$Z_p^{ij}(\tilde{\pi}, \tilde{\gamma}) = \min \sum_{k \in K} (c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}) x_{ij}^k + f_{ij} y_{ij}, \quad (2.5)$$

Sujet à :

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad (2.6)$$

$$x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall k \in K, \quad (2.7)$$

$$y_{ij} \leq 1, \quad (2.8)$$

$$x_{ij}^k \geq 0, \quad \forall k \in K, \quad (2.9)$$

$$y_{ij} \geq 0. \quad (2.10)$$

Nous allons trouver une solution primale optimale (\bar{x}, \bar{y}) du sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ à l'aide d'une méthode exacte, puis déduire une solution duale optimale du sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ à l'aide de (\bar{x}, \bar{y}) et de la théorie des écarts complémentaires.

Solution primale

Montrons d'abord qu'il existe toujours une solution optimale du sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ telle que $y_{ij} \in \{0, 1\}$.

Théorème : Toute solution optimale du sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ où y_{ij} est fractionnaire ne peut être meilleure que la meilleure solution dans laquelle $y_{ij} \in \{0, 1\}$.

Avant de procéder à la preuve du théorème, nous allons d'abord analyser les sous-problèmes obtenus lorsque $y_{ij} \in \{0, 1\}$. Cette analyse sera ensuite utilisée dans la preuve du théorème.

Si $y_{ij} = 0$, alors les contraintes (2.6) et (2.7) imposent $x_{ij}^k = 0, \forall k \in K$, et cette solution a un objectif de valeur $Z_p^{ij}(\tilde{\pi}, \tilde{\gamma}) = 0$. Notons que le seul cas où il sera avantageux d'avoir une solution (x, y) non nulle sera lorsque certains coûts réduits $c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}$ sont négatifs et qu'il existe une solution (x, y) telle que $\sum_{k \in K} (c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij})x_{ij}^k + f_{ij}y_{ij} < 0$.

Si $y_{ij} = 1$, alors le sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ devient un problème de *sac à dos avec variables continues*, noté $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$, pour lequel les coûts des objets, les poids des objets, les capacités des objets et la capacité du sac sont, respectivement, $c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}$, 1, b_{ij}^k et u_{ij} :

$$Z_{px}^{ij}(\tilde{\pi}, \tilde{\gamma}) = \min \sum_{k \in K} (c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij})x_{ij}^k + f_{ij},$$

Sujet à :

$$\begin{aligned} \sum_{k \in K} x_{ij}^k &\leq u_{ij}, \\ 0 \leq x_{ij}^k &\leq b_{ij}^k, \quad \forall k \in K. \end{aligned}$$

Soit K^- l'ensemble des produits $k \in K$ à coût strictement négatif. Si $\sum_{k \in K^-} b_{ij}^k < u_{ij}$, alors une solution optimale du sous-problème $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$ consiste à inclure dans le sac tous les objets $k \in K^-$ jusqu'à leur borne supérieure b_{ij}^k et à exclure tous les autres objets. Si $\sum_{k \in K^-} b_{ij}^k \geq u_{ij}$, alors nous définissons un tableau T dans lequel les objets $k \in K^-$ sont triés en ordre croissant de coûts. Une solution optimale du sous-problème $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$ consiste dans ce cas à inclure dans le sac les premiers objets de T à pleine capacité, et ce tant que la capacité totale du sac n'est pas dépassée. Le prochain objet dans T , appelé objet critique et noté k^* , est partiellement inclus dans le sac afin de prendre la capacité résiduelle de celui-ci ; tous les autres objets sont exclus. La procédure "Sac à dos auxiliaire primal" trouve une solution optimale au sous-problème $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$:

Procédure “Sac à dos auxiliaire primal”

Si $\sum_{k \in K^-} b_{ij}^k < u_{ij}$ **alors**

Pour chaque $k \in K^-$ **faire**

$$\hat{x}_{ij}^k := b_{ij}^k;$$

Pour chaque $k \in K - K^-$ **faire**

$$\hat{x}_{ij}^k := 0;$$

Sinon

Pour chaque $k < k^*$ **dans le tableau** T **faire**

$$\hat{x}_{ij}^k := b_{ij}^k;$$

Pour chaque $k > k^*$ **dans le tableau** T **et pour chaque** $k \in K - K^-$ **faire**

$$\hat{x}_{ij}^k := 0;$$

Pour l'élément critique k^* **faire**

$$\hat{x}_{ij}^{k^*} := u_{ij} - \sum_{k \neq k^*} \hat{x}_{ij}^k.$$

Preuve du théorème : Soit $\hat{y}_{ij} = 1$ et $\hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma}) = \sum_{k \in K} (c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}) \hat{x}_{ij}^k + f_{ij}$ la valeur de l'objectif de la solution optimale \hat{x} du problème $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$, trouvée à l'aide de la procédure “Sac à dos auxiliaire primal”.

Soit $0 < \tilde{y}_{ij} < 1$. Le sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ correspondant à \tilde{y}_{ij} devient un problème de sac à dos avec variables continues pour lequel les coûts des objets, les poids des objets, les capacités des objets et la capacité du sac sont, respectivement, $c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}$, 1 , $\tilde{b}_{ij}^k = b_{ij}^k \tilde{y}_{ij}$ et $\tilde{u}_{ij} = u_{ij} \tilde{y}_{ij}$:

$$\min \sum_{k \in K} (c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}) x_{ij}^k + f_{ij} \tilde{y}_{ij},$$

Sujet à :

$$\begin{aligned} \sum_{k \in K} x_{ij}^k &\leq \tilde{u}_{ij}, \\ 0 \leq x_{ij}^k &\leq \tilde{b}_{ij}^k, \quad \forall k \in K. \end{aligned}$$

Comme ce problème a la même structure que le problème $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$, on peut y appliquer la procédure “Sac à dos auxiliaire primal” afin d'en obtenir la solution optimale

$\tilde{x}_{ij}^k = \hat{x}_{ij}^k \tilde{y}_{ij}, \forall k \in K$. En effet, l'ensemble K^- , le tableau T et l'élément critique k^* sont les mêmes que pour $\hat{y}_{ij} = 1$; seules les capacités \tilde{u}_{ij} et \tilde{b}_{ij}^k sont modifiées en étant toutes multipliées par le même facteur. Par conséquent, la quantité prise pour chaque objet change, mais sa proportion dans le poids total du sac à dos demeure la même. L'objectif de la solution (\tilde{x}, \tilde{y}) est donc :

$$\begin{aligned} \tilde{Z}_p^{ij}(\tilde{\pi}, \tilde{\gamma}) &= \sum_{k \in K} (c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}) \tilde{x}_{ij}^k + f_{ij} \tilde{y}_{ij} \\ &= \sum_{k \in K} (c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}) \hat{x}_{ij}^k \tilde{y}_{ij} + f_{ij} \tilde{y}_{ij} \\ &= \tilde{y}_{ij} \left(\sum_{k \in K} (c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}) \hat{x}_{ij}^k + f_{ij} \right) \\ &= \tilde{y}_{ij} \hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma}). \end{aligned}$$

Si $\hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma})$ est positif, alors $\tilde{Z}_p^{ij}(\tilde{\pi}, \tilde{\gamma})$ est aussi positif puisque $\tilde{y}_{ij} > 0$. La solution optimale du sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ est donc $(y_{ij} = 0; x_{ij}^k = 0, \forall k \in K)$, puisqu'on cherche à minimiser l'objectif, et sa valeur est 0.

Si $\hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma})$ est négatif, alors

$$\begin{aligned} \tilde{Z}_p^{ij}(\tilde{\pi}, \tilde{\gamma}) &= \tilde{y}_{ij} \hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma}) \\ \Rightarrow \tilde{Z}_p^{ij}(\tilde{\pi}, \tilde{\gamma}) &> \hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma}), \end{aligned}$$

et la solution optimale du sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$, de valeur $\hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma})$, est $(y_{ij} = 1; x_{ij}^k = \hat{x}_{ij}^k, \forall k \in K)$ puisqu'on cherche à minimiser l'objectif.

Si $\hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma}) = 0$, alors $\tilde{Z}_p^{ij}(\tilde{\pi}, \tilde{\gamma}) = 0$ aussi, tout comme la valeur de la solution $(y_{ij} = 0; x_{ij}^k = 0, \forall k \in K)$. Toutes ces solutions sont par conséquent optimales.

Il existe donc toujours une solution optimale de $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ telle que $y_{ij} = 0$ ou $y_{ij} = 1$.

□

En conclusion, pour trouver une solution primale optimale au sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$, il suffit d'exécuter la procédure "Sac à dos primal" :

Procédure “Sac à dos primal”

$(\hat{x}_{ij}^k, \hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma})) \leftarrow$ Sac à dos auxiliaire primal (K^-, T, k^*)

Si $\hat{Z}_{px}^{ij}(\tilde{\pi}, \tilde{\gamma}) < 0$ **alors**

$$\bar{y}_{ij} := 1;$$

$$\bar{x}_{ij}^k := \hat{x}_{ij}^k, \quad \forall k \in K;$$

Sinon

$$\bar{y}_{ij} := 0;$$

$$\bar{x}_{ij}^k := 0, \quad \forall k \in K.$$

Solution duale

Ayant trouvé une solution primale optimale (\bar{x}, \bar{y}) du sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ avec la procédure “Sac à dos primal”, nous allons déduire une solution duale optimale $(\bar{\alpha}, \bar{\lambda}, \bar{\beta})$ du sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ à l’aide de la théorie des écarts complémentaires. Les conditions des écarts complémentaires que nous devons satisfaire sont les suivantes :

$$(u_{ij}\bar{y}_{ij} - \sum_{k \in K} \bar{x}_{ij}^k)\bar{\alpha}_{ij} = 0, \quad (2.11)$$

$$(b_{ij}^k\bar{y}_{ij} - \bar{x}_{ij}^k)\bar{\beta}_{ij}^k = 0, \quad \forall k \in K, \quad (2.12)$$

$$(1 - \bar{y}_{ij})\bar{\lambda}_{ij} = 0, \quad (2.13)$$

$$(c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij} + \bar{\alpha}_{ij} + \bar{\beta}_{ij}^k)\bar{x}_{ij}^k = 0, \quad \forall k \in K, \quad (2.14)$$

$$(f_{ij} - u_{ij}\bar{\alpha}_{ij} - \sum_{k \in K} b_{ij}^k\bar{\beta}_{ij}^k + \bar{\lambda}_{ij})\bar{y}_{ij} = 0, \quad (2.15)$$

$$(\bar{x}, \bar{y}) \text{ solution réalisable de } P^{ij}(\tilde{\pi}, \tilde{\gamma}), \quad (2.16)$$

$$(\bar{\alpha}, \bar{\lambda}, \bar{\beta}) \text{ solution réalisable de } D^{ij}(\tilde{\pi}, \tilde{\gamma}). \quad (2.17)$$

Solution duale optimale lorsque $\bar{y}_{ij} = 1$

Lorsque $\bar{y}_{ij} = 1$ dans la solution primale optimale du sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$, il faut que $\bar{f}_{ij} = 0$ pour satisfaire la condition (2.15). Nous remarquons aussi que la condition (2.13) est toujours satisfaite puisque $(1 - \bar{y}_{ij}) = 0$.

Si $\sum_{k \in K} \bar{x}_{ij}^k < u_{ij}$, alors

- on doit poser $\bar{\alpha}_{ij} = 0$ dans la condition (2.11) puisque $\sum_{k \in K} \bar{x}_{ij}^k \neq u_{ij} \bar{y}_{ij}$;
- pour chaque produit $k \in K$, si $\bar{x}_{ij}^k = 0$, alors la condition (2.14) est satisfaite et on doit poser $\bar{\beta}_{ij}^k = 0$ pour satisfaire la condition (2.12) ; si, au contraire, $\bar{x}_{ij}^k = b_{ij}^k$, alors la condition (2.12) est satisfaite et on doit poser $\bar{\beta}_{ij}^k = \tilde{\pi}_j^k - \tilde{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} > 0$ pour satisfaire la condition (2.14) ;
- si $f_{ij} - \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k \leq 0$, alors on doit poser $\bar{\lambda}_{ij} = \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k - f_{ij} \geq 0$ pour satisfaire la condition (2.15) ; $f_{ij} - \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k$ ne peut être positif puisque le problème est réalisable et qu'on doit pouvoir imposer $\bar{f}_{ij} = 0$ pour trouver une solution optimale.

Si $\sum_{k \in K} \bar{x}_{ij}^k = u_{ij}$, alors

- les conditions (2.11) et (2.13) sont satisfaites puisque $\bar{y}_{ij} = 1$ et $\sum_{k \in K} \bar{x}_{ij}^k = u_{ij}$; les variables $\bar{\alpha}_{ij}$ et $\bar{\lambda}_{ij}$ sont donc libres ;
- on pose $\bar{\alpha}_{ij} = \tilde{\pi}_j^{k^*} - \tilde{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij} > 0$ puisque cette variable est libre ;
- pour chaque produit $k \in K$, si $\bar{x}_{ij}^k = 0$, alors la condition (2.14) est satisfaite et on doit poser $\bar{\beta}_{ij}^k = 0$ pour satisfaire la condition (2.12) ; si $\bar{x}_{ij}^k = b_{ij}^k$, alors la condition (2.12) est satisfaite et on doit poser $\bar{\beta}_{ij}^k = \tilde{\pi}_j^k - \tilde{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} - \bar{\alpha}_{ij} \geq 0$ pour satisfaire la condition (2.14) ; pour l'élément critique k^* , on pose $\bar{\beta}_{ij}^{k^*} = 0$ pour satisfaire la condition (2.12), et alors la condition (2.14) est nécessairement satisfaite puisque

$$\begin{aligned}
 & (c_{ij}^{k^*} + \tilde{\pi}_i^{k^*} - \tilde{\pi}_j^{k^*} + \tilde{\gamma}_{ij} + \bar{\alpha}_{ij} + \bar{\beta}_{ij}^{k^*}) \bar{x}_{ij}^{k^*} \\
 &= (c_{ij}^{k^*} + \tilde{\pi}_i^{k^*} - \tilde{\pi}_j^{k^*} + \tilde{\gamma}_{ij} + \bar{\alpha}_{ij}) \bar{x}_{ij}^{k^*} \\
 &= (c_{ij}^{k^*} + \tilde{\pi}_i^{k^*} - \tilde{\pi}_j^{k^*} + \tilde{\gamma}_{ij} + \tilde{\pi}_j^{k^*} - \tilde{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij}) \bar{x}_{ij}^{k^*} \\
 &= 0;
 \end{aligned}$$

- si $f_{ij} - u_{ij} \bar{\alpha}_{ij} - \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k \leq 0$, alors on doit poser $\bar{\lambda}_{ij} = u_{ij} \bar{\alpha}_{ij} + \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k - f_{ij}$ pour satisfaire la condition (2.15) ; $f_{ij} - u_{ij} \bar{\alpha}_{ij} - \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k$ ne peut être positif puisque le problème est réalisable et qu'on doit pouvoir imposer $\bar{f}_{ij} = 0$ pour trouver une solution optimale.

Toutes les variables duales $\bar{\alpha}_{ij}$, $\bar{\lambda}_{ij}$ et $\bar{\beta}_{ij}^k$ sont non négatives et forment une solution réalisable. Par conséquent, toutes les conditions des écarts complémentaires sont

satisfaites et la solution duale $(\bar{\alpha}_{ij}, \bar{\lambda}_{ij}, \bar{\beta}_{ij}^k)$ constitue une solution optimale du sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ lorsque $\bar{y}_{ij} = 1$.

Solution duale optimale lorsque $\bar{y}_{ij} = 0$

Si $\bar{y}_{ij} = 0$ dans la solution primale optimale, alors $\bar{x}_{ij}^k = 0, \forall k \in K$. Par conséquent, les conditions (2.11), (2.12), (2.14), (2.15) et (2.16) des écarts complémentaires sont satisfaites. Pour satisfaire la condition (2.13), on doit poser $\bar{\lambda}_{ij} = 0$, fixant ainsi l'objectif du dual $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ à 0. Toutes les conditions des écarts complémentaires sont satisfaites, sauf la condition (2.17) ; il suffit donc de trouver $\bar{\alpha}$ et $\bar{\beta}$ tels que $(\bar{\alpha}, \bar{\lambda}, \bar{\beta})$ est une solution réalisable du sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ pour avoir une solution duale optimale de ce sous-problème. La résolution du problème auxiliaire suivant, $D_{\alpha\beta}^{ij}(\tilde{\pi}, \tilde{\gamma})$, nous permettra d'obtenir une telle solution réalisable, tout en tentant de maximiser la variable d'écart $\bar{f}_{ij} = f_{ij} - (u_{ij}\alpha_{ij} + \sum_{k \in K} b_{ij}^k \beta_{ij}^k) + \bar{\lambda}_{ij} \geq 0$:

$$Z_{d\alpha\beta}^{ij}(\tilde{\pi}, \tilde{\gamma}) = \min \quad u_{ij}\alpha_{ij} + \sum_{k \in K} b_{ij}^k \beta_{ij}^k,$$

Sujet à :

$$\begin{aligned} \beta_{ij}^k + \alpha_{ij} &\geq \tilde{\pi}_j^k - \tilde{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij}, \quad \forall k \in K, \\ \alpha_{ij} &\geq 0, \\ \beta_{ij}^k &\geq 0, \quad \forall k \in K. \end{aligned}$$

Or, ce problème n'est autre que le dual du problème auxiliaire $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$. Nous utiliserons par conséquent la solution primale optimale du problème auxiliaire $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$ trouvée par la procédure "Sac à dos auxiliaire primal" et la théorie des écarts complémentaires pour trouver la solution optimale $(\bar{\alpha}, \bar{\beta})$ du problème $D_{\alpha\beta}^{ij}(\tilde{\pi}, \tilde{\gamma})$.

Solution duale optimale $(\bar{\alpha}, \bar{\beta})$ du problème auxiliaire $D_{\alpha\beta}^{ij}(\tilde{\pi}, \tilde{\gamma})$

Les écarts complémentaires, qui nous permettent d'identifier une solution duale optimale du problème $D_{\alpha\beta}^{ij}(\tilde{\pi}, \tilde{\gamma})$, s'expriment ainsi :

$$(u_{ij} - \sum_{k \in K} x_{ij}^k) \alpha_{ij} = 0, \quad (2.18)$$

$$(b_{ij}^k - x_{ij}^k) \beta_{ij}^k = 0, \quad \forall k \in K, \quad (2.19)$$

$$(c_{ij}^k + \tilde{\pi}_i - \tilde{\pi}_j + \tilde{\gamma}_{ij} + \beta_{ij}^k + \alpha_{ij}) x_{ij}^k = 0, \quad \forall k \in K, \quad (2.20)$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij}, \quad (2.21)$$

$$0 \leq x_{ij}^k \leq b_{ij}^k, \quad \forall k \in K, \quad (2.22)$$

$$\beta_{ij}^k + \alpha_{ij} \geq \tilde{\pi}_j - \tilde{\pi}_i - c_{ij}^k - \tilde{\gamma}_{ij}, \quad \forall k \in K, \quad (2.23)$$

$$\alpha_{ij} \geq 0, \quad (2.24)$$

$$\beta_{ij}^k \geq 0, \quad \forall k \in K. \quad (2.25)$$

La solution primale optimale du problème $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$ trouvée par la procédure "Sac à dos auxiliaire primal", \hat{x}_{ij}^k , satisfait les conditions (2.21) et (2.22) puisqu'elle est réalisable. Nous devons donc trouver une solution duale réalisable (c'est-à-dire qui satisfait les conditions (2.23), (2.24) et (2.25)) qui satisfera les conditions (2.18), (2.19) et (2.20) des écarts complémentaires.

Pour un arc (i, j) donné, si la capacité du sac à dos est assez grande pour contenir tous les produits à coût strictement négatif, nous avons alors $\sum_{k \in K} \hat{x}_{ij}^k < u_{ij}$ et nous devons poser $\bar{\alpha}_{ij} = 0$ pour satisfaire la condition (2.18). Puis, pour chaque produit $k \in K$, si le produit n'est pas sélectionné, alors :

- nous devons poser $\bar{\beta}_{ij}^k = 0$ pour satisfaire les conditions (2.19), puisque $\hat{x}_{ij}^k \neq b_{ij}^k$;
- les conditions (2.20) sont satisfaites puisque $\hat{x}_{ij}^k = 0$;
- les conditions (2.23) sont satisfaites puisque $\bar{\alpha}_{ij} = 0$, $\bar{\beta}_{ij}^k = 0$ et le coût $c_{ij}^k + \tilde{\pi}_i - \tilde{\pi}_j + \tilde{\gamma}_{ij}$ est positif (étant donné que le produit n'entre pas dans le sac).

Si le produit est pris à pleine capacité, alors :

- les conditions (2.19) sont satisfaites puisque $\hat{x}_{ij}^k = b_{ij}^k$;

- nous devons poser $\bar{\beta}_{ij}^k = \bar{\pi}_j^k - \bar{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} > 0$ pour satisfaire les conditions (2.20), puisque $\hat{x}_{ij}^k \neq 0$;
- les conditions (2.23) sont satisfaites puisque $\bar{\alpha}_{ij} = 0$ et $\bar{\beta}_{ij}^k = \bar{\pi}_j^k - \bar{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij}$.

Si, par contre, la capacité du sac à dos est trop petite pour contenir tous les produits à coût strictement négatif, nous avons alors $\sum_{k \in K} \hat{x}_{ij}^k = u_{ij}$ et la condition (2.18) est satisfaite. Nous pouvons alors poser $\bar{\alpha}_{ij} = \bar{\pi}_j^{k^*} - \bar{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij}$, c'est-à-dire donner à $\bar{\alpha}_{ij}$ l'inverse de la valeur du coût (négatif) de l'élément critique. Puis, pour chaque produit $k \in K$, si le produit n'est pas sélectionné, alors :

- nous devons poser $\bar{\beta}_{ij}^k = 0$ pour satisfaire les conditions (2.19), puisque $\hat{x}_{ij}^k \neq b_{ij}^k$;
- les conditions (2.20) sont satisfaites puisque $\hat{x}_{ij}^k = 0$;
- les conditions (2.23) sont satisfaites puisque $\bar{\beta}_{ij}^k = 0$ et que $\bar{\alpha}_{ij}$ prend la valeur inverse de la valeur du coût de l'élément critique, qui sera nécessairement plus grande ou égale que la valeur inverse du coût des produits qui ne sont pas présents dans le sac.

Si le produit est pris à pleine capacité, alors :

- les conditions (2.19) sont satisfaites puisque $\hat{x}_{ij}^k = b_{ij}^k$;
- nous devons poser $\bar{\beta}_{ij}^k = \bar{\pi}_j^k - \bar{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} - \bar{\alpha}_{ij} > 0$ pour satisfaire les conditions (2.20), puisque $\hat{x}_{ij}^k \neq 0$;
- les conditions (2.23) sont satisfaites puisque

$$\begin{aligned} & \bar{\beta}_{ij}^k + \bar{\alpha}_{ij} \geq \bar{\pi}_j^k - \bar{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} \\ \Leftrightarrow & \bar{\pi}_j^k - \bar{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} - \bar{\alpha}_{ij} + \bar{\alpha}_{ij} \geq \bar{\pi}_j^k - \bar{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} \\ \Leftrightarrow & \bar{\pi}_j^k - \bar{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} \geq \bar{\pi}_j^k - \bar{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} \\ \Leftrightarrow & 0 \geq 0. \end{aligned}$$

Pour l'élément critique k^* ,

- puisque l'élément critique est nécessairement pris dans le sac ($\hat{x}_{ij}^{k^*} \neq 0$), nous devons, pour satisfaire les conditions (2.20), donner la valeur suivante à $\bar{\beta}_{ij}^{k^*}$:

$$\begin{aligned} & c_{ij}^{k^*} + \bar{\pi}_i^{k^*} - \bar{\pi}_j^{k^*} + \tilde{\gamma}_{ij} + \bar{\beta}_{ij}^{k^*} + \bar{\alpha}_{ij} = 0 \\ \Leftrightarrow & c_{ij}^{k^*} + \bar{\pi}_i^{k^*} - \bar{\pi}_j^{k^*} + \tilde{\gamma}_{ij} + \bar{\pi}_j^{k^*} - \bar{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij} + \bar{\beta}_{ij}^{k^*} = 0 \\ \Leftrightarrow & \bar{\beta}_{ij}^{k^*} = 0; \end{aligned}$$

- puisque $\bar{\beta}_{ij}^{k^*} = 0$, les conditions (2.19) sont satisfaites ;
- les conditions (2.23) sont satisfaites puisque

$$\begin{aligned}
& \bar{\beta}_{ij}^{k^*} + \bar{\alpha}_{ij} \geq \tilde{\pi}_j^{k^*} - \tilde{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij} \\
\Leftrightarrow & \bar{\alpha}_{ij} \geq \tilde{\pi}_j^{k^*} - \tilde{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij} \\
\Leftrightarrow & \tilde{\pi}_j^{k^*} - \tilde{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij} \geq \tilde{\pi}_j^{k^*} - \tilde{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij} \\
\Leftrightarrow & 0 \geq 0.
\end{aligned}$$

Toutes les variables $\bar{\alpha}_{ij}$ et $\bar{\beta}_{ij}^k$ sont non négatives ; la solution duale trouvée satisfait donc toutes les conditions des écarts complémentaires, et est par conséquent optimale. Le pseudocode suivant détaille la procédure de déduction d'une solution duale optimale du problème de sac à dos auxiliaire $D_{\alpha\beta}^{ij}(\tilde{\pi}, \tilde{\gamma})$ à partir d'une solution primale optimale \hat{x}_{ij}^k du problème $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$:

Procédure "Sac à dos auxiliaire dual"

Si $\sum_{k \in K} \hat{x}_{ij}^k < u_{ij}$ **alors**

$$\bar{\alpha}_{ij} := 0;$$

Pour chaque $k \in K$ **faire**

Si $c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij} < 0$ **alors**

$$\bar{\beta}_{ij}^k := \tilde{\pi}_j^k - \tilde{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij};$$

Sinon

$$\bar{\beta}_{ij}^k := 0;$$

Sinon

$$\bar{\alpha}_{ij} := \tilde{\pi}_j^{k^*} - \tilde{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij};$$

Pour chaque $k \in K$ **faire**

Si $k < k^*$ **dans le tableau** T **alors**

$$\bar{\beta}_{ij}^k := \tilde{\pi}_j^k - \tilde{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} - \bar{\alpha}_{ij};$$

Sinon

$$\bar{\beta}_{ij}^k := 0.$$

Solution du sous-problème de sac à dos continu

L'algorithme "Sac à dos" résout en deux étapes un sous-problème de sac à dos pour chaque arc $(i, j) \in A$. Dans un premier temps, la procédure "Sac à dos primal" nous permet de trouver une solution primale optimale du sous-problème $P^{ij}(\tilde{\pi}, \tilde{\gamma})$. Puis, nous utilisons cette solution primale optimale et la théorie des écarts complémentaires pour déduire une solution duale optimale du sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$.

Algorithme "Sac à dos"

Pour chaque arc $(i, j) \in A$ faire

Étape 1 : Trouver les solutions optimales (\bar{x}, \bar{y}) et \hat{x} de $P^{ij}(\tilde{\pi}, \tilde{\gamma})$ et $P_x^{ij}(\tilde{\pi}, \tilde{\gamma})$:

$(\hat{x}_{ij}^k, \bar{x}_{ij}^k, \bar{y}_{ij}) \leftarrow$ Sac à dos primal (\cdot).

Étape 2 : Déduire une solution duale optimale de $D^{ij}(\tilde{\pi}, \tilde{\gamma})$:

Si $\bar{y}_{ij} = 0$ alors

$\bar{\lambda}_{ij} := 0$;

$(\bar{\alpha}_{ij}, \bar{\beta}_{ij}^k) \leftarrow$ Sac à dos auxiliaire dual (\hat{x}_{ij}^k) ;

Sinon si $\sum_{k \in K} \bar{x}_{ij}^k < u_{ij}$ alors

$\bar{\alpha}_{ij} := 0$;

$$\bar{\beta}_{ij}^k := \begin{cases} 0, & \forall k \in \{k | \bar{x}_{ij}^k = 0\}; \\ \tilde{\pi}_j^k - \tilde{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij}, & \forall k \in \{k | \bar{x}_{ij}^k = b_{ij}^k\}; \end{cases}$$

Si $f_{ij} - \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k \leq 0$ alors

$\bar{\lambda}_{ij} := \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k - f_{ij}$;

Sinon

$\bar{\alpha}_{ij} := \tilde{\pi}_j^{k^*} - \tilde{\pi}_i^{k^*} - c_{ij}^{k^*} - \tilde{\gamma}_{ij}$;

$$\bar{\beta}_{ij}^k := \begin{cases} 0, & \forall k \in \{k | \bar{x}_{ij}^k = 0 \text{ ou } k = k^*\}; \\ \tilde{\pi}_j^k - \tilde{\pi}_i^k - c_{ij}^k - \tilde{\gamma}_{ij} - \bar{\alpha}_{ij}, & \forall k \in \{k | \bar{x}_{ij}^k = b_{ij}^k\}; \end{cases}$$

Si $f_{ij} - u_{ij} \bar{\alpha}_{ij} - \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k \leq 0$ alors

$\bar{\lambda}_{ij} := u_{ij} \bar{\alpha}_{ij} + \sum_{k \in K} b_{ij}^k \bar{\beta}_{ij}^k - f_{ij}$.

2.2.4 Sous-problème de multiflot

Dans le troisième type de sous-problème, nous fixons les variables α_{ij} , λ_{ij} et β_{ij}^k . Nous tentons alors de modifier les autres variables duales (π_i^k et γ_{ij}) de façon à augmenter l'objectif du dual. Le dual correspondant $D(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})$ s'écrit alors :

$$Z_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta}) = \max \sum_{k \in K} d^k \pi_{D(k)}^k - \sum_{(i,j) \in A} u_{ij} \gamma_{ij} - \sum_{(i,j) \in A} \tilde{\lambda}_{ij}, \quad (2.26)$$

Sujet à :

$$\pi_j^k - \pi_i^k - \gamma_{ij} \leq c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\beta}_{ij}^k, \quad \forall (i, j) \in A, k \in K, \quad (2.27)$$

$$0 \leq f_{ij} - u_{ij} \tilde{\alpha}_{ij} - \sum_{k \in K} b_{ij}^k \tilde{\beta}_{ij}^k + \tilde{\lambda}_{ij}, \quad \forall (i, j) \in A, \quad (2.28)$$

$$\gamma_{ij} \geq 0, \quad \forall (i, j) \in A.$$

Les contraintes de type (2.28) sont maintenant inutiles puisque f_{ij} est une constante, tandis que les variables $\tilde{\alpha}_{ij}$, $\tilde{\lambda}_{ij}$ et $\tilde{\beta}_{ij}^k$ sont fixées ; aucune variable ne peut être modifiée dans ces contraintes, qui demeurent donc satisfaites. On peut par conséquent simplifier le sous-problème à résoudre ainsi :

$$Z_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta}) = \max \sum_{k \in K} d^k \pi_{D(k)}^k - \sum_{(i,j) \in A} u_{ij} \gamma_{ij} - \sum_{(i,j) \in A} \tilde{\lambda}_{ij},$$

Sujet à :

$$\pi_j^k - \pi_i^k - \gamma_{ij} \leq c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\beta}_{ij}^k, \quad \forall (i, j) \in A, k \in K,$$

$$\gamma_{ij} \geq 0, \quad \forall (i, j) \in A.$$

Or, le primal de ce sous-problème, $P(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})$, est un problème de *multiflot à coût*

minimum, avec des coûts de transport modifiés $\hat{c}_{ij}^k = c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\beta}_{ij}^k$:

$$Z_p(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta}) = \min \sum_{(i,j) \in A} \sum_{k \in K} (c_{ij}^k + \tilde{\alpha}_{ij} + \tilde{\beta}_{ij}^k) x_{ij}^k - \sum_{(i,j) \in A} \tilde{\lambda}_{ij},$$

Sujet à :

$$\sum_{j \in N | (j,i) \in A} x_{ji}^k - \sum_{j \in N | (i,j) \in A} x_{ij}^k = \begin{cases} -d^k, & i = O(k), \\ d^k, & i = D(k), \\ 0, & i \neq O(k), D(k), \end{cases} \quad \begin{matrix} \forall i \in N, \\ k \in K, \end{matrix}$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij}, \quad \forall (i,j) \in A,$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, k \in K.$$

Nous espérons ici être en mesure d'augmenter l'objectif du dual $D(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})$ en résolvant le sous-problème de multiflot avec une méthode de montée duale spécialisée, appelée par la procédure *DualMultiflot*. Nous avons implanté deux versions de cette procédure : l'une résout la formulation primale du sous-problème de multiflot, $P(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})$, avec l'algorithme dual du simplexe et l'autre résout la formulation duale du sous-problème de multiflot, $D(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})$, avec l'algorithme primal du simplexe. Notons qu'en théorie, les deux approches sont équivalentes, c'est-à-dire qu'on peut simuler le comportement de l'une en utilisant l'autre. Toutefois, avec l'implantation du simplexe que nous utilisons (CPLEX [5]), les deux approches se comportent différemment, en raison des méthodes utilisées pour choisir la variable d'entrée dans un pivot. Une série de tests préliminaires nous a permis de constater que notre heuristique de décomposition par montée duale obtenait des bornes de qualité équivalente peu importe la version de la procédure *DualMultiflot*. Par contre, la version dans laquelle nous utilisons l'algorithme primal du simplexe pour résoudre la formulation duale était en moyenne plus rapide. Nous avons donc choisi d'utiliser cette version pour l'ensemble des tests effectués dans ce mémoire.

Notons que, comme nous utilisons une méthode duale de résolution du sous-problème de multiflot, il n'est pas nécessaire de résoudre celui-ci à l'optimum. En effet, n'importe quelle solution duale réalisable du sous-problème de multiflot qui a une valeur d'objectif $\bar{Z}_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta}) > \bar{Z}_d$ peut être utilisée pour mettre à jour les variables π_i^k et γ_{ij} ,

et ainsi faire augmenter l'objectif du dual. Par conséquent, nous pouvons limiter le nombre d'itérations de la méthode duale spécialisée de façon à obtenir une solution duale réalisable, tout en évitant de consacrer trop de temps au sous-problème de multiflot à chaque itération de la méthode de montée duale pure.

Solution du sous-problème de multiflot

Notre procédure appelle la méthode duale spécialisée pour la résolution du sous-problème de multiflot en y injectant :

- les variables duales fixées $\tilde{\alpha}_{ij}$, $\tilde{\lambda}_{ij}$ et $\tilde{\beta}_{ij}^k$ pour calculer les coûts de transport modifiés \tilde{c}_{ij}^k ;
- les variables duales courantes $\hat{\gamma}_{ij}$ et $\hat{\pi}_i^k$ pour donner une solution duale réalisable de départ à la méthode spécialisée et qu'ainsi, $\bar{Z}_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})$ soit assurément supérieur ou égal à \bar{Z}_d .

Après un certain nombre d'itérations t (donné en paramètre), si la valeur de l'objectif du sous-problème de multiflot $\bar{Z}_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})$ est supérieure à la valeur de l'objectif courant \bar{Z}_d , alors les valeurs des variables duales du sous-problème de multiflot sont utilisées pour mettre à jour les variables γ_{ij} et π_i^k , et la procédure "Multiflot" se termine. Si la valeur de l'objectif du sous-problème de multiflot $\bar{Z}_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})$ est égale à la valeur de l'objectif courant \bar{Z}_d , alors les trois cas suivants peuvent survenir :

- si les valeurs des variables duales du sous-problème de multiflot sont différentes des valeurs initiales, alors les variables duales sont mises à jour en utilisant ces valeurs, et la procédure "Multiflot" se termine ; dans ce cas, l'objectif du dual demeure inchangé par le sous-problème de multiflot, mais les nouvelles valeurs des variables duales devraient permettre aux autres sous-problèmes de faire progresser l'objectif du dual ;
- si les valeurs des variables duales du sous-problème de multiflot sont identiques aux valeurs initiales et que le multiflot a été résolu à l'optimum, alors la procédure "Multiflot" se termine ;
- si les valeurs des variables duales du sous-problème de multiflot sont identiques aux valeurs initiales et que le multiflot n'a pas été résolu à l'optimum, alors la

résolution du sous-problème de multiflot se poursuit pour t itérations supplémentaires.

Procédure “Multiflot”

$$(\bar{\gamma}_{ij}, \bar{\pi}_i^k, \bar{Z}_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})) \leftarrow \text{DualMultiflot}(\hat{c}_{ij}^k, \hat{\gamma}_{ij}, \hat{\pi}_i^k, t);$$

Tant que (STOP n'est pas atteint) **faire** :

Si $(\bar{Z}_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta}) > \bar{Z}_d)$ **ou** $(\bar{\pi} \neq \hat{\pi} \text{ ou } \bar{\gamma} \neq \hat{\gamma})$ **faire** :

$$\pi_i^k \leftarrow \bar{\pi}_i^k, \quad \forall k \in K, i \in N;$$

$$\gamma_{ij} \leftarrow \bar{\gamma}_{ij}, \quad \forall (i, j) \in A;$$

$$\bar{Z}_d \leftarrow \bar{Z}_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta});$$

$$\bar{c}_{ij}^k \leftarrow \bar{c}_{ij}^k + (\bar{\gamma}_{ij} - \gamma_{ij}) + (\bar{\pi}_i^k - \pi_i^k) - (\bar{\pi}_j^k - \pi_j^k), \quad \forall (i, j) \in A, k \in K;$$

STOP;

Sinon si (l'optimum du multiflot a été atteint) **faire** :

STOP;

Sinon faire :

$$\hat{\pi}_i^k \leftarrow \bar{\pi}_i^k, \quad \forall k \in K, i \in N;$$

$$\hat{\gamma}_{ij} \leftarrow \bar{\gamma}_{ij}, \quad \forall (i, j) \in A;$$

$$(\bar{\gamma}_{ij}, \bar{\pi}_i^k, \bar{Z}_d(\tilde{\alpha}, \tilde{\lambda}, \tilde{\beta})) \leftarrow \text{DualMultiflot}(\hat{c}_{ij}^k, \hat{\gamma}_{ij}, \hat{\pi}_i^k, t).$$

2.3 Combinaison avec l'approche primale-duale

L'approche primale-duale est une des méthodes utilisées, l'autre étant la méthode des faisceaux (voir section 2.4), pour redémarrer la montée duale pure lorsque celle-ci ne réussit plus à faire augmenter l'objectif du dual. Nous combinons les deux méthodes, montée duale pure et approche primale-duale, de la façon suivante :

Montée duale pure;

Tant que (l'objectif du dual augmente)

Approche primale-duale;

Montée duale pure ;

Les différentes étapes de l'approche primale-duale peuvent s'énoncer ainsi :

- résoudre un sous-problème de multiflot (que nous définissons plus loin), en appelant la procédure *PrimalMultiflot*, pour déduire une solution primale réalisable du problème original ;
- fixer un certain nombre de variables duales afin de satisfaire les conditions des écarts complémentaires le plus possible ;
- repartir la montée duale pure.

On espère ainsi guider la montée duale pure vers une meilleure solution grâce à ces variables duales fixées pour satisfaire les écarts complémentaires.

Les conditions des écarts complémentaires entre les formulations primales et duales s'expriment ainsi :

$$(c_{ij}^k - \pi_j^k + \pi_i^k + \alpha_{ij} + \beta_{ij}^k + \gamma_{ij})x_{ij}^k = 0, \quad \forall(i, j) \in A, k \in K, \quad (2.29)$$

$$(f_{ij} - u_{ij}\alpha_{ij} - \sum_{k \in K} b_{ij}^k \beta_{ij}^k + \lambda_{ij})y_{ij} = 0, \quad \forall(i, j) \in A, \quad (2.30)$$

$$(u_{ij}y_{ij} - \sum_{k \in K} x_{ij}^k)\alpha_{ij} = 0, \quad \forall(i, j) \in A, \quad (2.31)$$

$$(b_{ij}^k y_{ij} - x_{ij}^k)\beta_{ij}^k = 0, \quad \forall(i, j) \in A, k \in K, \quad (2.32)$$

$$(u_{ij} - \sum_{k \in K} x_{ij}^k)\gamma_{ij} = 0, \quad \forall(i, j) \in A, \quad (2.33)$$

$$(1 - y_{ij})\lambda_{ij} = 0, \quad \forall(i, j) \in A. \quad (2.34)$$

Nous avons implanté deux versions différentes de l'approche primale-duale : l'approche primale-duale avec primal restreint et l'approche primale-duale avec primal non restreint. Les deux prochaines sous-sections décrivent ces deux versions qui diffèrent l'une de l'autre par le sous-problème de multiflot résolu. Les deux versions seront testées et comparées au chapitre 3.

2.3.1 Approche primale-duale avec primal restreint

Pour trouver une solution primale réalisable du problème original, nous résolvons le sous-problème de multiflot suivant à l'aide d'une méthode spécialisée :

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} \sum_{k \in K} \hat{c}_{ij}^k x_{ij}^k, \\ \text{Sujet à :} \quad & \\ \sum_{j \in N | (j,i) \in A} x_{ji}^k - \sum_{j \in N | (i,j) \in A} x_{ij}^k = & \begin{cases} -d^k, & i = O(k), \\ d^k, & i = D(k), \\ 0, & i \neq O(k), D(k), \end{cases} & \forall i \in N, \\ \sum_{k \in K} x_{ij}^k \leq \hat{u}_{ij}, & \forall (i,j) \in A, \\ 0 \leq x_{ij}^k \leq b_{ij}^k, & \forall (i,j) \in A, k \in K. \end{aligned}$$

où les coûts de transports modifiés sont donnés par $\hat{c}_{ij}^k = c_{ij}^k + \alpha_{ij} + \beta_{ij}^k$ et où on empêche l'utilisation de l'arc (i, j) en forçant $\hat{u}_{ij} = 0$ si $\bar{f}_{ij} > 0$ (à la fin de la montée duale pure) pour satisfaire les conditions (2.30) des écarts complémentaires. Notons par contre que si trop d'arcs sont prohibés, il se peut que le problème ne soit plus réalisable, auquel cas l'approche primale-duale avec primal restreint mène à un cul-de-sac. Si le problème demeure réalisable malgré l'interdiction d'utiliser certains arcs, alors la solution primale optimale de ce problème de multiflot nous donne la valeur des variables de flot, \bar{x}_{ij}^k , et il nous suffit de poser

$$\bar{y}_{ij} = \begin{cases} 1, & \text{si } \sum_{k \in K} \bar{x}_{ij}^k > 0, \\ 0, & \text{sinon,} \end{cases}$$

pour obtenir une solution primale réalisable du problème original. Notons que cette solution primale réalisable $(\bar{x}_{ij}^k, \bar{y}_{ij})$ nous permet de calculer une borne supérieure pour le problème original : $\sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \bar{x}_{ij}^k + \sum_{(i,j) \in A} f_{ij} \bar{y}_{ij}$.

Comme le problème de multiflot est résolu à l'optimalité lorsqu'il est réalisable, les conditions (2.29) et (2.33) des écarts complémentaires sont satisfaites. De plus, comme nous avons fermé les arcs (i, j) tels que $\bar{f}_{ij} > 0$, les conditions (2.30) des écarts complémentaires sont aussi satisfaites. Les conditions (2.31), (2.32) et (2.34) ne sont quant à elles pas nécessairement satisfaites ; nous allons donc tenter de fixer

certaines variables duales de façon à satisfaire le plus possible ces conditions des écarts complémentaires.

Fixation des variables duales

Pour chacune des conditions des écarts complémentaires qui peut être violée, nous allons examiner comment nous pouvons fixer certaines variables duales pour empêcher que cela se produise.

Pour la condition (2.31),

- la condition est nécessairement satisfaite si $\bar{y}_{ij} = 0$ ($\Rightarrow \bar{x}_{ij}^k = 0, \quad \forall k \in K$) ou si $\sum_{k \in K} \bar{x}_{ij}^k = u_{ij}$, puisque dans ces deux cas $(u_{ij}\bar{y}_{ij} - \sum_{k \in K} \bar{x}_{ij}^k) = 0$;
- si $0 < \sum_{k \in K} \bar{x}_{ij}^k < u_{ij}$, alors on doit absolument poser $\bar{\alpha}_{ij} = 0$ pour satisfaire la condition.

Pour la condition (2.32),

- la condition est nécessairement satisfaite si $\bar{y}_{ij} = 0$ ($\Rightarrow \bar{x}_{ij}^k = 0, \quad \forall k \in K$) ou si $\bar{x}_{ij}^k = b_{ij}^k$, puisque dans ces deux cas $(b_{ij}^k\bar{y}_{ij} - \bar{x}_{ij}^k) = 0$;
- si $\bar{x}_{ij}^k < b_{ij}^k$ et $\bar{y}_{ij} = 1$, alors on doit absolument poser $\bar{\beta}_{ij}^k = 0$ pour satisfaire la condition.

Pour la condition (2.34),

- la condition est nécessairement satisfaite si $\bar{y}_{ij} = 1$ puisque dans ce cas $(1 - \bar{y}_{ij}) = 0$;
- si $\bar{y}_{ij} = 0$, alors on doit absolument poser $\bar{\lambda}_{ij} = 0$ pour satisfaire la condition.

Comme les variables duales β_{ij}^k et λ_{ij} sont déjà mises à zéro lors de l'initialisation (voir section 2.2.1), il suffit de faire une initialisation normale, puis de poser $\bar{\alpha}_{ij} = 0$ si $0 < \sum_{k \in K} \bar{x}_{ij}^k < u_{ij}$ avant de repartir la montée duale pure. Soit nous permettons ensuite aux variables $\bar{\beta}_{ij}^k$, $\bar{\alpha}_{ij}$ et $\bar{\lambda}_{ij}$ fixées à 0 d'être modifiées par la suite, soit nous les gardons fixées ; ces deux variantes seront testées et comparées au chapitre 3.

Garder les variables $\tilde{\beta}_{ij}^k$, $\tilde{\alpha}_{ij}$ et $\tilde{\lambda}_{ij}$ fixées

Si certaines variables $\bar{\beta}_{ij}^k$, $\bar{\alpha}_{ij}$ et $\bar{\lambda}_{ij}$ doivent demeurer fixées à 0, il nous faut alors apporter les modifications suivantes à la montée duale pure :

- Dans le sous-problème de conception de réseaux sans capacité, les variables α_{ij} et λ_{ij} sont fixées tandis que les variables β_{ij}^k sont mises à jour ainsi :

$$\beta_{ij}^k \leftarrow \beta_{ij}^k + \delta, \forall (i, j) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k = 0\}.$$

Pour chaque produit $k \in K$, il suffit donc d'imposer $\delta = 0$ s'il existe un arc $(i, j) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k = 0\}$ tel que β_{ij}^k est fixé, pour empêcher que cette variable ne soit modifiée.

- Dans le sous-problème de multiflot, aucun changement n'est requis puisque les variables β_{ij}^k , α_{ij} et λ_{ij} sont déjà fixées.
- Dans le sous-problème de sac à dos, les quatre cas suivants peuvent survenir :
 1. Si β_{ij}^k est fixé à 0 pour tous les $k \in \bar{K} \subseteq K$, alors le sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ et son primal, $P^{ij}(\tilde{\pi}, \tilde{\gamma})$, deviennent :

$$\begin{array}{l|l} \max -\lambda_{ij}, & \min \sum_{k \in K \setminus \bar{K}} \hat{c}_{ij}^k x_{ij}^k + f_{ij} y_{ij} + \sum_{k \in \bar{K}} \hat{c}_{ij}^k w_{ij}^k, \\ \text{Sujet à :} & \text{Sujet à :} \\ -\alpha_{ij} - \beta_{ij}^k \leq \hat{c}_{ij}^k, \quad \forall k \in K \setminus \bar{K}, & \sum_{k \in K \setminus \bar{K}} x_{ij}^k + \sum_{k \in \bar{K}} w_{ij}^k \leq u_{ij} y_{ij}, \\ -\alpha_{ij} \leq \hat{c}_{ij}^k, \quad \forall k \in \bar{K}, & x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall k \in K \setminus \bar{K}, \\ f_{ij} \geq -\lambda_{ij} + u_{ij} \alpha_{ij} + \sum_{k \in K \setminus \bar{K}} b_{ij}^k \beta_{ij}^k, & y_{ij} \leq 1, \\ \alpha_{ij}, \lambda_{ij} \geq 0, & w_{ij}^k \geq 0, \forall k \in \bar{K}, \\ \beta_{ij}^k \geq 0, \quad \forall k \in K \setminus \bar{K}, & x_{ij}^k \geq 0, \forall k \in K \setminus \bar{K}, \\ & y_{ij} \geq 0, \end{array}$$

où $\hat{c}_{ij}^k = c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}$.

Nous remarquons qu'il s'agit du même problème que le sous-problème de sac à dos original, sauf que les items $w_{ij}^k, \forall k \in \bar{K}$, n'ont pas de capacité partielle. Il suffit donc de poser $b_{ij}^k = u_{ij}, \forall k \in \bar{K}$, et de résoudre le problème comme précédemment

pour garder les variables β_{ij}^k fixées à 0, $\forall k \in \overline{K}$.

2. Si α_{ij} est fixé à 0, alors le sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ et son primal, $P^{ij}(\tilde{\pi}, \tilde{\gamma})$, deviennent :

$$\begin{array}{l|l} \max -\lambda_{ij}, & \min \sum_{k \in K} \hat{c}_{ij}^k x_{ij}^k + f_{ij} y_{ij}, \\ \text{Sujet à :} & \text{Sujet à :} \\ -\beta_{ij}^k \leq \hat{c}_{ij}^k, \forall k \in K, & x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall k \in K, \\ f_{ij} \geq -\lambda_{ij} + \sum_{k \in K} b_{ij}^k \beta_{ij}^k, & y_{ij} \leq 1, \\ \lambda_{ij} \geq 0, & y_{ij} \geq 0, \\ \beta_{ij}^k \geq 0, \quad \forall k \in K, & x_{ij}^k \geq 0, \quad \forall k \in K, \end{array}$$

où $\hat{c}_{ij}^k = c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}$.

Nous remarquons qu'il s'agit du même problème que le sous-problème de sac à dos original, sauf que la capacité du sac à dos est infinie. Il suffit donc de poser $u_{ij} = \sum_{k \in K} b_{ij}^k$ et de résoudre le problème comme précédemment pour garder la variable α_{ij} fixée à 0.

3. Si α_{ij} est fixé à 0 et β_{ij}^k est fixé à 0 pour tous les $k \in \overline{K} \subseteq K$, alors le sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ et son primal, $P^{ij}(\tilde{\pi}, \tilde{\gamma})$, deviennent :

$$\begin{array}{l|l} \max -\lambda_{ij}, & \min \sum_{k \in K \setminus \overline{K}} \hat{c}_{ij}^k x_{ij}^k + f_{ij} y_{ij}, \\ \text{Sujet à :} & \text{Sujet à :} \\ -\beta_{ij}^k \leq \hat{c}_{ij}^k, \quad \forall k \in K \setminus \overline{K}, & x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall k \in K \setminus \overline{K}, \\ f_{ij} \geq -\lambda_{ij} + \sum_{k \in K \setminus \overline{K}} b_{ij}^k \beta_{ij}^k, & y_{ij} \leq 1, \\ \lambda_{ij} \geq 0, & y_{ij} \geq 0, \\ \beta_{ij}^k \geq 0, \quad \forall k \in K \setminus \overline{K}, & x_{ij}^k \geq 0, \quad \forall k \in K \setminus \overline{K}, \end{array}$$

où $\hat{c}_{ij}^k = c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}$.

Nous remarquons qu'il s'agit du même problème que le sous-problème de sac à dos original, sauf que la capacité du sac à dos est infinie et que les items k tels que $k \in \overline{K}$ n'ont pas de capacité partielle. Il suffit donc de poser $PROFIT_item[k] = 0, \forall k \in \overline{K}$, pour que le problème soit borné inférieurement et que les variables

$\beta_{ij}^k, \forall k \in \overline{K}$, demeurent fixées à 0, puis de poser $u_{ij} = \sum_{k \in K} b_{ij}^k$ pour garder la variable α_{ij} fixée à 0.

4. Si λ_{ij} est fixé à 0, alors le sous-problème $D^{ij}(\tilde{\pi}, \tilde{\gamma})$ et son primal, $P^{ij}(\tilde{\pi}, \tilde{\gamma})$, deviennent :

$$\begin{array}{l|l} \max & 0, \\ \text{Sujet à :} & \\ -\alpha_{ij} - \beta_{ij}^k \leq \hat{c}_{ij}^k, & \forall k \in K, \\ f_{ij} \geq u_{ij}\alpha_{ij} + \sum_{k \in K} b_{ij}^k \beta_{ij}^k, & \\ \alpha_{ij} \geq 0, & \\ \beta_{ij}^k \geq 0, & \forall k \in K, \end{array} \quad \begin{array}{l} \min \sum_{k \in K} \hat{c}_{ij}^k x_{ij}^k + f_{ij} y_{ij}, \\ \text{Sujet à :} \\ \sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \\ x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall k \in K, \\ y_{ij} \geq 0 \\ x_{ij}^k \geq 0, \quad \forall k \in K, \end{array}$$

où $\hat{c}_{ij}^k = c_{ij}^k + \tilde{\pi}_i^k - \tilde{\pi}_j^k + \tilde{\gamma}_{ij}$.

Comme la valeur optimale de l'objectif est 0, alors la solution primale ($\bar{y}_{ij} = 0, \bar{x}_{ij}^k = 0 \quad \forall k \in K$) est optimale. Il suffit donc de poser $\bar{y}_{ij} = 0$ dans la solution primale optimale du sous-problème de sac à dos et de déduire une solution duale optimale comme précédemment pour garder la variable λ_{ij} fixée à 0.

Algorithme "Approche primale-duale avec primal restreint"

Soit F l'ensemble des variables qui doivent demeurer fixées lorsqu'on redémarre la méthode de décomposition par montée duale; alors l'algorithme "Approche primale-duale avec primal restreint" s'énonce ainsi :

Étape 1 : Résolution du multiflot :

Si le multiflot est réalisable, alors

$(\bar{x}_{ij}^k, \bar{y}_{ij}) \leftarrow \text{PrimalMultiflot}(\hat{c}_{ij}^k, \hat{u}_{ij});$

Sinon

STOP.

Étape 2 : Réinitialisation des variables duales :

$F \leftarrow \emptyset;$

$$\begin{aligned}
\alpha_{ij} &:= \tau * f_{ij}/u_{ij}, \quad \forall (i, j) \in A | \sum_{k \in K} \bar{x}_{ij}^k \in \{0, u_{ij}\}; \\
\alpha_{ij} &:= 0 \text{ et } F \leftarrow F \cup \{\alpha_{ij}\}, \quad \forall (i, j) \in A | 0 < \sum_{k \in K} \bar{x}_{ij}^k < u_{ij}; \\
\gamma_{ij} &:= 0, \quad \forall (i, j) \in A; \\
\lambda_{ij} &:= 0, \quad \forall (i, j) \in A | \bar{y}_{ij} = 1; \\
\lambda_{ij} &:= 0 \text{ et } F \leftarrow F \cup \{\lambda_{ij}\}, \quad \forall (i, j) \in A | \bar{y}_{ij} = 0; \\
\pi_i^k &:= 0, \quad \forall i \in N, k \in K; \\
\beta_{ij}^k &:= 0, \quad \forall (i, j) \in A, k \in K | \bar{y}_{ij} = 0 \text{ ou } \bar{x}_{ij}^k = b_{ij}^k; \\
\beta_{ij}^k &:= 0 \text{ et } F \leftarrow F \cup \{\beta_{ij}^k\}, \quad \forall (i, j) \in A, k \in K | \bar{x}_{ij}^k < b_{ij}^k \text{ et } \bar{y}_{ij} = 1; \\
\bar{c}_{ij}^k &:= c_{ij}^k + \alpha_{ij}, \quad \forall (i, j) \in A, k \in K; \\
\bar{f}_{ij} &:= f_{ij} - u_{ij}\alpha_{ij}, \quad \forall (i, j) \in A; \\
\bar{Z}_d &:= 0.
\end{aligned}$$

Étape 3 : Repartir la montée duale pure :

Si les variables duales α_{ij} , λ_{ij} et β_{ij}^k doivent demeurer fixées, alors

Montée duale pure (F);

Sinon

Montée duale pure (\emptyset).

2.3.2 Approche primale-duale avec primal non restreint

Contrairement à l'approche primale-duale avec primal restreint, l'approche primale-duale avec primal non restreint ne ferme aucun arc lors de la résolution du sous-problème de multiflot; celui-ci est donc toujours réalisable et l'approche primale-duale avec primal non restreint ne peut mener à un cul-de-sac.

Après avoir résolu le même sous-problème de multiflot que l'approche primale-duale avec primal restreint (mais en gardant les capacités originales u_{ij}), la solution primale réalisable (\bar{x}, \bar{y}) du problème original est déduite de la même façon que dans l'approche primale-duale avec primal restreint. Nous pouvons, dans ce cas aussi, calculer une borne supérieure pour le problème original : $\sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k \bar{x}_{ij}^k + \sum_{(i,j) \in A} f_{ij} \bar{y}_{ij}$.

Comme le multiflot est résolu à l'optimalité, les conditions (2.29) et (2.33) des écarts

complémentaires sont toujours satisfaites. Par contre, comme nous n'avons fermé aucun arc (i, j) , les conditions (2.30) des écarts complémentaires ne sont pas nécessairement satisfaites. Les conditions (2.31), (2.32) et (2.34) ne sont pas non plus nécessairement satisfaites ; nous allons donc tenter de fixer certaines variables duales de façon à satisfaire le plus possible ces conditions des écarts complémentaires.

Les conditions (2.31), (2.32) et (2.34) seront traitées de la même façon que dans l'approche primale-duale avec primal restreint. Pour la condition (2.30) :

- la condition est nécessairement satisfaite si $\bar{y}_{ij} = 0$;
- si $\bar{y}_{ij} = 1$, alors on doit absolument forcer $\bar{f}_{ij} = 0$ pour satisfaire la condition.

Forcer $\bar{f}_{ij} = 0$ si $\bar{y}_{ij} = 1$

Puisque $\bar{f}_{ij} = f_{ij} - u_{ij}\alpha_{ij} - \sum_{k \in K} b_{ij}^k \beta_{ij}^k + \lambda_{ij}$, nous allons devoir forcer $\bar{f}_{ij} = 0$ en jouant sur les variables α_{ij} , β_{ij}^k et λ_{ij} , lorsque celles-ci ne sont pas fixées, pour satisfaire les autres conditions des écarts complémentaires.

Soit le tableau suivant qui exprime les huit cas pouvant survenir pour un arc (i, j) donné. Dans le cas des variables α_{ij} ou λ_{ij} , un - indique que la variable correspondante est libre et un 0 indique qu'elle est fixée à 0 pour satisfaire les conditions (2.31), (2.32) ou (2.34). Dans le cas des variables β_{ij}^k , un - indique qu'il existe au moins un $k \in K$ tel que β_{ij}^k est libre, alors qu'un 0 indique que β_{ij}^k est fixé à 0 pour tous les $k \in K$.

Cas	I	II	III	IV	V	VI	VII	VIII
α_{ij}	0	0	-	0	0	-	-	-
λ_{ij}	0	0	0	-	-	0	-	-
$\beta_{ij}^k, \forall k$	0	-	0	0	-	-	0	-

On remarque que les trois premiers cas (I-III) ne peuvent survenir puisque λ_{ij} est fixé à 0 quand $\bar{y}_{ij} = 0$, tandis que si α_{ij} et β_{ij}^k sont fixés à 0, alors $\bar{y}_{ij} = 1$.

Si α_{ij} est fixé à zéro, λ_{ij} est libre et tous les $\beta_{ij}^k, k \in K$, sont fixés à zéro (cas IV), alors on ne peut forcer $\bar{f}_{ij} = 0$ si $f_{ij} > 0$ puisqu'il faudrait donner une valeur négative

à λ_{ij} , soit $\lambda_{ij} = -f_{ij} < 0$. Puisqu'il sera impossible dans ce cas de satisfaire toutes les conditions des écarts complémentaires, nous allons forcer $\bar{f}_{ij} = 0$ de la façon suivante :

$$\tilde{\alpha}_{ij} = \frac{f_{ij}}{u_{ij}}, \tilde{\lambda}_{ij} = 0, \tilde{\beta}_{ij}^k = 0, \forall k \in K,$$

même si la condition (2.31) ne sera alors pas nécessairement satisfaite.

Soit K_{ij}^l l'ensemble des produits $k \in K$ tels que β_{ij}^k est libre pour l'arc (i, j) . Si α_{ij} est fixé à zéro, λ_{ij} est libre et certains $\beta_{ij}^k, k \in K$, sont libres (cas V), alors on force $\bar{f}_{ij} = 0$ ainsi :

$$\tilde{\alpha}_{ij} = 0, \tilde{\lambda}_{ij} = 0, \tilde{\beta}_{ij}^k = \frac{f_{ij}}{\sum_{k \in K_{ij}^l} b_{ij}^k}, \forall k \in K_{ij}^l, \tilde{\beta}_{ij}^k = 0, \forall k \notin K_{ij}^l.$$

Si α_{ij} est libre, λ_{ij} est fixé à zéro et certains $\beta_{ij}^k, k \in K$, sont libres (cas VI) ou si les variables $\alpha_{ij}, \lambda_{ij}$ et certains $\beta_{ij}^k, k \in K$, sont libres (cas VIII), alors on force $\bar{f}_{ij} = 0$ ainsi :

$$\tilde{\alpha}_{ij} = \tau \frac{f_{ij}}{u_{ij}}, \tilde{\lambda}_{ij} = 0, \tilde{\beta}_{ij}^k = \frac{f_{ij} - u_{ij} \tilde{\alpha}_{ij}}{\sum_{k \in K_{ij}^l} b_{ij}^k}, \forall k \in K_{ij}^l, \tilde{\beta}_{ij}^k = 0, \forall k \notin K_{ij}^l,$$

où $0 \leq \tau \leq 1$ est le paramètre d'initialisation (voir section 2.2.1).

Si α_{ij} est libre, λ_{ij} est libre et tous les $\beta_{ij}^k, k \in K$, sont fixés à zéro (cas VII), alors on force $\bar{f}_{ij} = 0$ ainsi :

$$\tilde{\alpha}_{ij} = \frac{f_{ij}}{u_{ij}}, \tilde{\lambda}_{ij} = 0, \tilde{\beta}_{ij}^k = 0, \forall k \in K.$$

Après avoir réinitialisé les variables duales ainsi, nous repartons la montée duale pure et soit nous permettons aux variables $\tilde{\beta}_{ij}^k, \tilde{\alpha}_{ij}, \tilde{\lambda}_{ij}$ et \bar{f}_{ij} d'être modifiées, soit nous les gardons fixées ; ces deux variantes seront testées et comparées au chapitre 3.

Garder les variables $\tilde{\beta}_{ij}^k, \tilde{\alpha}_{ij}, \tilde{\lambda}_{ij}$ et \bar{f}_{ij} fixées

Si certaines variables $\tilde{\beta}_{ij}^k, \tilde{\alpha}_{ij}, \tilde{\lambda}_{ij}$ et \bar{f}_{ij} doivent demeurer fixées, alors soit \bar{f}_{ij} est libre et il nous suffit d'apporter les mêmes modifications que pour l'approche primale-duale avec primal restreint à la montée duale pure, soit \bar{f}_{ij} est fixée et nous devons apporter les modifications supplémentaires suivantes à la montée duale pure :

- Dans le sous-problème de conception de réseaux sans capacité, les variables α_{ij} et λ_{ij} sont fixées tandis que les variables d'écart \bar{f}_{ij} et les variables β_{ij}^k sont mises à jour ainsi :

$$\begin{aligned}\bar{f}_{ij} &\leftarrow \bar{f}_{ij} - b_{ij}^k \delta, \forall (i, j) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k = 0\}; \\ \beta_{ij}^k &\leftarrow \beta_{ij}^k + \delta, \forall (i, j) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k = 0\}.\end{aligned}$$

Pour chaque produit $k \in K$, il suffit donc d'imposer $\delta = 0$ s'il existe un arc $(i, j) \in \{A(N_u(k)) \mid \bar{c}_{ij}^k = 0\}$ tel que \bar{f}_{ij} est fixé pour empêcher que cette variable d'écart, ainsi que les variables β_{ij}^k , ne bougent.

- Si \bar{f}_{ij} est fixé, alors on ne résout pas le sous-problème de sac à dos pour cet arc (i, j) puisque les variables α_{ij} , β_{ij}^k et λ_{ij} ne peuvent être modifiées non plus.
- Dans le sous-problème de multiflot, aucun changement n'est requis puisque \bar{f}_{ij} ne peut être modifiée, les variables β_{ij}^k , α_{ij} et λ_{ij} étant déjà fixées.

Algorithme “Approche primale-duale avec primal non restreint”

Soit F l'ensemble des variables qui doivent demeurer fixées lorsqu'on redémarre la méthode de décomposition par montée duale; alors l'algorithme “Approche primale-duale avec primal non restreint” s'énonce ainsi :

Étape 1 : Résolution du multiflot :

$$(\bar{x}_{ij}^k, \bar{y}_{ij}) \leftarrow \text{PrimalMultiflot}(\hat{c}_{ij}^k).$$

Étape 2 : Réinitialisation des variables duales :

Appliquer l'Étape 2 : Réinitialisation des variables duales de l'algorithme “Approche primale-duale avec primal restreint”.

Pour chaque arc (i, j) faire

Si $\bar{y}_{ij} = 1$ alors

$$\bar{f}_{ij} := 0;$$

$$F \leftarrow F \cup \{\bar{f}_{ij}\};$$

Si tous les β_{ij}^k sont fixées, alors

$$\bar{c}_{ij}^k := \bar{c}_{ij}^k + \frac{f_{ij}}{u_{ij}} - \alpha_{ij}, \quad \forall k \in K;$$

$$\bar{\alpha}_{ij} := \frac{f_{ij}}{u_{ij}};$$

Sinon si α_{ij} est fixée, alors

$$\bar{\beta}_{ij}^k := \frac{f_{ij}}{\sum_{k \in K_{ij}^l} b_{ij}^k}, \quad \forall k \in K_{ij}^l;$$

$$\bar{c}_{ij}^k := \bar{c}_{ij}^k + \bar{\beta}_{ij}^k, \quad \forall k \in K_{ij}^l;$$

Sinon

$$\bar{\beta}_{ij}^k := \frac{f_{ij} - u_{ij} \alpha_{ij}}{\sum_{k \in K_{ij}^l} b_{ij}^k}, \quad \forall k \in K_{ij}^l;$$

$$\bar{c}_{ij}^k := \bar{c}_{ij}^k + \bar{\beta}_{ij}^k, \quad \forall k \in K_{ij}^l.$$

Étape 3 : Repartir la montée duale pure :

Si les variables duales α_{ij} , λ_{ij} , β_{ij}^k et \bar{f}_{ij} doivent demeurer fixées, alors

Montée duale pure (F);

Sinon

Montée duale pure (\emptyset).

2.4 Combinaison avec la méthode des faisceaux

La méthode des faisceaux, deuxième approche utilisée pour redémarrer l'heuristique de décomposition par montée duale, utilise la formulation suivante de la relaxation continue du PCMC, dans laquelle les contraintes redondantes de capacité pure (1.10) sont absentes, pour calculer une borne inférieure :

$$\min \sum_{(i,j) \in A} \sum_{k \in K} c_{ij}^k x_{ij}^k + \sum_{(i,j) \in A} f_{ij} y_{ij},$$

Sujet à :

$$\sum_{j \in N | (j,i) \in A} x_{ji}^k - \sum_{j \in N | (i,j) \in A} x_{ij}^k = \begin{cases} -d^k, & i = O(k), \\ d^k, & i = D(k), \\ 0, & i \neq O(k), D(k), \end{cases} \quad \begin{matrix} \forall i \in N, \\ k \in K, \end{matrix} \quad (\pi_i^k),$$

$$\sum_{k \in K} x_{ij}^k \leq u_{ij} y_{ij}, \quad \forall (i,j) \in A, \quad (\alpha_{ij} \geq 0),$$

$$x_{ij}^k \leq b_{ij}^k y_{ij}, \quad \forall (i,j) \in A, k \in K, \quad (\beta_{ij}^k \geq 0),$$

$$y_{ij} \leq 1, \quad \forall (i,j) \in A, \quad (\lambda_{ij} \geq 0),$$

$$x_{ij}^k \geq 0, \quad \forall (i,j) \in A, k \in K,$$

$$y_{ij} \geq 0, \quad \forall (i,j) \in A.$$

Comme nous l'avons mentionné au chapitre 1, la méthode des faisceaux peut être utilisée pour résoudre les duaux lagrangiens issus soit de la relaxation lagrangienne des contraintes de conservation de flot, soit de la relaxation lagrangienne des contraintes de capacité mutuelle et individuelle. Si les contraintes de conservation de flot sont relaxées, alors le sous-problème lagrangien est décomposable par arc et, pour chaque arc, il suffit de résoudre un problème de sac à dos équivalent au sous-problème de sac à dos continu résolu dans notre heuristique de décomposition par montée duale. Si les contraintes de capacité mutuelle et individuelle sont relaxées, alors le sous-problème lagrangien se résume à un problème de plus courts chemins (voir Crainic, Frangioni et Gendron [7]). Nous n'entrerons pas plus en détail dans la description de la méthode des faisceaux dans ce mémoire, puisque nous nous servons de celle-ci comme d'une "boîte noire" (voir la thèse de doctorat de Frangioni [11] pour de plus amples détails).

Lorsque notre heuristique de montée duale ne progresse plus, nous passons à la méthode des faisceaux en y injectant soit les valeurs $\bar{\pi}_i^k$ courantes comme multiplicateurs lagrangiens initiaux lorsque les contraintes de conservation de flot sont relaxées, soit les valeurs $\bar{\alpha}_{ij}$ et $\bar{\beta}_{ij}^k$ courantes comme multiplicateurs lagrangiens initiaux lorsque les contraintes de capacité sont relaxées. Nous espérons ainsi guider la méthode des

faisceaux vers une meilleure solution avec l'information duale obtenue à partir de notre heuristique de décomposition par montée duale.

Comme nous l'avons déjà mentionné, le sous-problème lagrangien issu de la relaxation des contraintes de conservation de flot est identique au sous-problème de sac à dos continu résolu dans notre méthode de décomposition par montée duale. Par conséquent, les solutions primales optimales du sous-problème de sac à dos continu obtenues à chaque itération de notre méthode de décomposition par montée duale constituent des sous-gradients relativement à cette relaxation lagrangienne. Donc, avant de démarrer la méthode des faisceaux, en plus d'utiliser les valeurs $\bar{\pi}_i^k$ comme multiplicateurs lagrangiens initiaux, nous pouvons ajouter ces sous-gradients au faisceau (voir section 1.3). Ainsi, la méthode des faisceaux devrait être en mesure d'identifier une meilleure direction de montée lors de ses premières itérations puisqu'elle ne débute pas avec un faisceau vide.

Par contre, une telle initialisation du faisceau ne peut être effectuée lorsque la relaxation des contraintes de capacité est utilisée. En effet, pour y arriver, il faudrait avoir accès aux solutions primales optimales des problèmes de plus courts chemins qui constituent le sous-problème lagrangien issu de la relaxation des contraintes de capacité. Or, ces problèmes ne sont pas résolus dans la méthode de décomposition par montée duale. Nous prévoyons donc que la méthode des faisceaux donnera de meilleurs résultats lorsque la relaxation des contraintes de conservation de flot sera utilisée.

Pour chacune des deux versions de la méthode des faisceaux (relaxation des contraintes de conservation de flot et relaxation des contraintes de capacité), nous avons développé deux approches pour la combiner avec l'heuristique de décomposition par montée duale.

Départ accéléré avec l'heuristique de montée duale

La première approche consiste à utiliser l'heuristique de décomposition par montée duale afin de fournir des valeurs initiales des multiplicateurs lagrangiens à la méthode des faisceaux et, dans le cas de la relaxation des contraintes de conservation de flot,

d'initialiser également un faisceau de sous-gradients. Ensuite, nous laissons la méthode des faisceaux prendre définitivement la relève, espérant ainsi que l'information duale générée par l'heuristique de montée duale lui permettra d'obtenir de meilleurs résultats.

Redémarrage de l'heuristique de montée duale

La deuxième approche consiste à utiliser la méthode des faisceaux pour redémarrer l'heuristique de décomposition par montée duale, comme nous le faisons avec l'approche primale-duale. Dans ce cas, lorsque l'heuristique de montée duale ne réussit plus à faire augmenter la valeur de l'objectif du dual, la méthode des faisceaux prend la relève pour un certain nombre d'itérations. Nous récupérons ensuite les valeurs des variables obtenues par la méthode des faisceaux, en s'assurant que celles-ci ne violent aucune contrainte du dual, puis nous redémarrons l'heuristique de décomposition par montée duale. Nous alternons finalement entre les deux méthodes de résolution jusqu'à ce que la valeur de l'objectif demeure inchangée par la dernière exécution de chacune des deux méthodes :

Montée duale pure ;
Tant que (l'objectif du dual augmente) {
 Méthode des faisceaux ;
 Montée duale pure ;
 }

Quand la méthode des faisceaux est utilisée pour résoudre le dual lagrangien issu de la relaxation des contraintes de conservation de flot, nous récupérons les valeurs des variables duales α_{ij} , λ_{ij} and β_{ij}^k pour les mettre à jour. De plus, les variables duales correspondant aux contraintes redondantes de capacité pure, γ_{ij} , sont absentes de la formulation duale du PCMC résolu avec la méthode des faisceaux et sont par conséquent toutes mises à zéro. Par contre, les valeurs des multiplicateurs lagrangiens π_i^k obtenues avec la méthode des faisceaux peuvent violer certaines contraintes duales (1.13). Pour s'assurer de satisfaire ces contraintes, nous résolvons un problème de plus

courts chemins pour chaque produit $k \in K$ afin d'obtenir de nouvelles valeurs réalisables pour les variables π_i^k :

$$\max d^k \pi_{D(k)}^k,$$

Sujet à :

$$\pi_j^k - \pi_i^k \leq c_{ij}^k + \alpha_{ij} + \beta_{ij}^k, \quad \forall (i, j) \in A.$$

Quand la méthode des faisceaux est utilisée pour résoudre le dual lagrangien issu de la relaxation des contraintes de capacité mutuelle et individuelle, nous récupérons les valeurs des variables duales π_i^k pour les mettre à jour. Ici aussi, les variables duales γ_{ij} sont toutes mises à zéro. Par contre, les valeurs des multiplicateurs lagrangiens α_{ij} and β_{ij}^k identifiées avec la méthode des faisceaux peuvent violer certaines contraintes duales (1.13) et (1.14). Pour s'assurer de satisfaire ces contraintes, nous résolvons un sous-problème de sac à dos continu, comme dans la méthode de décomposition par montée duale, afin d'obtenir de nouvelles valeurs réalisables pour les variables α_{ij} , λ_{ij} and β_{ij}^k .

2.5 Synthèse des approches de résolution

Cette section vise à rappeler brièvement les différentes méthodes de résolution développées dans ce mémoire et qui feront l'objet des tests du chapitre 3.

Dans un premier temps, nous avons développé une heuristique de montée duale qui applique la décomposition par montée duale au PCMC en résolvant trois sous-problèmes l'un à la suite de l'autre : un sous-problème de conception de réseaux sans capacité, un sous-problème de sac à dos continu et un sous-problème de multiflot. Les paramètres suivants devront être testés afin d'identifier des valeurs prometteuses :

- le paramètre d'initialisation τ ;
- l'ordre dans lequel les sous-problèmes sont résolus à chaque itération de l'heuristique de montée duale ;
- les deux critères d'arrêt suivants de l'heuristique de montée duale : le nombre

- le nombre d'itérations et l'amélioration minimale de l'objectif entre deux itérations ;
- le nombre d'itérations de l'algorithme du simplexe lors de la résolution du sous-problème de multiflot.

Dans un deuxième temps, nous avons développé deux heuristiques primales-duales afin de redémarrer notre méthode de montée duale pure lorsque celle-ci ne réussit plus à faire augmenter l'objectif du dual. La première version de l'approche primale-duale résout un sous-problème de multiflot dans lequel certains arcs sont prohibés pour trouver une solution primale réalisable du problème général, alors que la deuxième version résout un sous-problème de multiflot non restreint. En utilisant la solution primale réalisable trouvée par l'une ou l'autre des deux versions de l'approche primale-duale, nous fixons ensuite certaines variables duales afin de satisfaire les conditions des écarts complémentaires le plus possible, avant de redémarrer l'heuristique de montée duale. Pour chacune des deux versions, il est possible de garder les variables duales fixées ou de les laisser libres lorsque l'heuristique de montée duale est redémarrée.

Finalement, nous avons aussi utilisé la méthode des faisceaux pour redémarrer notre méthode de montée duale pure lorsque celle-ci ne réussit plus à faire augmenter l'objectif du dual. Deux versions ont été développées : une première dans laquelle la méthode des faisceaux sert à redémarrer l'heuristique de montée duale et une deuxième où l'heuristique de montée duale est utilisée uniquement comme "départ rapide" pour la méthode des faisceaux. Pour chacune de ces deux versions, nous utilisons la méthode des faisceaux pour résoudre les duaux lagrangiens issus soit de la relaxation lagrangienne des contraintes de conservation de flot, soit de la relaxation lagrangienne des contraintes de capacité mutuelle et individuelle.

Chapitre 3

Analyse et résultats

3.1 Cadre expérimental

3.1.1 Environnement

Nous avons implanté les différentes versions de notre approche de montée duale dans un programme codé en C++ pour lequel nous avons utilisé le compilateur GNU g++, version 2.95.3. La méthode des faisceaux est implantée dans un programme codé en C++ et réalisée par Antonio Frangioni. Tous les tests effectués dans le cadre de ce mémoire ont été exécutés sur une machine Sun Ultra 60 équipée de 2 GO de mémoire vive et opérant à 300 MHz.

3.1.2 Données

Nous avons effectué nos tests sur 196 problèmes obtenus avec le générateur de réseaux décrit dans Gendron et Crainic [12, 14] et utilisés dans [8, 16, 20], entre autres. Les problèmes sont divisés en deux classes. La classe C contient des problèmes ayant soit un nombre élevé de produits (considérablement plus que le nombre de noeuds : classe C-1), soit très peu de produits (habituellement moins que le nombre de noeuds : classe C-2) ; la classe R contient des problèmes à 10 ou 20 noeuds (classe R-1 et R-2, respectivement) générés pour faciliter l'analyse de la performance des méthodes relativement aux caractéristiques des problèmes. Chaque problème est caractérisé par la classe à laquelle il appartient (C ou R), son nombre de noeuds ($|N|$), son nombre d'arcs

($|A|$), son nombre de produits ($|K|$), ainsi que par l'importance de ses coûts fixes et de ses capacités. Le nom de chaque problème débute par la lettre c ou r, selon la classe de problèmes à laquelle il appartient, suivie du triplet ($|N|, |A|, |K|$).

Si le problème fait partie des 43 problèmes de la classe C, alors son nom se termine par deux lettres. La première lettre indique si les coûts fixes sont importants : un F est utilisé si les coûts fixes dominent par rapport aux coûts variables de transport et un V est utilisé pour la situation inverse. La deuxième lettre indique si les capacités des arcs sont grandes ou petites : un L est utilisé si les capacités des arcs sont grandes et les contraintes de capacités peu actives (L signifie "loose"), et un T est utilisé si les capacités des arcs sont petites et les contraintes de capacités restrictives (T signifie "tight").

Si le problème fait partie des 153 problèmes de la classe R, alors son nom se termine par une paire (F, C) indiquant l'importance des coûts fixes et des contraintes de capacités :

$$F = |K| \frac{\sum_{(i,j) \in A} f_{ij}}{\sum_{k \in K} d^k \sum_{(i,j) \in A} c_{ij}^k} \text{ et } C = |A| \frac{\sum_{k \in K} d^k}{\sum_{(i,j) \in A} u_{ij}}.$$

Quand C tend vers 1, les contraintes de capacités ne sont pas restrictives, mais elles le deviennent lorsque C augmente. Lorsque F est voisin de 0, les coûts fixes sont petits par rapport aux coûts de transport ; plus F est grand, plus les coûts fixes prennent de l'importance. Pour les problèmes de la classe R avec lesquels nous avons effectué nos tests, F prend les valeurs 0.01, 0.05 ou 0.10, et, pour chacune des valeurs prises par F , C prend les valeurs 1, 2 ou 8. Par exemple, dans un problème pour lequel $(F, C) = (0.01, 1)$, les coûts fixes ne sont pas importants par rapport aux coûts variables et les contraintes de capacités sont peu restrictives. Au contraire, dans un problème pour lequel $(F, C) = (0.10, 8)$, les coûts fixes sont très importants par rapport aux coûts variables et les contraintes de capacités sont très restrictives. Les problèmes pour lesquels $(F, C) = (0.10, 8)$ sont donc difficiles par rapport aux problèmes pour lesquels $(F, C) = (0.01, 1)$; il y a aussi toutes les combinaisons de difficultés "intermédiaires". Les problèmes de la classe R permettent d'évaluer avec davantage de précision que ceux de la classe C la sensibilité d'une méthode aux caractéristiques des problèmes.

Le tableau 3.1 présente les 196 problèmes utilisés dans le cadre de ce mémoire, classés selon leurs dimensions (représentées par un triplet $(|N|, |A|, |K|)$). Le nombre de problèmes est donné entre parenthèses et les problèmes non réalisables ont été supprimés.

Classe C-1 (31)	Classe C-2 (12)	Classe R-1 (72)	Classe R-2 (81)
20,230,40 (3)	25,100,10 (3)	10,35,10 (6)	20,120,40 (9)
20,230,200 (4)	25,100,30 (3)	10,35,25 (6)	20,120,100 (9)
20,300,40 (4)	100,400,10 (3)	10,35,50 (6)	20,120,200 (9)
20,300,200 (4)	100,400,30 (3)	10,60,10 (9)	20,220,40 (9)
30,520,100 (4)		10,60,25 (9)	20,220,100 (9)
30,520,400 (4)		10,60,50 (9)	20,220,200 (9)
30,700,100 (4)		10,85,10 (9)	20,320,40 (9)
30,700,400 (4)		10,85,25 (9)	20,320,100 (9)
		10,85,50 (9)	20,320,200 (9)

TAB. 3.1 – Classification des problèmes quant à leur dimension.

3.1.3 Plan d'expérimentation

Nous avons choisi, parmi les 196 problèmes, un sous-ensemble de 44 problèmes de difficultés variables et présentant des caractéristiques différentes afin d'effectuer des tests préliminaires. Les 44 problèmes choisis sont présentés dans les tableaux 3.2 et 3.3.

Une première série de tests préliminaires nous a permis de déterminer des valeurs prometteuses pour les paramètres de la méthode de décomposition par montée duale. Ensuite, une deuxième série de tests préliminaires nous a permis d'analyser les différentes versions de la méthode des faisceaux et de l'approche primale-duale. Finalement, nous avons comparé les meilleures versions de notre approche de montée duale avec CPLEX [5] et avec la méthode des faisceaux de Frangioni [11] sur l'ensemble des 196 problèmes.

Classe C-1	(8)	Classe C-2	(5)
20_230_200_F_L	(1)	c100_400_10_F_T	(1)
20_300_200_V_L	(1)	c100_400_30_F_L	(1)
30_520_100_V_T	(1)	c100_400_30_V_T	(1)
30_520_400_V_T	(1)	c25_100_10_V_L	(1)
30_520_400_F_T	(1)	c25_100_30_F_T	(1)
30_700_400_V_L	(1)		
30_700_400_F_L	(1)		
30_700_400_V_T	(1)		

TAB. 3.2 – Problèmes de la classe C choisis pour les tests préliminaires.

Classe R-1	(1)	Classe R-2	(30)
r10_35_50_1.0_0.01	(1)	r20_120_200_8.0_0.05	(1)
		r20_220_100_2.0_0.10	(1)
		r20_220_200_1.0_0.10	(1)
		r20_320_40	(9)
		r20_320_100	(9)
		r20_320_200	(9)

TAB. 3.3 – Problèmes de la classe R choisis pour les tests préliminaires.

3.1.4 Mesures de performance

Cinq mesures de performance ont été utilisées pour comparer les résultats obtenus par les différentes méthodes de résolution du PCMC. Ces mesures apparaîtront dans les tableaux de résultats des prochaines sections :

- le nombre d’itérations : IT ;
- le temps CPU (en secondes) : T ;
- le nombre de fois qu’une méthode obtient la meilleure borne inférieure : N^* ;
- le pourcentage d’écart entre la borne inférieure obtenue Z et la meilleure borne

inférieure connue Z_* , GAP_{lb} :

$$\text{GAP}_{lb} = \frac{Z_* - Z}{Z_*} 100;$$

- le pourcentage d'écart entre la borne supérieure obtenue Z et la meilleure borne inférieure connue Z_* (dans le cas de l'approche primale-duale seulement), GAP_{ub} :

$$\text{GAP}_{ub} = \frac{Z - Z_*}{Z_*} 100.$$

Notons que pour GAP_{lb} et GAP_{ub} , Z_* correspond à la valeur optimale de la relaxation linéaire du PCMC, sauf pour quelques problèmes difficiles pour lesquels des limites technologiques nous empêchent de calculer celle-ci (voir Crainic, Frangioni et Gendron [7]).

3.2 Calibrage de la méthode de décomposition par montée duale

La première série de tests a pour but de déterminer, à l'aide de 44 problèmes présentant des caractéristiques variées, la meilleure valeur à choisir pour chacun des paramètres de notre méthode de décomposition par montée duale : le paramètre d'initialisation τ ; l'ordre dans lequel les sous-problèmes sont résolus à chaque itération de la montée duale ; l'amélioration minimale de la valeur de l'objectif dual entre deux itérations de la montée duale ; le nombre maximum d'itérations de la montée duale ; le nombre d'itérations t de l'algorithme du simplexe (CPLEX) lors de la résolution du sous-problème de multiflot.

Chacun de ces paramètres prendra, jusqu'à ce qu'une valeur plus prometteuse puisse être identifiée, la valeur par défaut suivante :

- $\tau = 1.00$;
- ordre des sous-problèmes : conception de réseaux sans capacité \rightarrow sac à dos continu \rightarrow multiflot ;
- amélioration minimale de l'objectif entre deux itérations de la montée duale : inactive (c'est-à-dire que ce critère d'arrêt de la méthode de décomposition par montée duale n'est pas considéré) ;

- nombre maximum d’itérations de la montée duale : 100 ;
- nombre d’itérations de CPLEX dans le sous-problème de multiflot : 100.

3.2.1 Paramètre d’initialisation τ

Le paramètre d’initialisation τ doit être inclus dans l’intervalle $[0, 1]$ (voir section 2.2.1). Si $\tau = 0$, alors toutes les variables duales, de même que l’objectif dual, ont des valeurs initiales nulles tandis que si $\tau = 1$, on obtient alors une formulation équivalente à la relaxation linéaire de la formulation faible du PCMC.

Nous avons testé les valeurs suivantes de τ pour chacun des 44 problèmes : 0.00, 0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95 et 1.00. Le paramètre τ est le seul paramètre qui est modifié d’un test à l’autre, les autres paramètres prenant leurs valeurs par défaut.

Le tableau 3.4 présente, pour chacune des valeurs de τ testées, la moyenne des résultats obtenus par la méthode de décomposition par montée duale sur l’ensemble des 44 problèmes choisis. Chaque case donne, pour une valeur de τ fixée, les valeurs de GAP_{lb} et de N^* sur la première ligne, ainsi que les valeurs de T et IT sur la deuxième ligne.

τ	GAP_{lb}, N^* (T, IT)	τ	GAP_{lb}, N^* (T, IT)	τ	GAP_{lb}, N^* (T, IT)
0.00	22.48 %, 2 (232.93 s., 41.36)	0.25	20.55 %, 1 (224.44 s., 38.20)	0.90	17.88 %, 0 (144.18 s., 25.97)
0.05	21.63 %, 0 (235.59 s., 39.63)	0.50	19.41 %, 4 (154.48 s., 33.40)	0.95	17.55 %, 5 (91.10 s., 23.52)
0.10	21.11 %, 0 (216.86 s., 36.70)	0.75	18.71 %, 0 (217.14 s., 31.04)	1.00	17.05 %, 32 (65.85 s., 18.61)

TAB. 3.4 – Choix du paramètre d’initialisation τ .

On remarque que lorsque $\tau = 1$, T et GAP_{lb} prennent les plus petites valeurs. De

plus, nous avons observé qu'avec $\tau = 1$, on obtient la meilleure borne 32 fois sur 44. Enfin, lorsque $\tau = 1$, l'écart est supérieur de plus de 2.1% à celui obtenu avec une autre valeur de τ qu'une seule fois. Ainsi, même lorsque les résultats obtenus avec $\tau = 1$ ne sont pas les meilleurs, ils sont tout de même comparables à ceux obtenus avec la meilleure valeur de τ . Nous garderons donc le paramètre d'initialisation τ à 1 pour les autres tests.

3.2.2 Ordre des sous-problèmes

Après avoir choisi de fixer le paramètre d'initialisation τ à 1, nous avons effectué une deuxième série de tests afin de déterminer le meilleur ordre pour la résolution des sous-problèmes dans l'heuristique de décomposition par montée duale. Nous avons donc testé pour chacun des 44 problèmes les six ordres possibles pour les trois sous-problèmes décrits à la section 2.2. En raison de la rapidité de résolution du sous-problème de sac à dos continu (KS), nous avons aussi tenté de résoudre quatre sous-problèmes à chaque itération de la méthode de décomposition par montée duale en intercalant le sous-problème KS entre les sous-problèmes de conception de réseaux sans capacité (UNDP) et de multiflot (MF). Ceci engendre deux combinaisons supplémentaires, selon qu'on choisisse de résoudre le sous-problème de conception de réseaux sans capacité avant ou après le sous-problème de multiflot. Pour tous ces tests, les autres paramètres sont demeurés fixés à leurs valeurs par défaut.

Le tableau 3.5 présente, dans le même format que le tableau 3.4, les résultats moyens obtenus pour chacun des ordres testés.

Ordre	GAP _{lb} , N* (T, IT)
UNDP → KS → MF	17.05 %, 1 (65.75 s., 18.61)
<i>suite à la page suivante</i>	

UNDP \rightarrow MF \rightarrow KS	16.80 %, 8 (79.02 s., 18.25)
MF \rightarrow UNDP \rightarrow KS	15.65 %, 24 (85.41 s., 23.65)
MF \rightarrow KS \rightarrow UNDP	18.33 %, 6 (266.15 s., 42.31)
KS \rightarrow UNDP \rightarrow MF	21.79 %, 0 (239.62 s., 40.22)
KS \rightarrow MF \rightarrow UNDP	20.56 %, 0 (195.08 s., 41.40)
UNDP \rightarrow KS \rightarrow MF \rightarrow KS	17.13 %, 0 (73.78 s., 19.88)
MF \rightarrow KS \rightarrow UNDP \rightarrow KS	18.72 %, 10 (270.92 s., 44.79)

TAB. 3.5 – Choix de l'ordre des sous-problèmes.

Dans un premier temps, notons que nous pouvons éliminer les quatre ordres suivants puisqu'ils obtiennent les quatre plus grandes valeurs de GAP_{lb} , ainsi que les temps T les plus élevés : (KS \rightarrow UNDP \rightarrow MF), (KS \rightarrow MF \rightarrow UNDP), (MF \rightarrow KS \rightarrow UNDP) et (MF \rightarrow KS \rightarrow UNDP \rightarrow KS). Nous pouvons aussi éliminer l'ordre (UNDP \rightarrow KS \rightarrow MF \rightarrow KS) puisque dans ce cas, nous obtenons sensiblement les mêmes résultats en enlevant le dernier sous-problème de sac à dos continu.

Lorsque nous utilisons l'ordre (MF \rightarrow UNDP \rightarrow KS), nous obtenons les meilleurs écarts en moyenne avec un temps moyen très compétitif. De plus, nous avons observé que cet ordre permet d'obtenir la meilleure borne inférieure 24 fois sur 44. Enfin, en utilisant cet ordre de sous-problème, l'écart est supérieur de plus de 2.25% à celui obtenu avec un autre ordre qu'à deux reprises seulement. Ainsi, même lorsque les résultats obtenus avec cet ordre ne sont pas les meilleurs, ils sont tout de même comparables à ceux obtenus avec le meilleur ordre.

Les deux ordres débutant par un sous-problème de conception de réseaux sans capacité permettent d'obtenir les meilleurs temps moyens et de bons écarts moyens. En particulier, nous obtenons avec ces deux ordres des résultats équivalents à ceux obtenus avec l'ordre (MF→UNDP→KS) sur les problèmes de la série R ayant des contraintes de capacités très restrictives (paramètre $C = 8$, voir section 3.1.2), obtenant des résultats parfois légèrement meilleurs qu'avec (MF→UNDP→KS), parfois légèrement moins bons. Par contre, avec les deux ordres débutant par un sous-problème de conception de réseaux sans capacité, nous obtenons plus souvent qu'avec l'ordre (MF→UNDP→KS) un écart de plus de 5% supérieur à celui obtenu avec le meilleur ordre. Par conséquent, nous garderons l'ordre de sous-problèmes (MF→UNDP→KS) pour les autres tests.

3.2.3 Nombre d'itérations de la montée duale et du sous-problème de multiflot

La troisième série de tests a pour but de déterminer les meilleures valeurs pour le nombre maximum d'itérations de la méthode de décomposition par montée duale (DAiter) et pour le nombre d'itérations du simplexe dans le sous-problème de multiflot (MFiter). Nous testons ces deux paramètres ensemble puisqu'ils sont reliés. En effet, si nous permettons au sous-problème de multiflot d'effectuer un grand nombre d'itérations du simplexe afin de trouver une solution duale de meilleure qualité, alors le nombre d'itérations de la méthode de décomposition par montée duale diminuera ; à l'opposé, si nous restreignons le sous-problème de multiflot à effectuer un petit nombre d'itérations du simplexe, alors nous passerons moins de temps dans ce sous-problème, mais le nombre d'itérations de la méthode de décomposition par montée duale requis pour trouver une bonne borne inférieure augmentera.

Nous avons adopté les deux stratégies suivantes afin de déterminer des valeurs prometteuses pour les paramètres DAiter et MFiter :

- stratégie A : donner une grande valeur à MFiter (100, 300, 500, 750, 1000 et 1500) et une petite valeur à DAiter (1, 2, 3, 5, 10 ou 25) ;

- stratégie B : donner une petite valeur à MFiter (10, 20 ou 50) et une grande valeur à DAiter (25, 50, 75, 100 ou 200).

Les autres paramètres demeurent fixés à leurs valeurs par défaut, sauf l'ordre des sous-problèmes qui prend la valeur la plus prometteuse identifiée dans la série de tests précédente : multiflot → conception de réseaux sans capacité → sac à dos continu.

Le tableau 3.6 présente, pour chacune des combinaisons (DAiter,MFiter) de la stratégie A, les résultats moyens obtenus sur les 44 problèmes choisis. Chaque case donne, pour une combinaison déterminée, la valeur de GAP_{lb} sur la première ligne, ainsi que les valeurs de T et IT sur la deuxième ligne.

(DAiter,MFiter)	GAP_{lb} (T, IT)	(DAiter,MFiter)	GAP_{lb} (T, IT)
(1, 100)	27.76 % (3.07 s., 1)	(1, 300)	26.80 % (3.10 s., 1)
(2, 100)	18.04 % (7.38 s., 2)	(2, 300)	16.05 % (7.80 s., 2)
(3, 100)	17.17 % (11.35 s., 3)	(3, 300)	15.14 % (11.97 s., 3)
(5, 100)	16.86 % (19.25 s., 4.95)	(5, 300)	14.76 % (20.07 s., 4.97)
(10, 100)	16.46 % (34.59 s., 8.40)	(10, 300)	14.54 % (34.37 s., 8.13)
(25, 100)	16.03 % (64.74 s., 15.25)	(25, 300)	14.30 % (53.72 s., 13.40)
(1, 500)	26.46 % (3.26 s., 1)	(1, 750)	26.00 % (3.79 s., 1)
(2, 500)	15.48 % (8.00 s., 2)	(2, 750)	14.79 % (9.20 s., 2)
<i>suite à la page suivante</i>			

(3, 500)	14.43 % (12.29 s., 3)	(3, 750)	13.74 % (14.02 s., 3)
(5, 500)	14.07 % (20.83 s., 5)	(5, 750)	13.34 % (22.84 s., 5)
(10, 500)	13.73 % (33.42 s., 8.06)	(10, 750)	13.10 % (37.86 s., 8.31)
(25, 500)	13.56 % (46.84 s., 12.56)	(25, 750)	13.10 % (60.72 s., 14.29)
(1, 1000)	25.70 % (3.87 s., 1)	(1, 1500)	25.29 % (4.42 s., 1)
(2, 1000)	14.43 % (9.76 s., 2)	(2, 1500)	14.03 % (11.01 s., 2)
(3, 1000)	13.38 % (14.61 s., 3)	(3, 1500)	13.00 % (15.87 s., 3)
(5, 1000)	13.05 % (23.06 s., 5)	(5, 1500)	12.70 % (24.14 s., 5)
(10, 1000)	12.92 % (39.52 s., 8.11)	(10, 1500)	12.69 % (38.31 s., 7.97)
(25, 1000)	12.92 % (56.44 s., 12.63)	(25, 1500)	12.69 % (51.47 s., 11.97)

TAB. 3.6 – Choix des paramètres DAiter et MFiter (stratégie A).

Les combinaisons où DAiter est égal à 2 semblent donner les meilleurs résultats. En effet, lorsque DAiter est égal à 1, les écarts moyens (GAP_{lb}) sont très élevés et lorsque DAiter est supérieur à 2, les temps moyens augmentent considérablement, mais les écarts moyens ne diminuent pas beaucoup. Quant à MFiter, il semble que la valeur 500 permette d'obtenir un bon compromis entre le temps moyen et l'écart moyen. En effet, lorsque MFiter prend une valeur inférieure à 500, l'écart GAP_{lb} augmente sensiblement (de 15.5% lorsque MFiter est à 500 à 18.0% lorsque MFiter est à 100) tandis que le temps T est à peu près équivalent (8.0 secondes lorsque MFiter est à 500, comparativement à 7.4 secondes lorsque MFiter est à 100). De plus, lorsque MFiter prend une valeur supérieure à 500, l'écart GAP_{lb} diminue (de 15.5% lorsque MFiter est à 500, il passe à 14.8% lorsque MFiter est à 750 et à 14% lorsque MFiter est à 1500), mais le temps T augmente considérablement (de 8.0 secondes lorsque MFiter est à 500, il passe à 9.2 secondes lorsque MFiter est à 750 et à 11.0 secondes lorsque MFiter est à 1500).

Le tableau 3.7 présente, dans le même format que le tableau 3.6, les résultats obtenus pour chacune des combinaisons (DAiter,MFiter) de la stratégie B.

(DAiter,MFiter)	GAP_{lb} (T, IT)	(DAiter,MFiter)	GAP_{lb} (T, IT)
(25, 10)	18.30 % (277.14 s., 17.47)	(100, 20)	17.24 % (322.08 s., 27.06)
(50, 10)	17.95 % (466.71 s., 26.75)	(200, 20)	17.02 % (380.03 s., 32.88)
(75, 10)	17.86 % (594.10 s., 31.52)	(25, 50)	16.99 % (83.75 s., 13.70)
(100, 10)	17.75 % (635.40 s., 34.27)	(50, 50)	16.73 % (102.55 s., 16.88)
(200, 10)	17.53 % (770.69 s., 41.97)	(75, 50)	16.57 % (116.02 s., 19.02)
<i>suite à la page suivante</i>			

(25, 20)	17.74 % (152.95 s., 16.02)	(100, 50)	16.45 % (117.94 s., 20.72)
(50, 20)	17.46 % (254.56 s., 22.18)	(200, 50)	16.30 % (122.23 s., 26)
(75, 20)	17.33 % (302.06 s., 25.18)		

TAB. 3.7 – Choix des paramètres DAiter et MFiter (stratégie B).

Remarquons que les écarts moyens sont tous supérieurs à l'écart moyen obtenu pour le couple (DAiter = 2, MFiter = 500). De plus, les temps moyens sont beaucoup plus élevés que tous ceux obtenus dans le tableau de résultats précédent. Par conséquent, il semble plus avantageux de donner une grande valeur à MFiter et de faire un petit nombre d'itérations de la méthode de décomposition par montée duale que l'inverse. Nous choisissons donc les valeurs suivantes pour les autres tests : (DAiter = 2, MFiter = 500).

3.2.4 Amélioration minimale de l'objectif entre deux itérations de la montée duale

Dans la quatrième série de tests, nous évaluons l'efficacité et la pertinence du critère d'arrêt " Z_d amélioration" de la méthode de décomposition par montée duale qui demande un certain pourcentage d'amélioration de la valeur de l'objectif dual par rapport à l'itération précédente pour poursuivre. Nous avons testé les valeurs suivantes pour " Z_d amélioration" : inactive (c'est-à-dire que ce critère d'arrêt n'est pas considéré), 0.001%, 0.01%, 0.1%, 1%, 5%, 10%, 15%, 20% et 25%. Pour tous ces tests, les autres paramètres demeurent fixés à leurs valeurs par défaut, sauf DAiter, MFiter et l'ordre des sous-problèmes qui prennent les valeurs les plus prometteuses identifiées dans les séries de tests précédentes : 2, 500 et multiflot → conception de réseaux sans capacité → sac à dos continu.

Le tableau 3.8 présente, dans le même format que le tableau 3.7, les résultats obtenus

pour chacune des valeurs testées.

Z_d amélioration	GAP_{lb} (T, IT)	Z_d amélioration	GAP_{lb} (T, IT)
inactive	15.48 % (7.95 s., 2)	5 %	15.48 % (7.97 s., 2)
0.001 %	15.48 % (7.95 s., 2)	10 %	15.48 % (7.96 s., 2)
0.01 %	15.48 % (7.93 s., 2)	15 %	15.48 % (7.96 s., 2)
0.1 %	15.48 % (7.94 s., 2)	20 %	15.48 % (7.95 s., 2)
1 %	15.48 % (7.95 s., 2)	25 %	15.48 % (7.96 s., 2)

TAB. 3.8 – Choix de l'amélioration minimale de l'objectif entre deux itérations de la montée duale (première partie).

Nous constatons que ce critère d'arrêt est inutile puisque le nombre maximum d'itérations est toujours atteint. Nous avons donc effectué une autre série de tests en fixant le nombre maximum d'itérations de la méthode de décomposition par montée duale à 3, tout en faisant varier le nombre d'itérations du simplexe dans le sous-problème de multiflot et le pourcentage d'amélioration de l'objectif. Comme nous l'avons vu à la section 3.2.3, effectuer une troisième itération de la méthode de décomposition par montée duale n'améliore pas suffisamment l'écart moyen GAP_{lb} pour justifier, pour tous les problèmes, le temps supplémentaire requis pour la compléter. Par contre, nous espérons que le nouveau critère d'arrêt développé permettra d'effectuer une troisième itération seulement lorsque ce sera bénéfique.

Le tableau 3.9 présente les résultats moyens obtenus pour chaque couple (Z_d amélioration, MFiter) sur les 44 problèmes choisis. Nous présentons les résultats des tests effectués avec les trois valeurs qui semblaient les plus prometteuses pour le nombre

d'itérations du simplexe dans le sous-problème de multiflot à la section 3.2.3 (500, 750 et 1000), ainsi qu'avec les pourcentages d'amélioration de l'objectif qui ont permis de restreindre le nombre moyen d'itérations de la méthode de décomposition par montée duale à une valeur supérieure à 2 et inférieure à 3 (5%, 10%, 15%, 20% et 25%).

Z_d amélioration	MFiter		
	500	750	1000
5 %	14.48 % (9.69 s., 2.79)	13.80 % (10.93 s., 2.77)	13.46 % (11.49 s., 2.77)
10 %	14.67 % (9.01 s., 2.56)	13.95 % (10.45 s., 2.61)	13.56 % (11.08 s., 2.61)
15 %	15.14 % (8.52 s., 2.29)	14.40 % (9.76 s., 2.31)	13.86 % (10.43 s., 2.34)
20 %	15.32 % (8.07 s., 2.15)	14.63 % (9.25 s., 2.15)	14.27 % (9.84 s., 2.15)
25 %	15.34 % (8.06 s., 2.13)	14.65 % (9.27 s., 2.13)	14.29 % (9.83 s., 2.13)

TAB. 3.9 – Choix de l'amélioration minimale de l'objectif entre deux itérations de la montée duale (deuxième partie).

La paire (DAiter, MFiter) la plus prometteuse identifiée à la section 3.2.3, (2, 500), permettait d'obtenir un écart moyen de 15.48% en un temps moyen de 8.00 secondes. Si nous privilégions le temps par rapport à l'écart GAP_{lb} , alors cette configuration de paramètres demeure un choix prometteur. Par contre, si nous recherchons un bon compromis entre le temps et l'écart GAP_{lb} , alors nous pourrions opter pour les paramètres suivants : DAiter = 3, MFiter = 1000 et Z_d amélioration = 10%. En effet, avec cette configuration de paramètres, l'écart moyen diminue d'environ 2% par rapport à (DAiter = 2, MFiter = 500, Z_d amélioration : inactive), tandis que le temps moyen augmente d'environ trois secondes.

3.2.5 Résultats de la phase de calibrage

Les quatre premières séries de tests nous ont permis d'identifier deux configurations de paramètres prometteuses. La configuration P_1 met l'accent sur la rapidité de la méthode et les paramètres prennent les valeurs suivantes :

- $\tau = 1.00$;
- ordre des sous-problèmes : multiflot \rightarrow conception de réseaux sans capacité \rightarrow sac à dos continu ;
- nombre maximum d'itérations de la montée duale : 2 ;
- nombre d'itérations de CPLEX dans le sous-problème de multiflot : 500 ;
- amélioration minimale de l'objectif entre deux itérations de la montée duale : inactive.

La configuration de paramètres P_2 tente de trouver un bon compromis entre la valeur de la borne inférieure et la rapidité de la méthode. Les paramètres prennent alors les valeurs suivantes :

- $\tau = 1.00$;
- ordre des sous-problèmes : multiflot \rightarrow conception de réseaux sans capacité \rightarrow sac à dos continu ;
- nombre maximum d'itérations de la montée duale : 3 ;
- nombre d'itérations de CPLEX dans le sous-problème de multiflot : 1000 ;
- amélioration minimale de l'objectif entre deux itérations de la montée duale : 0.10.

3.3 Approche primale-duale

Rappelons que l'approche primale-duale permet de résoudre un sous-problème de multiflot afin de trouver une solution primale réalisable du PCMC. À l'aide de cette solution primale (qui nous permet de calculer une borne supérieure du PCMC), nous réinitialisons les variables duales de manière à satisfaire les conditions des écarts complémentaires le plus possible, avant de redémarrer la méthode de décomposition par

montée duale. Si la méthode de décomposition par montée duale réussit à trouver une meilleure borne inférieure grâce à l'approche primale-duale, le processus se répète ; sinon, l'exécution de la méthode de résolution se termine.

Nous avons testé quatre versions de l'approche primale-duale qui dépendent de deux choix : résoudre l'approche primale-duale avec primal restreint (RPD) ou avec primal non restreint (UPD), et garder certaines variables duales fixées ou non lorsque nous redémarrons la méthode de décomposition par montée duale.

Le tableau 3.10 présente, pour chacune des configurations de paramètres prometteuses P_1 et P_2 identifiées précédemment, la moyenne des résultats obtenus par les quatre versions de la méthode de résolution combinant l'heuristique de décomposition par montée duale et l'approche primale-duale sur l'ensemble des 44 problèmes choisis. Chaque case donne, pour une version de la méthode de résolution et une configuration de paramètres de la méthode de décomposition par montée duale, GAP_{ub} sur la première ligne, GAP_{lb} sur la deuxième ligne, ainsi que le temps moyen et le nombre moyen d'itérations sur la troisième ligne. Si la méthode de résolution s'arrête après une itération, cela indique que la méthode de décomposition par montée duale n'a pas réussi à améliorer sa borne inférieure grâce à l'approche primale-duale. Si la méthode de résolution s'arrête après t itérations, $t > 1$, cela indique que la méthode de décomposition par montée duale a réussi à améliorer sa borne inférieure à $t - 1$ reprise(s) grâce à l'approche primale-duale.

Remarquons tout d'abord que l'écart moyen GAP_{ub} est très élevé pour les deux versions de l'approche primale-duale avec primal non restreint. Ceci s'explique par le fait que les bornes supérieures calculées ici forment des solutions primales réalisables du PCMC (c'est-à-dire avec variables de conception y_{ij} entières), alors que l'écart est calculé avec la solution optimale de la relaxation linéaire du PCMC (qui constitue donc elle-même une borne inférieure de la solution optimale du PCMC). De plus, comme l'objectif de ce mémoire consiste à trouver de bonnes bornes inférieures à la relaxation linéaire du PCMC, nous avons concentré nos efforts sur cette tâche plutôt qu'à trouver de bonnes bornes supérieures. Néanmoins, remarquons que lorsque le sous-problème

	UPD libre	UPD fixé	RPD libre	RPD fixé
	GAP_{ub}	GAP_{ub}	GAP_{ub}	GAP_{ub}
	GAP_{lb}	GAP_{lb}	GAP_{lb}	GAP_{lb}
	(T, IT)	(T, IT)	(T, IT)	(T, IT)
Moyenne	701.89 %	704.20 %	20.87 %	20.87 %
pour	14.53 %	15.17 %	15.44 %	15.46 %
P_1	(23.07 s., 1.36)	(23.23 s., 1.15)	(80.61 s., 1.02)	(83.18 s., 1.02)
Moyenne	417.94 %	423.75 %	20.49 %	20.67 %
pour	12.45 %	13.23 %	13.40 %	13.31 %
P_2	(32.95 s., 1.50)	(28.78 s., 1.15)	(266.89 s., 1.13)	(267.15 s., 1.06)

TAB. 3.10 – Choix de l’approche primale-duale.

de multiflot est réalisable, les deux versions de l’approche primale-duale avec primal restreint obtiennent des bornes supérieures respectables (l’écart GAP_{ub} est de l’ordre de 20%). Par contre, le sous-problème de multiflot s’est avéré non réalisable dans la majorité des cas, soit 37 fois sur 44 pour la configuration de paramètres P_1 et 32 fois sur 44 pour la configuration de paramètres P_2 . Notons finalement que les solutions réalisables identifiées par l’approche primale-duale peuvent servir à initialiser d’autres méthodes, comme celles proposées dans Ghamlouch, Crainic et Gendreau [16], Crainic, Gendreau et Farvolden [8] et Hernu [20].

L’approche primale-duale avec primal non restreint et variables libres (UPD libre) a obtenu les meilleurs écarts moyens GAP_{lb} et a permis de redémarrer la méthode de décomposition par montée duale plus souvent que les trois autres versions de l’approche primale-duale (puisque’elle a obtenu les nombres moyens d’itérations les plus élevés). De plus, cette version de l’approche primale-duale a obtenu la meilleure borne inférieure des quatre versions pour tous les problèmes, sauf un. Pour P_1 comme pour P_2 , l’approche primale-duale avec primal non restreint et variables libres s’est donc clairement établie comme la plus prometteuse des quatre versions de l’approche primale-duale. Pour les deux configurations de paramètres testées, la méthode de résolution combinant l’ap-

proche primale-duale “UPD libre” et la méthode de décomposition par montée duale a permis de diminuer l’écart moyen GAP_{lb} d’environ 1% en multipliant le temps moyen d’exécution par un facteur de trois par rapport aux résultats obtenus sans l’approche primale-duale.

3.4 Méthode des faisceaux

La méthode des faisceaux de Frangioni [11] est la deuxième approche utilisée pour redémarrer l’heuristique de décomposition par montée duale. Rappelons que la méthode des faisceaux peut être utilisée pour résoudre les duaux lagrangiens issus soit de la relaxation lagrangienne des contraintes de conservation de flot du PCMC (KB), soit de la relaxation lagrangienne des contraintes de capacité mutuelle et individuelle du PCMC (SB). Dans les deux cas, nous pouvons alterner entre la méthode des faisceaux et l’heuristique de décomposition par montée duale jusqu’à ce que la valeur de l’objectif du dual n’augmente plus, ou bien appeler les deux méthodes une seule fois en utilisant l’heuristique de montée duale pour amorcer rapidement la méthode des faisceaux. Nous avons donc testé quatre versions de la combinaison de ces deux méthodes pour chacune des valeurs suivantes du nombre maximum d’itérations de la méthode des faisceaux : 25, 50, 100, 250, 500.

Les tableaux 3.11 et 3.12 présentent, pour chacune des configurations prometteuses de paramètres P_1 et P_2 identifiées précédemment, la moyenne des résultats obtenus par les quatre versions de la méthode de résolution combinant l’heuristique de décomposition par montée duale et la méthode des faisceaux sur l’ensemble des 44 problèmes choisis. Chaque case de ces deux tableaux donne, pour un nombre maximum déterminé d’itérations de la méthode des faisceaux, l’écart GAP_{lb} obtenu par une des quatre versions de la méthode de résolution sur la première ligne. Sur la deuxième ligne, nous retrouvons le temps moyen et le nombre moyen d’itérations requis pour exécuter la méthode de résolution. Lorsque la méthode de décomposition par montée duale est utilisée comme départ rapide pour la méthode des faisceaux, le nombre d’itérations de la méthode combinée de résolution sera toujours 1 puisque les deux méthodes ne sont

appelées qu'une seule fois chacune. Lorsque nous alternons entre les deux méthodes, un nombre t d'itérations signifie que la borne supérieure trouvée initialement par la méthode de décomposition par montée duale a été améliorée à $t - 1$ reprise(s) par l'une ou l'autre des deux méthodes combinées.

Nombre d'itérations	DA \Rightarrow KB	DA \Leftrightarrow KB	DA \Rightarrow SB	DA \Leftrightarrow SB
	GAP _{lb} (T, IT)	GAP _{lb} (T, IT)	GAP _{lb} (T, IT)	GAP _{lb} (T, IT)
25	8.47 % (8.98 s., 1)	2.78 % (42.20 s., 6.25)	12.25 % (11.53 s., 1)	2.64 % (128.81 s., 11.77)
50	4.88 % (10.04 s., 1)	0.87 % (60.65 s., 8.18)	9.21 % (15.26 s., 1)	1.54 % (218.99 s., 12.00)
100	2.64 % (12.03 s., 1)	0.38 % (78.15 s., 7.56)	6.84 % (23.44 s., 1)	0.72 % (360.69 s., 12.97)
250	1.08 % (20.28 s., 1)	0.12 % (254.22 s., 8.38)	3.84 % (52.14 s., 1)	0.17 % (901.16 s., 13.27)
500	0.54 % (39.08 s., 1)	0.02 % (738.33 s., 8.65)	2.02 % (103.81 s., 1)	0.06 % (2184.39 s., 15.31)

TAB. 3.11 – Choix de la combinaison de la méthode des faisceaux et de la méthode de décomposition par montée duale (configuration P_1).

Remarquons que, peu importe la configuration de paramètres utilisée pour l'heuristique de décomposition par montée duale, il est préférable de résoudre, à l'aide de la méthode des faisceaux, le dual lagrangien issu de la relaxation des contraintes de conservation de flot (KB) plutôt que le dual lagrangien issu de la relaxation des contraintes de capacité (SB). En effet, si nous utilisons SB, alors nous obtenons des écarts moyens équivalents ou supérieurs aux écarts moyens obtenus avec KB, peu importe la combinaison de méthodes utilisée. De plus, les temps moyens obtenus par SB sont toujours plus élevés que les temps moyens obtenus avec la version équivalente de la méthode de résolution utilisant KB. Ayant écarté "DA \Rightarrow SB" et "DA \Leftrightarrow SB", remarquons que nous pouvons également éliminer "DA \Leftrightarrow KB" puisque les temps moyens de résolution de

Nombre d'itérations	DA \Rightarrow KB	DA \Leftrightarrow KB	DA \Rightarrow SB	DA \Leftrightarrow SB
	GAP _{lb} (T, IT)	GAP _{lb} (T, IT)	GAP _{lb} (T, IT)	GAP _{lb} (T, IT)
25	7.79 % (12.14 s., 1)	3.25 % (34.71 s., 7.04)	11.74 % (14.84 s., 1)	2.53 % (104.48 s., 11.61)
50	4.48 % (13.16 s., 1)	1.55 % (51.16 s., 7.54)	8.55 % (18.35 s., 1)	1.46 % (176.42 s., 12.93)
100	2.37 % (15.20 s., 1)	0.47 % (88.61 s., 8.75)	5.80 % (26.48 s., 1)	0.64 % (364.92 s., 13.22)
250	0.94 % (23.41 s., 1)	0.11 % (263.77 s., 8.25)	2.93 % (54.64 s., 1)	0.19 % (812.32 s., 14.68)
500	0.47 % (41.69 s., 1)	0.03 % (855.40 s., 9.47)	1.66 % (105.08 s., 1)	0.05 % (1974.45 s., 15.31)

TAB. 3.12 – Choix de la combinaison de la méthode des faisceaux et de la méthode de décomposition par montée duale (configuration P_2).

cette version sont beaucoup trop élevés par rapport à “DA \Rightarrow KB” ; la version “DA \Rightarrow KB” de la méthode combinée s’est donc établie comme la plus prometteuse des quatre versions testées.

Avec la version “DA \Rightarrow KB” de la méthode de résolution combinée, il semble préférable d’utiliser la configuration de paramètres P_1 plutôt que P_2 pour l’heuristique de décomposition par montée duale. En effet, lorsque nous utilisons la configuration P_2 plutôt que P_1 , les écarts moyens diminuent d’au plus 0.70%, tandis que les temps moyens augmentent d’environ trois secondes. Il est donc avantageux de privilégier le temps plutôt qu’un si mince gain pour l’écart moyen. Finalement, un nombre maximum d’itérations de la méthode des faisceaux de 100 offre un bon compromis entre le temps moyen et l’écart moyen pour la version “DA \Rightarrow KB” utilisée avec la configuration de paramètres P_1 ; nous allons donc opter pour cette valeur.

3.5 Comparaison des méthodes retenues avec CPLEX et la méthode des faisceaux

L'ensemble des tests effectués dans les sections 3.2, 3.3 et 3.4 nous a permis d'analyser les différentes versions des trois méthodes développées dans le cadre de ce mémoire pour trouver une borne inférieure au PCMC : l'heuristique de décomposition par montée duale, la combinaison de l'approche primale-duale et de l'heuristique de décomposition par montée duale, la combinaison de la méthode des faisceaux et de l'heuristique de décomposition par montée duale. Nous avons retenu la version de chacune de ces trois méthodes qui nous semblait la plus prometteuse afin d'effectuer une série de tests sur l'ensemble des 196 problèmes décrits à la section 3.1.2.

La première méthode retenue, DA (P_1), est l'heuristique de décomposition par montée duale qui met l'accent sur la rapidité (c'est-à-dire avec la configuration de paramètres P_1 , voir section 3.2.5). La deuxième méthode retenue, UPD libre, est la version la plus prometteuse de la combinaison de l'approche primale-duale et de l'heuristique de décomposition par montée duale, soit l'approche primale-duale avec primal non restreint et variables duales libres lors du redémarrage de la montée duale (avec la configuration de paramètres P_1). La méthode de résolution qui utilise l'heuristique de décomposition par montée duale (avec la configuration de paramètres P_1 également) pour amorcer rapidement la méthode des faisceaux (qui résout le dual lagrangien issu de la relaxation des contraintes de conservation de flot avec un nombre maximum d'itérations de 100) constitue la troisième méthode retenue : DA \Rightarrow KB.

Afin d'évaluer l'efficacité des trois méthodes retenues, nous comparons les résultats de celles-ci avec ceux obtenus par l'algorithme dual du simplexe de CPLEX (temps maximum : 10 heures) et par la méthode des faisceaux (KB) utilisée sans l'heuristique de décomposition par montée duale pour résoudre le dual lagrangien issu de la relaxation des contraintes de conservation de flot du PCMC avec un nombre maximum d'itérations de 100.

3.5.1 Analyse des résultats en moyenne

Les résultats détaillés obtenus par les cinq méthodes de résolution du PCMC retenues pour la dernière série de tests sur les 196 problèmes se retrouvent à l'annexe A. Le tableau 3.13 résume ces résultats en présentant la moyenne des écarts GAP_{lb} (par rapport à la borne inférieure obtenue par CPLEX) et des temps CPU de chacune des cinq méthodes sur les 196 problèmes testés.

	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
GAP_{lb}	11.99 %	10.48 %	1.63 %	1.00 %	—
(T)	(3.35 s.)	(11.78 s.)	(5.46 s.)	(2.08 s.)	(1204.03 s.)

TAB. 3.13 – Résultats moyens des méthodes retenues sur les 196 problèmes.

L'algorithme dual du simplexe (CPLEX) a réussi à trouver, en moins de dix heures, la solution optimale de la relaxation linéaire du PCMC pour 195 des 196 problèmes testés (voir annexe A). De plus, notons que pour la majorité des problèmes ayant peu de produits et peu de noeuds (les problèmes c20_230_40, c25_100_10 et r10_35_10, par exemple), CPLEX requiert des temps CPU très bas (moins de deux secondes) ; cette méthode semble donc être relativement efficace pour résoudre des instances de petite taille du PCMC. Par contre, dès que le nombre de produits ou de noeuds augmente (les problèmes c100_400_30, c30_520_100 et r20_120_200, par exemple), les temps requis par CPLEX sont extrêmement élevés, rendant cette méthode (peu adaptée à la structure particulière du PCMC) inefficace pour résoudre des instances de grande taille. Nous remarquons d'ailleurs que le temps moyen obtenu par CPLEX est très élevé (environ 20 minutes, contre quelques secondes pour les quatre autres méthodes). Ceci démontre la nécessité de développer des méthodes spécialisées afin de trouver *rapidement* de bonnes bornes inférieures à la relaxation linéaire des problèmes difficiles du PCMC.

Les résultats obtenus sur les 196 problèmes testés indiquent qu'en moyenne, l'approche primale-duale UPD libre a réussi à améliorer l'écart GAP_{lb} obtenu par la méthode DA (P_1) d'environ 1.5% en multipliant le temps CPU par un facteur d'environ 3.5. La méthode UPD libre a mené à une amélioration de la borne inférieure obtenue

par la méthode DA (P_1) pour 102 des 196 problèmes testés ; sur ces 102 problèmes, l'écart GAP_{lb} s'est amélioré de 2.90% en moyenne grâce à l'approche primale-duale et le temps CPU a été multiplié par un facteur de 4.8. Nous observons que le temps requis pour redémarrer l'heuristique de décomposition par montée duale avec l'approche primale-duale est trop important pour la faible amélioration de la borne inférieure que cela procure. Ce constat n'est guère surprenant puisque l'approche primale-duale remet l'objectif dual à 0 en réinitialisant les variables duales (contrairement à la méthode des faisceaux qui, elle, ne peut diminuer l'objectif du dual obtenu par l'heuristique de décomposition par montée duale) ; l'heuristique de décomposition par montée duale redémarre donc à zéro après chaque exécution de l'approche primale-duale et ce, sans garantir que l'information duale recueillie mène à une meilleure borne. Bien qu'il se soit avéré inefficace dans notre cas, il est intéressant de noter que ce type d'approche est souvent suggéré dans la littérature pour redémarrer des méthodes de montée duale.

La figure 3.1 montre l'évolution typique de l'objectif du dual en fonction du temps pour la méthode UPD libre et indique par une flèche les moments où le contrôle de l'exécution passe de l'heuristique de décomposition par montée duale (DA) à l'approche primale-duale (UPD), et vice-versa. La solution optimale de la relaxation linéaire du PCMC, obtenue par CPLEX, est indiquée par une ligne pointillée. Nous remarquons que, pour ce problème, l'heuristique de décomposition par montée duale a légèrement amélioré sa borne inférieure initiale à 2 reprises, avant de s'arrêter en obtenant une borne inférieure de moindre qualité, après avoir multiplié le temps d'exécution de la méthode de montée duale par 14 !

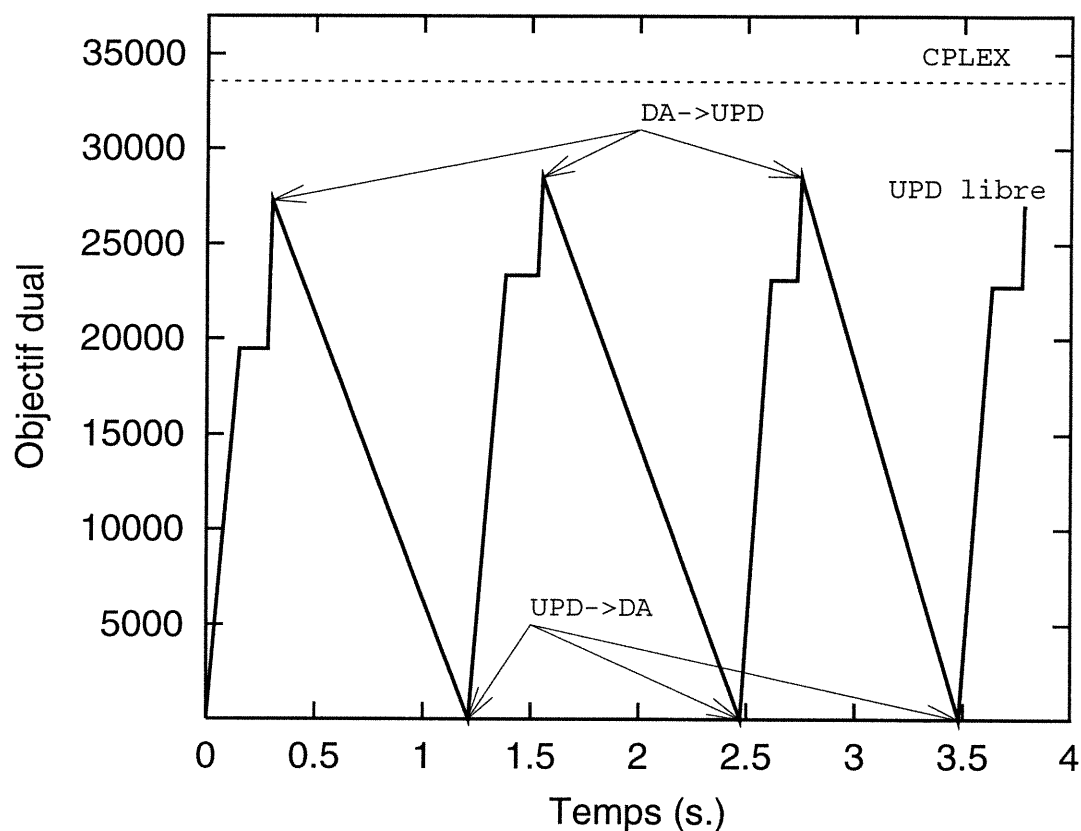


FIG. 3.1 – Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème c25_100_30_F.L résolu avec la méthode UPD libre.

Le tableau 3.13 montre que l'écart moyen obtenu par la méthode DA (P_1), environ 12%, est le plus élevé des cinq méthodes testées. Nous nous attendions à ce résultat puisque les méthodes UPD libre et DA \Rightarrow KB utilisent la méthode DA (P_1) comme point de départ et ne peuvent par conséquent qu'améliorer la borne obtenue par cette méthode; les méthodes KB et CPLEX devraient quant à elles obtenir de meilleures bornes que la méthode DA (P_1) puisqu'elles traitent l'ensemble des variables du PCMC en même temps, contrairement à la méthode de décomposition par montée duale qui, elle, ne traite qu'un sous-ensemble des variables duales du PCMC dans chacun de ses sous-problèmes.

La méthode de résolution DA (P_1) est la deuxième méthode la plus rapide, en

moyenne, après la méthode KB. La figure 3.2 indique que le sous-problème de multiflot (MF) fait perdre beaucoup de temps à la méthode de décomposition par montée duale. En effet, la méthode DA (P_1) passe en moyenne 86% de son temps CPU dans le sous-problème MF, contre 10% et 4% dans les sous-problèmes de conception de réseaux sans capacité (UNDP) et de sac à dos (KS). Nous avons choisi d'utiliser l'algorithme dual

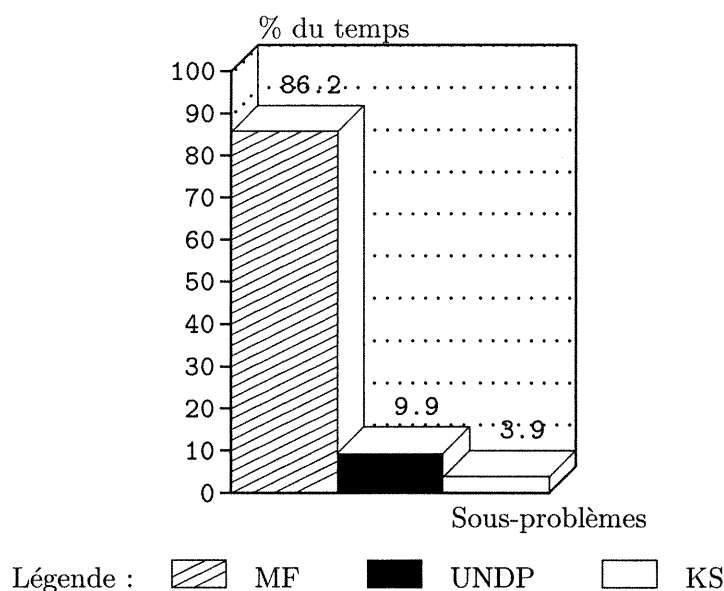


FIG. 3.2 – Proportion du temps pris par chacun des trois sous-problèmes de la méthode de décomposition par montée duale sur les 196 problèmes.

du simplexe (CPLEX) pour résoudre le sous-problème de multiflot puisqu'il s'agit d'une méthode robuste et que nous croyions pouvoir doser le temps requis pour résoudre ce sous-problème en limitant le nombre d'itérations de la méthode de résolution (voir section 2.2.4). Par contre, nous remarquons que, d'après les tableaux de résultats présentés à la section 3.2.3, le nombre d'itérations de l'algorithme du simplexe n'influence pas beaucoup le temps CPU. En effet, sur les 44 problèmes testés à la section 3.2.3, une itération de la méthode de décomposition par montée duale prenait en moyenne 3.07 secondes avec 100 itérations du simplexe dans le sous-problème de multiflot et 3.87 secondes avec 1000 itérations du simplexe, soit moins d'une seconde de différence pour 900 itérations du simplexe de plus ! À la lumière de ces résultats, il semble qu'il serait préférable d'utiliser une méthode plus rapide que l'algorithme du simplexe (la méthode

des faisceaux, par exemple) pour résoudre le sous-problème de multiflot dans l'heuristique de décomposition par montée duale. De cette façon, nous devrions obtenir, avec DA (P_1), la méthode la plus rapide pour obtenir des bornes inférieures du PCMC.

La méthode DA \Rightarrow KB obtient un excellent écart moyen : 1.63%, soit à peu près l'équivalent de l'écart moyen obtenu par la méthode KB (1.00%). Celle-ci obtient parfois une meilleure borne que la méthode DA \Rightarrow KB, parfois une borne de moindre qualité. Rappelons que l'heuristique de décomposition par montée duale permet d'aider à démarrer la méthode KB de deux façons : en donnant des valeurs initiales aux multiplicateurs lagrangiens et en initialisant le faisceau de sous-gradients. Or, comme nous ne faisons que très peu d'itérations de la méthode de décomposition par montée duale, le faisceau ne comptera aussi que très peu de sous-gradients au départ de la méthode KB. De plus, les multiplicateurs lagrangiens initiaux seront modifiés dès la deuxième itération de la méthode des faisceaux. Par conséquent, il semble que la méthode de décomposition par montée duale n'aide la méthode des faisceaux que dans les premières itérations seulement et c'est pourquoi elle ne guide pas nécessairement celle-ci vers une meilleure borne après cent itérations.

Le temps CPU moyen de la méthode DA \Rightarrow KB est, sans surprise, à peu près équivalent aux temps moyens additionnés des deux méthodes combinées pour former cette méthode de résolution. Par conséquent, si nous diminuons le temps moyen de la méthode DA (P_1) en utilisant une méthode plus rapide que l'algorithme du simplexe pour résoudre le sous-problème de multiflot, nous augmenterions d'autant la rapidité de la méthode DA \Rightarrow KB. Par contre, la méthode KB semble pour l'instant meilleure que la méthode DA \Rightarrow KB puisqu'elle est beaucoup plus rapide.

Notons qu'il est clairement préférable, comme mentionné plus haut, d'utiliser la méthode des faisceaux plutôt que l'approche primale-duale pour redémarrer l'heuristique de décomposition par montée duale. En effet, la figure 3.3 montre bien que les méthodes DA et KB forment une méthode combinée de montée duale puisque, contrairement à la méthode UPD libre (voir figure 3.1), l'objectif du dual ne diminue jamais.

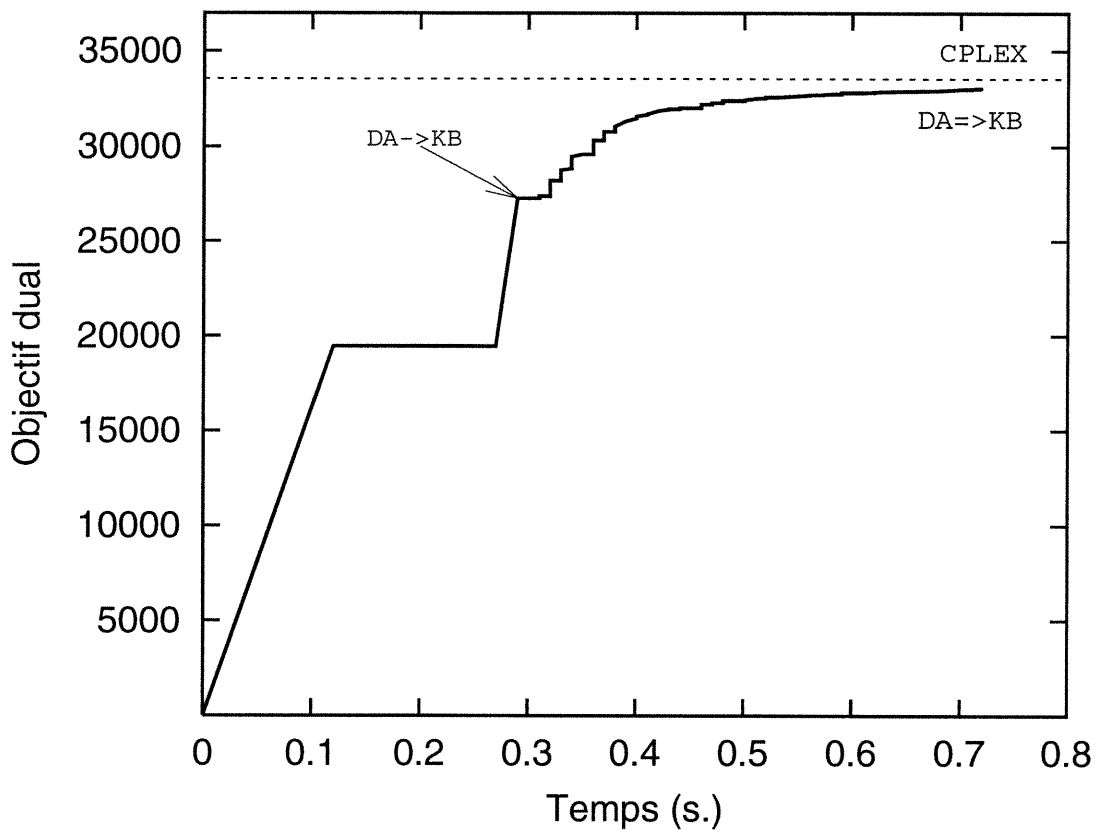


FIG. 3.3 – Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème c25_100_30_F_L résolu avec la méthode $DA \Rightarrow KB$.

Notons finalement que la méthode des faisceaux (KB) obtient le meilleur écart moyen GAP_{lb} , en plus du temps moyen le plus rapide ; cette méthode semble donc être la meilleure pour résoudre le PCMC. Par contre, il est fort probable que la méthode DA (P_1) serait plus rapide que la méthode KB en utilisant la méthode des faisceaux pour résoudre le sous-problème de multiflot puisque la résolution de ce sous-problème prend actuellement 86% du temps de DA (P_1), que les temps moyens des méthodes DA (P_1) et KB ne sont pas très éloignés (3.35 secondes par rapport à 2.08 secondes) et que la méthode des faisceaux est en général beaucoup plus rapide que l'algorithme du simplexe.

3.5.2 Analyse des résultats moyens en fonction des caractéristiques des problèmes

Le tableau 3.14 présente la moyenne des résultats obtenus par les cinq méthodes de résolution du PCMC retenues pour la dernière série de tests sur les 153 problèmes de la classe R. Chaque ligne du tableau donne les résultats moyens des problèmes de la classe R ayant les mêmes valeurs pour le couple (F, C) ; le nombre de problèmes est donné entre parenthèses.

Les résultats du tableau 3.14 reflètent les mêmes conclusions que les résultats moyens obtenus sur l'ensemble des 196 problèmes testés, peu importe les caractéristiques des problèmes :

- CPLEX obtient un temps moyen trop élevé pour s'avérer une méthode efficace ;
- la méthode UPD libre est trop lente pour la faible amélioration de la borne inférieure qu'elle obtient par rapport à DA (P_1) ;
- la méthode DA (P_1) obtient le deuxième meilleur temps moyen, pas très loin derrière KB, ainsi qu'un écart moyen entre 7% et 16%, selon la difficulté des problèmes ;
- la méthode DA \Rightarrow KB obtient le deuxième meilleur écart GAP_{lb} , pas très loin derrière KB, mais est moins rapide que cette dernière ;
- la méthode KB obtient le meilleur écart moyen et le meilleur temps moyen.

Notons que lorsque C prend une valeur petite (1.0) ou moyenne (2.0), le niveau de difficulté des problèmes dépend de la valeur prise par F , peu importe la méthode de résolution utilisée. En effet, l'écart moyen GAP_{lb} augmente à mesure que F augmente pour toutes les méthodes testées. Par contre, le temps moyen demeure à peu près constant pour ces valeurs de C . Lorsque C prend une grande valeur (8.0), alors les problèmes semblent difficiles à résoudre pour toutes les méthodes, peu importe la valeur prise par F . En effet, même lorsque F prend une petite valeur (0.01), les écarts moyens et les temps moyens sont élevés pour toutes les méthodes par rapport aux résultats obtenus lorsque $C \in \{1.0, 2.0\}$.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
	GAP _{lb} (T)	GAP _{lb} (T)	GAP _{lb} (T)	GAP _{lb} (T)	— (T)
0.01_1.0 (18)	7.65 % (1.45 s.)	5.04 % (4.5 s.)	0.94 % (2.55 s.)	0.43 % (1.11 s.)	— (23.12 s.)
0.05_1.0 (18)	13.14 % (1.39 s.)	11.16 % (4.15 s.)	2.19 % (2.57 s.)	0.99 % (1.22 s.)	— (465.08 s.)
0.10_1.0 (18)	15.91 % (1.38 s.)	14.25 % (4.24 s.)	2.16 % (2.58 s.)	1.19 % (1.25 s.)	— (848.69 s.)
0.01_2.0 (18)	7.59 % (1.57 s.)	5.60 % (7.64 s.)	0.99 % (2.72 s.)	0.78 % (1.11 s.)	— (43.91 s.)
0.05_2.0 (18)	12.13 % (1.40 s.)	10.33 % (4.96 s.)	1.64 % (2.62 s.)	0.99 % (1.19 s.)	— (568.97 s.)
0.10_2.0 (18)	14.35 % (1.39 s.)	12.83 % (4.34 s.)	1.36 % (2.65 s.)	1.04 % (1.25 s.)	— (1550.00 s.)
0.01_8.0 (15)	11.40 % (2.16 s.)	10.90 % (17.42 s.)	1.75 % (3.58 s.)	1.55 % (1.40 s.)	— (158.12 s.)
0.05_8.0 (15)	12.17 % (2.07 s.)	11.08 % (9.90 s.)	1.25 % (3.59 s.)	0.91 % (1.50 s.)	— (810.10 s.)
0.10_8.0 (15)	13.64 % (1.88 s.)	12.62 % (9.21 s.)	1.44 % (3.45 s.)	1.02 % (1.60 s.)	— (1309.37 s.)

TAB. 3.14 – Résultats moyens des méthodes retenues sur les 153 problèmes de la classe R en fonction des caractéristiques des problèmes.

3.5.3 Analyse de l'efficacité des sous-problèmes

Les figures 3.4, 3.5 et 3.6 montrent l'évolution typique de l'objectif du dual en fonction du temps pour la méthode DA (P_1). Une flèche indique la valeur de l'objectif du dual après chacun des sous-problèmes de l'heuristique de décomposition par montée duale : conception de réseaux sans capacité (UNDP), multiflot (MF) et sac à dos (KS). La solution optimale de la relaxation linéaire du PCMC, obtenue par CPLEX, est indiquée par une ligne pointillée.

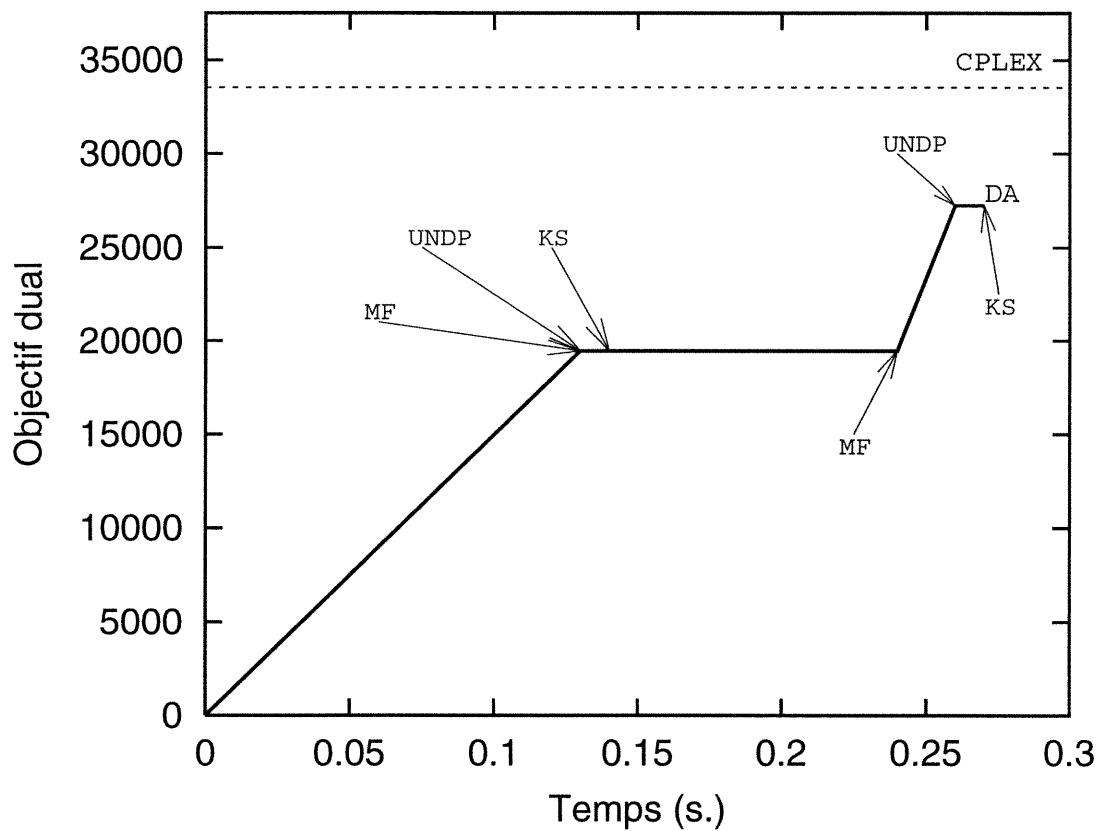


FIG. 3.4 – Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème c25_100_30_F_L résolu avec la méthode DA (P_1).

Les figures 3.4, 3.5 et 3.6 permettent de constater, à l'instar de la figure 3.2, que le sous-problème de multiflot prend effectivement beaucoup de temps à être résolu par rapport aux autres sous-problèmes. En revanche, nous remarquons que le sous-

problème MF réussit généralement à améliorer l'objectif du dual de façon significative. Par conséquent, nous constatons l'utilité du sous-problème MF, mais aussi la nécessité de remplacer l'algorithme du simplexe par une méthode plus rapide pour résoudre ce sous-problème.

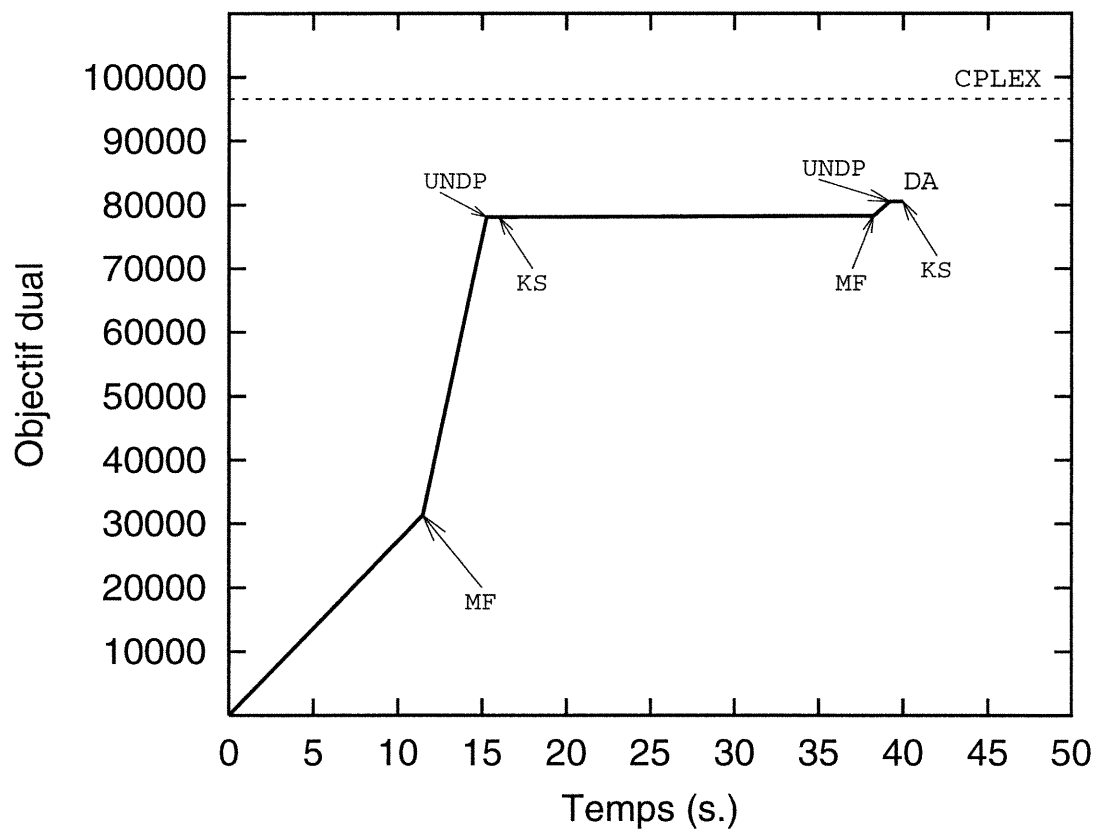


FIG. 3.5 – Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème c30_700-400_V_L résolu avec la méthode DA (P_1).

Le sous-problème UNDP, quant à lui, réussit généralement à améliorer l'objectif du dual de façon significative, en plus d'être très rapide. Finalement, le sous-problème de sac à dos ne réussit pas souvent à faire augmenter la valeur de l'objectif du dual puisque les variables duales π_i^k et γ_{ij} , présentes dans l'objectif, sont fixées. Le sous-problème KS tente, si c'est possible, de faire augmenter l'objectif du dual en modifiant les variables λ_{ij} ; sinon, il donnera plus de latitude au sous-problème UNDP pour faire augmenter l'objectif du dual en maximisant la valeur de la variable d'écart \bar{f} .

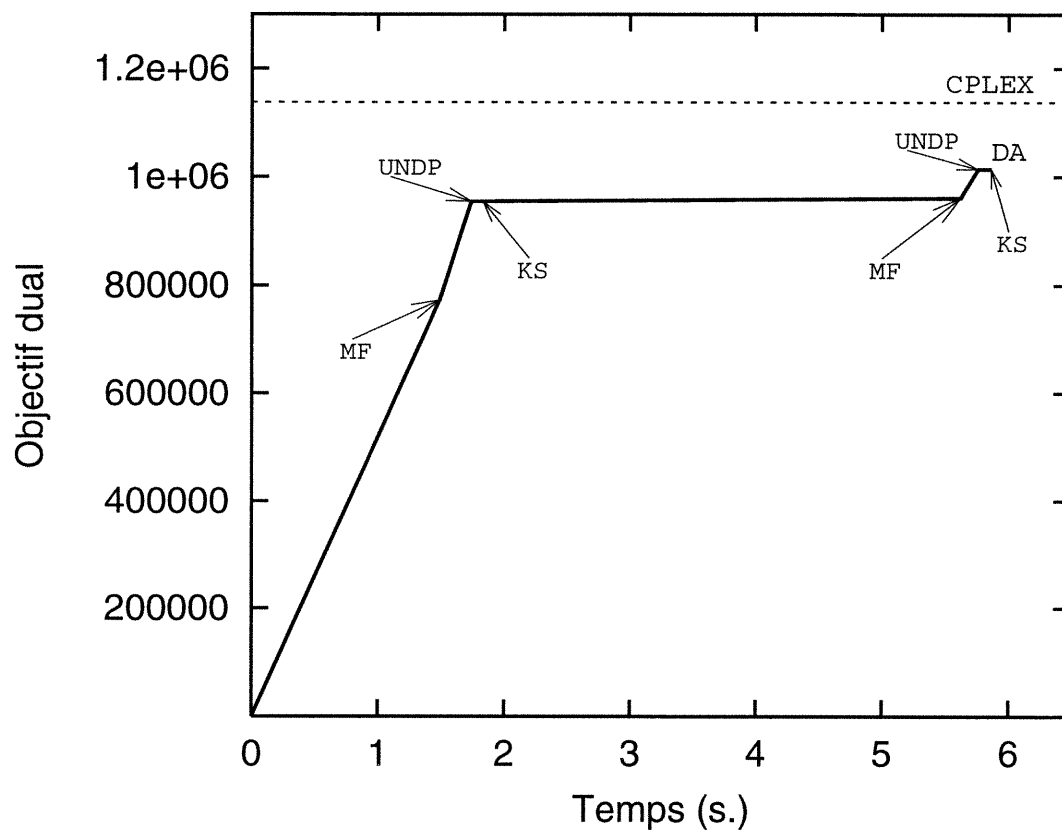


FIG. 3.6 – Graphique de l'évolution de l'objectif dual en fonction du temps pour le problème r20.220.200_0.01_2.0 résolu avec la méthode DA (P_1).

Chapitre 4

Conclusion

L'objectif principal visé dans le cadre de ce mémoire était de développer une méthode *rapide* pour obtenir une bonne borne inférieure à la relaxation linéaire du problème de conception de réseaux multiproduits avec coûts fixes et capacités. Dans cette optique, nous avons développé une heuristique qui résout, l'un à la suite de l'autre, trois sous-problèmes incluant chacun un sous-ensemble strict des variables du modèle dual global de façon à toujours augmenter la valeur de l'objectif dual. Afin d'améliorer davantage la borne inférieure ainsi obtenue, nous avons ensuite combiné cette heuristique de décomposition par montée duale à une des deux méthodes suivantes : une méthode de faisceaux et une approche primale-duale.

Une phase de calibrage nous a permis de déterminer la version la plus prometteuse de chacune des trois méthodes développées dans le cadre de ce mémoire (l'heuristique de décomposition par montée duale, la combinaison de la méthode des faisceaux et de l'heuristique de décomposition par montée duale, la combinaison de l'approche primale-duale et de l'heuristique de décomposition par montée duale). Nous avons ensuite effectué une série de tests sur 196 instances du PCMC afin de comparer les résultats obtenus avec ces trois méthodes retenues avec ceux obtenus avec le logiciel d'optimisation commercial CPLEX et avec la méthode des faisceaux.

La version la plus prometteuse de l'heuristique de décomposition par montée duale s'avère être une méthode rapide pour obtenir une borne inférieure relativement bonne du PCMC, obtenant en moyenne un écart d'environ 12% par rapport à la meilleure borne inférieure connue en 3.35 secondes. Bien que nous ne l'ayons pas vérifié dans ce mémoire, il est fort probable qu'en utilisant une méthode de résolution du sous-problème

de multiflot plus rapide que l'algorithme du simplexe, l'heuristique de décomposition par montée duale serait la méthode la plus rapide parmi les méthodes testées, ce qui suggère une nouvelle avenue de recherche.

Le fait de traiter des sous-problèmes simples issus de la formulation globale du PCMC dans l'heuristique de décomposition par montée duale résulte en une méthode très rapide. Par contre, pour obtenir un écart relatif à la meilleure borne inférieure connue de moins de 5% en moyenne, l'heuristique de décomposition par montée duale doit être combinée à une méthode qui traite toutes les variables du modèle global en même temps. Par exemple, la méthode de résolution du PCMC combinant l'heuristique de décomposition par montée duale et la méthode des faisceaux a permis d'obtenir, en moyenne, un excellent écart GAP_{lb} de 1.63% en 5.46 secondes. De plus, notons qu'en améliorant la rapidité de l'heuristique de décomposition par montée duale, tel que suggéré plus haut, nous améliorerions d'autant la rapidité de cette méthode de résolution combinée ; de cette façon, la méthode combinée $DA \Rightarrow KB$ pourrait se rapprocher des temps obtenus par la méthode des faisceaux (KB), utilisée sans l'heuristique de décomposition par montée duale.

Nous avons développé l'approche primale-duale pour redémarrer l'heuristique de décomposition par montée duale puisque cette approche est souvent suggérée dans la littérature pour redémarrer des méthodes de montée duale. Nous ne nous attendions toutefois pas à obtenir d'excellents résultats grâce à cette méthode puisque, contrairement à la méthode des faisceaux, l'approche primale-duale remet l'objectif dual à zéro, plutôt que de s'inscrire dans un schéma général de montée duale. La version la plus prometteuse de la méthode de résolution du PCMC combinant l'approche primale-duale et l'heuristique de décomposition par montée duale a entraîné une augmentation beaucoup trop grande des temps de résolution par rapport à la faible amélioration de la valeur de la borne inférieure obtenue avec l'heuristique de décomposition par montée duale. Par contre, la méthode combinée utilisant l'approche primale-duale peut constituer une méthode rapide pour obtenir des solutions réalisables, servant à initialiser des méthodes plus sophistiquées.

Bibliographie

- [1] BALAKRISHNAN, A., T.L. MAGNANTI et R.T. WONG, *A Dual-Ascent Procedure for Large-Scale Uncapacitated Network Design*, Operations Research, vol. 37, 1989, pp. 716–740.
- [2] BARNHART, C., *Dual-Ascent Methods for Large-Scale MultiCommodity Flow Problems*, Naval Research Logistics, vol. 40, 1993, pp. 305–324.
- [3] BRASSARD, G. et P. BRATLEY, *Fundamentals of Algorithms*, Prentice Hall, 1996.
- [4] CHOUMAN, M., T.G. CRAINIC et B. GENDRON, *Revue des inégalités valides pertinentes aux problèmes de conception de réseaux*, Publication CRT-2001-38, Centre de recherche sur les transports, Université de Montréal, 2001.
- [5] CPLEX, *Cplex 6.6 Documentation*, ILOG, 2000.
- [6] CRAINIC, T.G., *Service Network Design in Freight Transportation*, European Journal of Operational Research, vol. 122, 2000, pp. 272–288.
- [7] CRAINIC, T.G., A. FRANGIONI et B. GENDRON, *Bundle-Based Relaxation Methods for Multicommodity Capacitated Fixed Charge Network Design Problems*, Discrete Applied Mathematics, vol. 112, 2001, pp. 73–99.
- [8] CRAINIC, T.G., M. GENDREAU et J.M. FARVOLDEN, *A Simplex-Based Tabu Search Method for Capacitated Network Design*, INFORMS Journal on Computing, vol. 12, no. 3, 2000, pp. 223–236.
- [9] ERLINKOTTER, D., *A Dual-Based Procedure for Uncapacitated Facility Location*, Operations Research, vol. 26, 1978, pp. 992–1009.
- [10] FISCHETTI, M. et P. TOTH, *An Additive Bounding Procedure for Combinatorial Optimization Problems*, Operations Research, vol. 37, 1989, pp. 319–328.
- [11] FRANGIONI, A., *Dual-Ascent Methods and Multicommodity Flow Problems*, Thèse de doctorat, 1997. Dipartimento di informatica, Università di Pisa.

- [12] GENDRON, B. et T.G. CRAINIC, *Relaxations for Multicommodity Capacitated Network Design Problems*, Publication CRT-965, Centre de recherche sur les transports, Université de Montréal, 1994.
- [13] GENDRON, B. et T.G. CRAINIC, *A Branch-and-Bound Algorithm for Depot Location and Container Fleet Management*, Location Science, vol. 3, 1995, pp. 39–53.
- [14] GENDRON, B. et T.G. CRAINIC, *Bounding Procedures for Multicommodity Capacitated Fixed Charge Network Design Problems*, Publication CRT-96-06, Centre de recherche sur les transports, Université de Montréal, 1996.
- [15] GENDRON, B., T.G. CRAINIC et A. FRANGIONI, *Multicommodity Capacitated Network Design*, Telecommunications Network Planning, Kluwer Academics Publishers, B. Sansó et P. Soriano (Eds.), 1999, pp. 1–19.
- [16] GHAMLOUCHE, I., T.G. CRAINIC et M. GENDREAU, *Cycle Based Neighborhood Structures for Fixed-Charge Capacitated Multicommodity Network Design*, Publication CRT-2001-01, Centre de recherche sur les transports, Université de Montréal, 2001.
- [17] GUIGNARD, M., *A Lagrangean Dual Ascent Algorithm for Simple Plant Location Problems*, European Journal of Operational Research, vol. 35, 1988, pp. 193–200.
- [18] GUIGNARD, M. et K. OPASWONGKARN, *Lagrangean Dual Ascent Algorithms for Computing Bounds in Capacitated Plant Location Problems*, European Journal of Operational Research, vol. 46, 1990, pp. 73–83.
- [19] GUIGNARD, M. et K. SPIELBERG, *A Direct Dual Method for the Mixed Plant Location Problem with Some Side Constraints*, Mathematical Programming, vol. 17, 1979, pp. 198–228.
- [20] HERNU, G., *Heuristiques basées sur la programmation mathématique pour le problème de conception de réseaux avec coûts fixes et capacités*, publication CRT-2001-11, Centre de recherche sur les transports, Université de Montréal, 2001.
- [21] HOLMBERG, K. et J. HELLSTRAND, *Solving the Uncapacitated Network Design Problem by a Lagrangean Heuristic and Branch-and-Bound*, Operations Research, vol. 46, 1998, pp. 247–259.

- [22] HOLMBERG, K. et D. YUAN, *A Lagrangean Heuristic Based Branch-and-Bound Approach for the Capacitated Network Design Problem*, Operations Research, vol. 48, no. 3, 2000, pp. 461–481.
- [23] LEMARÉCHAL, C., *Nondifferentiable Optimisation*, Handbooks in Operations Research and Management Science, North-Holland Publishers, G.L. Nemhauser, A.H.G. Rinnoy Kan, M.J. Todd (Eds.), vol. 1, 1989, pp. 529–572.
- [24] MAGNANTI, T.L., P. MIREAULT et R.T. WONG, *Tailoring Benders Decomposition for Network Design*, Math. Prog. Study, vol. 26, 1986, pp. 112–154.
- [25] MAGNANTI, T.L. et R.T. WONG, *Network Design and Transportation Planning : Models and Algorithms*, Transportation Science, vol. 18, 1984, pp. 1–55.
- [26] MINOUX, M., *Network Synthesis and Optimum Network Design Problem : Models, Solution Methods and Applications*, Networks, vol. 19, 1989, pp. 313–360.
- [27] MURTHY, I. et P.K. SEO, *A Dual-Ascent Procedure for the File Allocation and Join Site Selection Problem on a Telecommunications Network*, Networks, vol. 33, 1999, pp. 109–124.
- [28] SRIDHARAN, R., *The Capacitated Plant Location Problem*, European Journal of Operational Research, vol. 87, 1995, pp. 203–213.
- [29] VAN ROY, T.J., *A Cross Decomposition Algorithm for Capacitated Facility Location*, Operations Research, vol. 34, 1986, pp. 145–163.
- [30] VAN ROY, T.J. et D. ERLINKOTTER, *A Dual-Ascent Procedure for Dynamic Facility Location*, Management Science, vol. 28, 1982, pp. 1091–1105.

Annexe A : Résultats complets des tests comparatifs

Nous présentons les résultats obtenus par les méthodes de résolution DA (P_1), UPD libre, DA \Rightarrow KB, KB et CPLEX (voir chapitre 3) pour chacun des 196 problèmes des classes C et R. Chaque case des tableaux suivants donne, pour une méthode de résolution et un problème choisis, la borne inférieure obtenue sur la première ligne, l'écart GAP_{lb} par rapport à la borne obtenue par CPLEX sur la deuxième ligne, ainsi que le temps CPU sur la troisième ligne.

Si la borne inférieure obtenue par CPLEX est entourée de deux astérisques, cela signifie que le temps limite de 10 heures a été atteint avant que CPLEX ne trouve la solution optimale de la relaxation linéaire du PCMC.

Problème	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
c20_230_40_V_L	399452.3 (5.53 %) (0.64 s.)	417055.21 (1.37 %) 3.03 s.)	422508.09 (0.08 %) (1.21 s.)	422588.63 (0.06 %) (0.55 s.)	422853.26 — (0.43 s.)
c20_230_40_F_T	362292.99 (1.76 %) (0.81 s.)	366562.76 (0.61 %) 2.75 s.)	367280.02 (0.41 %) (1.43 s.)	367692.43 (0.30 %) (0.64 s.)	368819.45 — (0.70 s.)
c20_230_40_F_T	605758.1 (4.37 %) (0.79 s.)	623141.53 (1.62 %) 2.84 s.)	631455.87 (0.31 %) (1.41 s.)	631244.05 (0.35 %) (0.64 s.)	633466.16 — (0.78 s.)
c20_230_200_V_L	75635.72 (17.15 %) (5.32 s.)	75635.72 (17.15 %) 12.7 s.)	85956.90 (5.85 %) (9.28 s.)	89732.72 (1.71 %) (4.48 s.)	91300.64 — (1105.45 s.)
<i>suite à la page suivante</i>					

Problème	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
c20_230_200_F_L	108220.56 (18.03 %) (4.91 s.)	108220.56 (18.03 %) 13.39 s.)	125194.76 (5.18 %) (9.01 s.)	130401.68 (1.23 %) (4.36 s.)	132036.01 — (2450.18 s.)
c20_230_200_V_T	79711.28 (16.68 %) (5.24 s.)	79711.28 (16.68 %) 12.15 s.)	91714.81 (4.13 %) (9.29 s.)	94254.37 (1.47 %) (4.2 s.)	95669.33 — (592.60 s.)
c20_230_200_F_T	109543.32 (16.72 %) (4.76 s.)	109543.32 (16.72 %) 14.95 s.)	126604.89 (3.75 %) (8.95 s.)	130073.44 (1.11 %) (4.07 s.)	131544.12 — (2506.87 s.)
c20_300_40_V_L	406631.78 (4.98 %) (0.84 s.)	424830.73 (0.72 %) 2.81 s.)	427477.9 (0.10 %) (1.46 s.)	427780.37 (0.03 %) (0.69 s.)	427947.37 — (0.58 s.)
c20_300_40_F_L	532100.76 (7.50 %) (0.92 s.)	544248.4 (5.39 %) 3.54 s.)	569831.83 (0.94 %) (1.65 s.)	572419.7 (0.49 %) (0.76 s.)	575255.00 — (1.80 s.)
c20_300_40_V_T	453149.55 (1.68 %) (0.92 s.)	459175.76 (0.38 %) 3.15 s.)	460179.6 (0.16 %) (1.65 s.)	460135.85 (0.17 %) (0.7 s.)	460930.28 — (0.74 s.)
c20_300_40_F_T	560656.46 (6.06 %) (0.89 s.)	585301 (1.93 %) 3.37 s.)	595124.18 (0.28 %) (1.61 s.)	596048.68 (0.13 %) (0.7 s.)	596838.68 — (0.91 s.)
c20_300_200_V_L	62280.78 (14.83 %) (6.42 s.)	62280.78 (14.83 %) 20.06 s.)	72185.25 (1.28 %) (11.03 s.)	72435.47 (0.94 %) (4.59 s.)	73126.89 — (1250.63 s.)
c20_300_200_F_L	93788.03 (15.45 %) (7.1 s.)	93788.03 (15.45 %) 17.22 s.)	109043.01 (1.69 %) (11.85 s.)	109852.11 (0.96 %) (4.58 s.)	110926.16 — (3588.22 s.)
<i>suite à la page suivante</i>					

Problème	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
c20_300_200_V_T	63654.15 (13.98 %) (7.1 s.)	63654.15 (13.98 %) 18.94 s.)	71917.50 (2.81 %) (11.53 s.)	73537.90 (0.62 %) (4.48 s.)	74002.71 — (741.08 s.)
c20_300_200_F_T	89148.84 (13.97 %) (6.98 s.)	89148.84 (13.97 %) 19.01 s.)	102381.8 (1.20 %) (11.7 s.)	102427.43 (1.16 %) (4.87 s.)	103632.54 — (2319.79 s.)
c25_100_10_F_L	11091.96 (14.79 %) (0.08 s.)	11091.96 (14.79 %) 0.32 s.)	12933.82 (0.64 %) (0.26 s.)	12895.56 (0.93 %) (0.18 s.)	13017.50 — (0.58 s.)
c25_100_10_F_T	37280.78 (14.20 %) (0.09 s.)	38515.81 (11.36 %) 0.51 s.)	43110.66 (0.79 %) (0.29 s.)	43186.27 (0.61 %) (0.21 s.)	43454.67 — (0.61 s.)
c25_100_10_V_L	14319.89 (1.98 %) (0.06 s.)	14515.84 (0.64 %) 0.23 s.)	14591.14 (0.12 %) (0.24 s.)	14584.33 (0.17 %) (0.14 s.)	14610 — (0.06 s.)
c25_100_30_F_L	27257.33 (18.74 %) (0.27 s.)	28504.43 (15.02 %) 3.79 s.)	33036.36 (1.51 %) (0.74 s.)	33005.51 (1.60 %) (0.46 s.)	33543.45 — (5.47 s.)
c25_100_30_F_T	73904.54 (10.33 %) (0.35 s.)	78880.19 (4.29 %) 1.65 s.)	81581.38 (1.01 %) (0.78 s.)	81589.65 (1.00 %) (0.42 s.)	82418.50 — (1.79 s.)
c25_100_30_V_T	363637.26 (0.20 %) (0.98 s.)	363811.49 (0.15 %) 2.57 s.)	363928.91 (0.12 %) (1.42 s.)	361550.63 (0.77 %) (0.4 s.)	364390.25 — (0.63 s.)
c30_520_100_V_L	46354.49 (12.57 %) (6.28 s.)	46800.33 (11.73 %) 24.91 s.)	50579.95 (4.60 %) (9.92 s.)	52410.61 (1.15 %) (3.52 s.)	53022.90 — (56.15 s.)
<i>suite à la page suivante</i>					

Problème	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
c30_520_100_F_L	74740.20 (17.11 %) (6.29 s.)	74740.20 (17.11 %) 15.85 s.)	85034.68 (5.69 %) (9.87 s.)	88960.43 (1.34 %) (3.62 s.)	90174.16 — (1153.64 s.)
c30_520_100_V_T	47437.38 (7.57 %) (7.74 s.)	47967.34 (6.54 %) 29.05 s.)	50638.78 (1.33 %) (10.96 s.)	50975.33 (0.68 %) (3.49 s.)	51325.57 — (44.88 s.)
c30_520_100_F_T	81012.67 (13.82 %) (6.24 s.)	81012.67 (13.82 %) 14.66 s.)	90985.53 (3.21 %) (9.66 s.)	92602.85 (1.49 %) (3.45 s.)	94010.77 — (777.38 s.)
c30_520_400_V_L	97655.10 (12.62 %) (28.39 s.)	97655.10 (12.62 %) 82.6 s.)	110311.41 (1.29 %) (43.81 s.)	110632.18 (1.01 %) (14.83 s.)	111762.70 — (4431.86 s.)
c30_520_400_F_L	126206.11 (13.95 %) (30.45 s.)	126206.11 (13.95 %) 74.54 s.)	145022.95 (1.12 %) (47.11 s.)	145189.21 (1.01 %) (15.55 s.)	146680.22 — (10354.01 s.)
c30_520_400_V_T	99099.43 (13.11 %) (28.39 s.)	99099.43 (13.11 %) 113.07 s.)	113221.98 (0.73 %) (43.77 s.)	113254.86 (0.70 %) (14.66 s.)	114060.72 — (3259.62 s.)
c30_520_400_F_T	128556.82 (14.15 %) (29.2 s.)	128556.82 (14.15 %) 78.36 s.)	148326.78 (0.95 %) (45.74 s.)	148460.51 (0.86 %) (14.93 s.)	149750.52 — (10939.43 s.)
c30_700_100_V_L	41661.30 (11.93 %) (11.15 s.)	43377.12 (8.30 %) 68.44 s.)	45625.39 (3.55 %) (15.39 s.)	46916.51 (0.82 %) (4.1 s.)	47308.11 — (47.33 s.)
c30_700_100_F_L	49555.45 (14.86 %) (11.08 s.)	49555.45 (14.86 %) 24.38 s.)	54856.09 (5.75 %) (15.51 s.)	57198.35 (1.73 %) (4.32 s.)	58207.22 — (684.81 s.)
<i>suite à la page suivante</i>					

Problème	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
c30_700_100_V_T	41863.98 (7.12 %) (9.69 s.)	41863.98 (7.12 %) 23.48 s.)	44506.21 (1.26 %) (14.27 s.)	44642.71 (0.96 %) (4.13 s.)	45077.66 — (132.94 s.)
c30_700_100_F_T	47669.52 (11.16 %) (9.13 s.)	47949.40 (10.64 %) 36.51 s.)	51995.18 (3.10 %) (13.29 s.)	53059.20 (1.12 %) (4.22 s.)	53660.85 — (306.06 s.)
c30_700_400_V_L	80588.98 (16.57 %) (39.97 s.)	80588.98 (16.57 %) 119.87 s.)	95650.33 (0.98 %) (58.28 s.)	95639.92 (0.99 %) (17.75 s.)	96605.02 — (11845.61 s.)
c30_700_400_F_L	109737.16 (15.66 %) (43.91 s.)	109737.16 (15.66 %) 109.25 s.)	128642.44 (1.13 %) (63.87 s.)	129411.75 (0.54 %) (18.55 s.)	*130115.42* — (36040.22 s.)
c30_700_400_V_T	82057.1 (12.71 %) (39.13 s.)	82057.1 (12.71 %) 101.24 s.)	93002.78 (1.07 %) (55.87 s.)	93023.25 (1.05 %) (17.22 s.)	94011.91 — (16395.08 s.)
c30_700_400_F_T	110514.15 (13.37 %) (40.34 s.)	110514.15 (13.37 %) 98.78 s.)	126229.61 (1.05 %) (59.24 s.)	126463.49 (0.86 %) (17.44 s.)	127572.11 — (27552.77 s.)
c100_400_10_F_L	15418.84 (21.92 %) (0.48 s.)	15418.84 (21.92 %) 9.73 s.)	19070.24 (3.43 %) (1.04 s.)	19232.99 (2.60 %) (0.6 s.)	19748.12 — (12.52 s.)
c100_400_10_F_T	39084.99 (19.20 %) (0.55 s.)	40503.66 (16.27 %) 5.76 s.)	46859.96 (3.13 %) (1.14 s.)	47196.32 (2.43 %) (0.63 s.)	48375.39 — (15.77 s.)
c100_400_10_V_L	27099.25 (1.33 %) (0.42 s.)	27190.59 (1.00 %) 1.27 s.)	27394.72 (0.25 %) (0.97 s.)	27250.13 (0.78 %) (0.53 s.)	27465.31 — (0.35 s.)
<i>suite à la page suivante</i>					

Problème	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
c100_400_30_F_L	34289.41 (24.35 %) (2.3 s.)	36544.27 (19.38 %) 9.14 s.)	43579.52 (3.86 %) (4.38 s.)	43761.69 (3.46 %) (2.08 s.)	45332.35 — (107.28 s.)
c100_400_30_F_T	103267.89 (11.88 %) (2.04 s.)	103720.05 (11.49 %) 12.06 s.)	114374.39 (2.40 %) (4.09 s.)	114960.97 (1.90 %) (1.88 s.)	117195.66 — (99.72 s.)
c100_400_30_V_T	331533.26 (12.95 %) (1.91 s.)	331533.26 (12.95 %) 11.57 s.)	369437.72 (2.99 %) (3.71 s.)	368965.92 (3.12 %) (1.83 s.)	380857.66 — (3.77 s.)

TAB. 5.1 – Résultats obtenus par les méthodes retenues sur les problèmes de la série C.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	72760.96 (1.77 %) (0.02 s.)	74079 (0 %) 0.06 s.)	74079 (0 %) (0.03 s.)	74079 (0 %) (0.02 s.)	74079 — (0.02 s.)
0.05_1.0	86113.91 (6.80 %) (0.03 s.)	91213.22 (1.28 %) 0.07 s.)	92403 (0 %) (0.05 s.)	92403 (0 %) (0.03 s.)	92403 — (0.03 s.)
0.10_1.0	101823.51 (11.26 %) (0.02 s.)	108291.78 (5.62 %) 0.1 s.)	114746 (0 %) (0.06 s.)	114746 (0 %) (0.04 s.)	114746 — (0.03 s.)
0.01_2.0	81693.97 (1.35 %) (0.03 s.)	81943.26 (1.05 %) 0.07 s.)	82812.21 (0.00 %) (0.1 s.)	82816.97 (0 %) (0.13 s.)	82816.97 — (0.03 s.)
0.05_2.0	100326.45 (4.91 %) (0.02 s.)	102300.46 (3.04 %) 0.09 s.)	105455.44 (0.05 %) (0.12 s.)	105514.71 (0 %) (0.1 s.)	105514.75 — (0.02 s.)
0.10_2.0	122504.63 (8.29 %) (0.03 s.)	124359.76 (6.91 %) 0.08 s.)	133570.04 (0.01 %) (0.1 s.)	133554.15 (0.02 %) (0.1 s.)	133591.17 — (0.04 s.)

TAB. 5.2 – Résultats obtenus par les méthodes retenues sur les problèmes r10_35_10.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	226985.83 (1.59 %) (0.05 s.)	230085.53 (0.25 %) 0.2 s.)	230457.68 (0.09 %) (0.21 s.)	230597.46 (0.03 %) (0.16 s.)	230675.86 — (0.08 s.)
0.05_1.0	292799.53 (7.98 %) (0.06 s.)	310538.52 (2.41 %) 0.36 s.)	317788.99 (0.13 %) (0.21 s.)	317669.01 (0.17 %) (0.16 s.)	318219.70 — (0.14 s.)
0.10_1.0	358198.9 (12.45 %) (0.08 s.)	388624.74 (5.01 %) 0.23 s.)	408234.71 (0.22 %) (0.22 s.)	408256.52 (0.21 %) (0.16 s.)	409156.44 — (0.17 s.)
0.01_2.0	306670.35 (2.17 %) (0.1 s.)	312148.98 (0.43 %) 0.44 s.)	313302.68 (0.06 %) (0.26 s.)	302439.65 (3.52 %) (0.13 s.)	313497.15 — (0.13 s.)
0.05_2.0	398548.58 (5.37 %) (0.1 s.)	407830.29 (3.17 %) 0.28 s.)	420449.83 (0.17 %) (0.25 s.)	411988.06 (2.18 %) (0.16 s.)	421191.11 — (0.14 s.)
0.10_2.0	491387.24 (8.77 %) (0.07 s.)	522077.03 (3.07 %) 0.36 s.)	537686.97 (0.17 %) (0.23 s.)	536717.21 (0.35 %) (0.15 s.)	538629.53 — (0.19 s.)

TAB. 5.3 – Résultats obtenus par les méthodes retenues sur les problèmes r10_35_25.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	471594.21 (2.45 %) (0.25 s.)	480237.99 (0.66 %) 0.81 s.)	480937.1 (0.52 %) (0.53 s.)	481293.28 (0.44 %) (0.28 s.)	483467.46 — (0.26 s.)
0.05_1.0	638506.18 (8.09 %) (0.18 s.)	656360.06 (5.52 %) 0.71 s.)	689805.37 (0.71 %) (0.46 s.)	690732.69 (0.57 %) (0.27 s.)	694748.94 — (0.33 s.)
0.10_1.0	827196.98 (11.27 %) (0.14 s.)	860470.86 (7.70 %) 1.05 s.)	925955.93 (0.67 %) (0.44 s.)	926370.49 (0.63 %) (0.28 s.)	932263.12 — (0.53 s.)
0.01_2.0	692867.71 (1.59 %) (0.39 s.)	698752.01 (0.75 %) 1.07 s.)	701746.16 (0.33 %) (0.69 s.)	697244.88 (0.97 %) (0.27 s.)	704095.33 — (0.38 s.)
0.05_2.0	878309.2 (5.35 %) (0.25 s.)	889821.85 (4.11 %) 1.12 s.)	922755.19 (0.56 %) (0.55 s.)	919260.98 (0.94 %) (0.26 s.)	928014.64 — (0.43 s.)
0.10_2.0	1093923.6 (7.33 %) (0.23 s.)	1113446 (5.67 %) 0.91 s.)	1167798.2 (1.07 %) (0.51 s.)	1173565.5 (0.58 %) (0.29 s.)	1180474.13 — (0.46 s.)

TAB. 5.4 – Résultats obtenus par les méthodes retenues sur les problèmes r10_35_50.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	30946.50 (2.46 %) (0.02 s.)	31730 (0 %) 0.1 s.)	31730 (0 %) (0.05 s.)	31730 (0 %) (0.01 s.)	31730 — (0.02 s.)
0.05_1.0	44822.10 (8.37 %) (0.03 s.)	46127.30 (5.70 %) 0.18 s.)	48918.72 (0.00 %) (0.11 s.)	48920 (0 %) (0.05 s.)	48920 — (0.04 s.)
0.10_1.0	57853.10 (9.27 %) (0.04 s.)	60120.99 (5.71 %) 0.15 s.)	63767 (0 %) (0.09 s.)	63767 (0 %) (0.05 s.)	63767 — (0.05 s.)
0.01_2.0	31679.53 (3.46 %) (0.04 s.)	32438.75 (1.14 %) 0.15 s.)	32815.21 (0 %) (0.1 s.)	32815.21 (0 %) (0.07 s.)	32815.21 — (0.04 s.)
0.05_2.0	44875.64 (11.66 %) (0.04 s.)	48916.91 (3.71 %) 0.12 s.)	50743.14 (0.11 %) (0.12 s.)	50799.67 (0.00 %) (0.11 s.)	50803.49 — (0.06 s.)
0.10_2.0	58228.38 (14.66 %) (0.04 s.)	63527.61 (6.89 %) 0.1 s.)	68184.63 (0.07 %) (0.12 s.)	68231.24 (0.00 %) (0.1 s.)	68233.05 — (0.07 s.)
0.01_8.0	64532.50 (1.86 %) (0.07 s.)	65180.03 (0.88 %) 0.17 s.)	65759.81 (0.00 %) (0.15 s.)	65693.87 (0.10 %) (0.09 s.)	65761.69 — (0.08 s.)
0.05_8.0	98230.73 (5.83 %) (0.07 s.)	101526.77 (2.67 %) 0.14 s.)	104298.26 (0.01 %) (0.17 s.)	104299.99 (0.01 %) (0.11 s.)	104316.70 — (0.11 s.)
0.10_8.0	135637.09 (8.68 %) (0.05 s.)	141174.72 (4.96 %) 0.15 s.)	148545.26 (0 %) (0.15 s.)	148528.53 (0.01 %) (0.11 s.)	148545.25 — (0.10 s.)

TAB. 5.5 – Résultats obtenus par les méthodes retenues sur les problèmes r10_60_10.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	116725.44 (4.76 %) (0.08 s.)	120580.88 (1.61 %) 0.28 s.)	122206.21 (0.29 %) (0.24 s.)	122546.9 (0.01 %) (0.18 s.)	122563.92 — (0.07 s.)
0.05_1.0	151779.81 (9.64 %) (0.08 s.)	157715.25 (6.11 %) 0.3 s.)	166127.11 (1.10 %) (0.23 s.)	167443.45 (0.32 %) (0.17 s.)	167990.15 — (0.13 s.)
0.10_1.0	185001.25 (13.16 %) (0.06 s.)	187109.53 (12.17 %) 0.31 s.)	211296.13 (0.82 %) (0.28 s.)	211477.15 (0.73 %) (0.16 s.)	213048.12 — (0.28 s.)
0.01_2.0	124693.93 (3.68 %) (0.1 s.)	127283.46 (1.68 %) 0.4 s.)	129335.68 (0.09 %) (0.27 s.)	129354.74 (0.08 %) (0.17 s.)	129459.11 — (0.10 s.)
0.05_2.0	173971.22 (8.99 %) (0.07 s.)	181435.56 (5.08 %) 0.32 s.)	190014.98 (0.60 %) (0.28 s.)	190648.41 (0.26 %) (0.16 s.)	191162.17 — (0.33 s.)
0.10_2.0	228782.36 (12.61 %) (0.1 s.)	236781.13 (9.55 %) 0.37 s.)	259820.96 (0.75 %) (0.3 s.)	259772.05 (0.77 %) (0.19 s.)	261805.17 — (0.65 s.)
0.01_8.0	267159.26 (2.14 %) (0.26 s.)	270783.3 (0.81 %) 0.77 s.)	271840.87 (0.43 %) (0.47 s.)	266885.78 (2.24 %) (0.2 s.)	273015.78 — (0.40 s.)
0.05_8.0	399142.89 (7.80 %) (0.17 s.)	415260.51 (4.08 %) 0.65 s.)	431227.44 (0.39 %) (0.38 s.)	429703.45 (0.74 %) (0.22 s.)	432935.15 — (0.43 s.)
0.10_8.0	545508.76 (10.33 %) (0.16 s.)	570369.45 (6.25 %) 0.55 s.)	602749.26 (0.92 %) (0.39 s.)	605615.81 (0.45 %) (0.2 s.)	608402.92 — (0.61 s.)

TAB. 5.6 – Résultats obtenus par les méthodes retenues sur les problèmes r10_60_25.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	230422.71 (5.53 %) (0.24 s.)	237447.07 (2.65 %) 0.77 s.)	243116.65 (0.32 %) (0.54 s.)	243566.96 (0.14 %) (0.27 s.)	243918.81 — (0.32 s.)
0.05_1.0	340124.69 (12.91 %) (0.19 s.)	348585.38 (10.75 %) 1.1 s.)	387987.31 (0.66 %) (0.5 s.)	388460 (0.54 %) (0.32 s.)	390577.36 — (0.99 s.)
0.10_1.0	442207.33 (17.62 %) (0.19 s.)	442207.33 (17.62 %) 0.53 s.)	529765.67 (1.30 %) (0.48 s.)	533555.32 (0.60 %) (0.34 s.)	536790.43 — (2.40 s.)
0.01_2.0	267355.85 (5.10 %) (0.24 s.)	275180.81 (2.33 %) 0.9 s.)	280561.18 (0.42 %) (0.55 s.)	280954.48 (0.28 %) (0.3 s.)	281750.78 — (0.50 s.)
0.05_2.0	430212.86 (10.03 %) (0.21 s.)	430212.86 (10.03 %) 0.57 s.)	474079.8 (0.86 %) (0.53 s.)	474944.32 (0.68 %) (0.33 s.)	478222.33 — (1.49 s.)
0.10_2.0	610569.26 (12.00 %) (0.18 s.)	610569.26 (12.00 %) 0.77 s.)	688142.88 (0.82 %) (0.52 s.)	689626.85 (0.61 %) (0.34 s.)	693864.63 — (2.68 s.)
0.01_8.0	668135.95 (1.91 %) (1.06 s.)	674011.08 (1.04 %) 3.21 s.)	679701.38 (0.21 %) (1.48 s.)	664916.06 (2.38 %) (0.41 s.)	681146.95 — (1.40 s.)
0.05_8.0	952204.07 (7.52 %) (0.95 s.)	999676.23 (2.91 %) 2.55 s.)	1020893.8 (0.85 %) (1.37 s.)	1018862.5 (1.05 %) (0.37 s.)	1029726 — (1.72 s.)
0.10_8.0	359784.63 (12.60 %) (0.13 s.)	359784.63 (12.60 %) 0.4 s.)	406293.29 (1.31 %) (0.43 s.)	408866.81 (0.68 %) (0.3 s.)	411687.99 — (1.27 s.)

TAB. 5.7 – Résultats obtenus par les méthodes retenues sur les problèmes r10_60-50.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	32395.99 (1.25 %) (0.05 s.)	32807 (0 %) 0.11 s.)	32807 (0 %) (0.07 s.)	32807 (0 %) (0.02 s.)	32807 — (0.03 s.)
0.05_1.0	44931.56 (4.86 %) (0.05 s.)	46996.48 (0.49 %) 0.12 s.)	47230.46 (0 %) (0.09 s.)	47230.46 (0 %) (0.05 s.)	47230.46 — (0.04 s.)
0.10_1.0	58227.58 (6.31 %) (0.05 s.)	60311.49 (2.95 %) 0.2 s.)	62113.61 (0.05 %) (0.13 s.)	62149.69 (0 %) (0.07 s.)	62149.69 — (0.05 s.)
0.01_2.0	35347.49 (2.54 %) (0.05 s.)	36000.91 (0.74 %) 0.16 s.)	36269.92 (0 %) (0.13 s.)	36269.92 (0 %) (0.04 s.)	36269.92 — (0.02 s.)
0.05_2.0	47210.81 (9.21 %) (0.05 s.)	50532.03 (2.82 %) 0.21 s.)	51999.05 (0.00 %) (0.15 s.)	52003.56 (0 %) (0.06 s.)	52003.55 — (0.05 s.)
0.10_2.0	61073.80 (11.51 %) (0.05 s.)	65647.07 (4.89 %) 0.2 s.)	68991.97 (0.04 %) (0.2 s.)	69025.15 (0 %) (0.08 s.)	69025.15 — (0.09 s.)
0.01_8.0	57361.94 (0.70 %) (0.07 s.)	57361.94 (0.70 %) 0.1 s.)	57733.82 (0.05 %) (0.17 s.)	57753.01 (0.02 %) (0.12 s.)	57766.31 — (0.10 s.)
0.05_8.0	89360.12 (3.75 %) (0.05 s.)	89360.12 (3.75 %) 0.11 s.)	92847.19 (0.00 %) (0.15 s.)	92791.29 (0.06 %) (0.11 s.)	92847.34 — (0.15 s.)
0.10_8.0	121725.26 (6.31 %) (0.05 s.)	122603.89 (5.63 %) 0.14 s.)	129924.62 (0.00 %) (0.17 s.)	129840.76 (0.06 %) (0.1 s.)	129928.99 — (0.22 s.)

TAB. 5.8 – Résultats obtenus par les méthodes retenues sur les problèmes r10_85_10.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01.1.0	97940.34 (3.87 %) (0.13 s.)	101096.24 (0.77 %) 0.36 s.)	101857.74 (0.03 %) (0.3 s.)	101888.32 (0 %) (0.15 s.)	101888.31 — (0.10 s.)
0.05.1.0	131297.92 (8.33 %) (0.1 s.)	135373.2 (5.49 %) 0.4 s.)	143036.34 (0.14 %) (0.29 s.)	143241.33 (0.00 %) (0.17 s.)	143242.25 — (0.15 s.)
0.10.1.0	155992.1 (14.09 %) (0.13 s.)	163760.07 (9.81 %) 0.39 s.)	181456.7 (0.07 %) (0.25 s.)	181256.16 (0.18 %) (0.2 s.)	181583.95 — (0.18 s.)
0.01.2.0	104271.04 (3.78 %) (0.13 s.)	107495.37 (0.80 %) 0.36 s.)	108224 (0.13 %) (0.29 s.)	108327.8 (0.03 %) (0.2 s.)	108370.25 — (0.12 s.)
0.05.2.0	135721.02 (11.28 %) (0.11 s.)	144829.22 (5.32 %) 0.39 s.)	152466.4 (0.33 %) (0.29 s.)	152683.53 (0.19 %) (0.17 s.)	152979.20 — (0.24 s.)
0.10.2.0	172465.75 (12.23 %) (0.09 s.)	174833.03 (11.03 %) 0.38 s.)	195685.14 (0.42 %) (0.29 s.)	194935.06 (0.80 %) (0.21 s.)	196511.69 — (0.58 s.)
0.01.8.0	145601.79 (2.56 %) (0.17 s.)	145601.79 (2.56 %) 0.36 s.)	148899.15 (0.35 %) (0.38 s.)	149250.9 (0.11 %) (0.17 s.)	149428.52 — (0.36 s.)
0.05.8.0	233517.75 (7.76 %) (0.18 s.)	237069.78 (6.35 %) 0.59 s.)	251849.63 (0.52 %) (0.38 s.)	251920.36 (0.49 %) (0.23 s.)	253167.05 — (0.77 s.)
0.10.8.0	332031.51 (10.83 %) (0.16 s.)	337142.76 (9.46 %) 0.75 s.)	370451.71 (0.52 %) (0.33 s.)	371259.28 (0.30 %) (0.21 s.)	372398.03 — (1.26 s.)

TAB. 5.9 – Résultats obtenus par les méthodes retenues sur les problèmes r10_85_25.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	158710.35 (6.95 %) (0.24 s.)	166898.15 (2.15 %) 0.9 s.)	169493.1 (0.63 %) (0.58 s.)	170402.46 (0.10 %) (0.32 s.)	170578.63 — (0.25 s.)
0.05_1.0	253786.78 (13.02 %) (0.28 s.)	253786.78 (13.02 %) 0.66 s.)	287521.63 (1.46 %) (0.61 s.)	289730.96 (0.71 %) (0.4 s.)	291805.31 — (1.43 s.)
0.10_1.0	342873.24 (15.94 %) (0.27 s.)	342873.24 (15.94 %) 0.74 s.)	403900.04 (0.98 %) (0.65 s.)	403699.06 (1.03 %) (0.37 s.)	407903.36 — (3.63 s.)
0.01_2.0	178821.94 (6.51 %) (0.27 s.)	185755.44 (2.89 %) 1.1 s.)	190439.96 (0.44 %) (0.6 s.)	190819.07 (0.24 %) (0.36 s.)	191285.14 — (0.53 s.)
0.05_2.0	306685.11 (11.91 %) (0.23 s.)	306685.11 (11.91 %) 0.85 s.)	345874.15 (0.66 %) (0.61 s.)	346641.9 (0.44 %) (0.36 s.)	348188.85 — (1.85 s.)
0.10_2.0	430519.04 (13.94 %) (0.24 s.)	430519.04 (13.94 %) 0.87 s.)	497519.36 (0.54 %) (0.64 s.)	497603.81 (0.53 %) (0.4 s.)	500263.90 — (4.54 s.)
0.01_8.0	329201.34 (3.22 %) (0.54 s.)	335465.85 (1.38 %) 3.47 s.)	338446.17 (0.51 %) (0.93 s.)	338882.91 (0.38 %) (0.37 s.)	340188.50 — (1.32 s.)
0.05_8.0	602812.05 (5.71 %) (0.4 s.)	602812.05 (5.71 %) 1.27 s.)	637696.58 (0.26 %) (0.73 s.)	637888.38 (0.23 %) (0.38 s.)	639376.01 — (2.71 s.)
0.10_8.0	855412.43 (7.91 %) (0.37 s.)	855412.43 (7.91 %) 1.1 s.)	925672.25 (0.34 %) (0.79 s.)	925943.18 (0.32 %) (0.36 s.)	928921.58 — (5.06 s.)

TAB. 5.10 – Résultats obtenus par les méthodes retenues sur les problèmes r10_85_50.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	186237.05 (6.37 %) (0.33 s.)	194935.24 (2.00 %) 1.15 s.)	198380.6 (0.26 %) (0.77 s.)	198213.68 (0.35 %) (0.48 s.)	198914.14 — (0.30 s.)
0.05_1.0	290875.86 (12.77 %) (0.32 s.)	290875.86 (12.77 %) 1 s.)	329527.9 (1.18 %) (0.84 s.)	326126.53 (2.20 %) (0.54 s.)	333474.06 — (2.06 s.)
0.10_1.0	392931.28 (15.88 %) (0.32 s.)	392931.28 (15.88 %) 1.31 s.)	458911.15 (1.76 %) (0.81 s.)	451864.44 (3.27 %) (0.55 s.)	467147.19 — (6.27 s.)
0.01_2.0	209836.49 (6.26 %) (0.44 s.)	216118.93 (3.45 %) 1.65 s.)	222288.1 (0.70 %) (0.91 s.)	222720.07 (0.51 %) (0.51 s.)	223863.77 — (1.07 s.)
0.05_2.0	347064.95 (10.14 %) (0.34 s.)	347064.95 (10.14 %) 1.05 s.)	381727.85 (1.16 %) (0.85 s.)	380759.09 (1.41 %) (0.5 s.)	386228.66 — (4.38 s.)
0.10_2.0	487212.21 (13.82 %) (0.33 s.)	487212.21 (13.82 %) 1.55 s.)	550591.02 (2.61 %) (0.88 s.)	552712.74 (2.23 %) (0.5 s.)	565351.47 — (9.27 s.)
0.01_8.0	445088.05 (3.99 %) (2.26 s.)	445088.05 (3.99 %) 4.87 s.)	460700.58 (0.62 %) (2.8 s.)	449990.14 (2.93 %) (0.51 s.)	463612.25 — (7.35 s.)
0.05_8.0	788693.78 (9.87 %) (1.09 s.)	806137.11 (7.88 %) 9.95 s.)	862965.36 (1.38 %) (1.6 s.)	861540.14 (1.55 %) (0.55 s.)	875113.06 — (11.48 s.)
0.10_8.0	1141138.3 (11.89 %) (0.75 s.)	1190117.2 (8.11 %) 7.72 s.)	1274312.9 (1.61 %) (1.31 s.)	1273447.6 (1.67 %) (0.52 s.)	1295189.63 — (15.29 s.)

TAB. 5.11 – Résultats obtenus par les méthodes retenues sur les problèmes r20_120_40.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	659833.36 (7.16 %) (1.64 s.)	674568.21 (5.09 %) 8.88 s.)	704617.66 (0.86 %) (3.02 s.)	704422.93 (0.89 %) (1.38 s.)	710752.75 — (7.33 s.)
0.05_1.0	1058545.2 (13.32 %) (1.3 s.)	1058545.2 (13.32 %) 6.31 s.)	1198969.5 (1.82 %) (2.76 s.)	1208055.8 (1.08 %) (1.51 s.)	1221287.84 — (57.19 s.)
0.10_1.0	1438953.1 (17.52 %) (1.22 s.)	1438953.1 (17.52 %) 10.03 s.)	1670877.6 (4.23 %) (2.72 s.)	1719762 (1.43 %) (1.5 s.)	1744721.84 — (126.39 s.)
0.01_2.0	810339.3 (5.85 %) (3.13 s.)	810339.3 (5.85 %) 7.76 s.)	854736.24 (0.69 %) (4.61 s.)	853164.1 (0.87 %) (1.39 s.)	860718.62 — (17.72 s.)
0.05_2.0	1379423.7 (12.03 %) (1.44 s.)	1379423.7 (12.03 %) 13.53 s.)	1542892.1 (1.60 %) (2.92 s.)	1551765.7 (1.04 %) (1.46 s.)	1568086.93 — (67.92 s.)
0.10_2.0	1925452.3 (16.32 %) (1.29 s.)	1925452.3 (16.32 %) 13.04 s.)	2269721.5 (1.36 %) (2.79 s.)	2277081.1 (1.04 %) (1.53 s.)	2301044.88 — (147.49 s.)
0.01_8.0	1339610.3 (41.47 %) (4.05 s.)	1339610.3 (41.47 %) 46.55 s.)	2063859.8 (9.82 %) (5.63 s.)	2120990.3 (7.33 %) (1.48 s.)	2288772.70 — (26.02 s.)
0.05_8.0	2639574.6 (24.30 %) (3.5 s.)	2639574.6 (24.30 %) 16.02 s.)	3381574.6 (3.02 %) (4.93 s.)	3428551.6 (1.68 %) (1.62 s.)	3487232.10 — (28.26 s.)
0.10_8.0	3725632.9 (18.42 %) (2.32 s.)	3725632.9 (18.42 %) 12.22 s.)	4456441.5 (2.42 %) (3.93 s.)	4485870.9 (1.78 %) (1.62 s.)	4567169.63 — (45.85 s.)

TAB. 5.12 – Résultats obtenus par les méthodes retenues sur les problèmes r20_120_100.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	1488466.3 (8.44 %) (2.74 s.)	1488466.3 (8.44 %) 9.3 s.)	1612531.7 (0.81 %) (5.62 s.)	1614326.8 (0.70 %) (2.99 s.)	1625827.35 — (64.19 s.)
0.05_1.0	2821954.7 (13.63 %) (3 s.)	2821954.7 (13.63 %) 7.96 s.)	3228367.1 (1.19 %) (6.3 s.)	3230413.3 (1.12 %) (3.19 s.)	3267328.16 — (446.06 s.)
0.10_1.0	4001667.3 (18.90 %) (2.89 s.)	4001667.3 (18.90 %) 7.77 s.)	4875125.8 (1.20 %) (6.41 s.)	4865926.9 (1.39 %) (3.55 s.)	4934740.66 — (1041.53 s.)
0.01_2.0	1857473 (18.87 %) (2.54 s.)	1857473 (18.87 %) 62.81 s.)	2238043.2 (2.24 %) (5.73 s.)	2231792.4 (2.52 %) (2.91 s.)	2289516.73 — (128.71 s.)
0.05_2.0	3919931.4 (15.35 %) (2.73 s.)	3919931.4 (15.35 %) 14.52 s.)	4581254.9 (1.08 %) (5.93 s.)	4570212.8 (1.31 %) (3.21 s.)	4631291.62 — (538.47 s.)
0.10_2.0	5711094.1 (19.00 %) (2.81 s.)	5711094.1 (19.00 %) 9.57 s.)	6912392.9 (1.96 %) (6.1 s.)	6916763.8 (1.90 %) (3.5 s.)	7050795.55 — (579.27 s.)
0.01_8.0	3226907.9 (57.72 %) (2.49 s.)	3226907.9 (57.72 %) 71.91 s.)	6993125.4 (8.38 %) (6.24 s.)	7398958.4 (3.06 %) (4.05 s.)	7633214.62 — (57.41 s.)
0.05_8.0	7096399.3 (29.50 %) (3.05 s.)	7096399.3 (29.50 %) 31.62 s.)	9608832.8 (4.54 %) (6.73 s.)	9843243.9 (2.21 %) (4.05 s.)	10066039.07 — (72.72 s.)
0.10_8.0	9746988.7 (18.55 %) (2.69 s.)	9949579.4 (16.86 %) 34.01 s.)	11728715 (1.99 %) (6.53 s.)	11698202 (2.25 %) (4.3 s.)	11967768 — (59.39 s.)

TAB. 5.13 – Résultats obtenus par les méthodes retenues sur les problèmes r20_120_200.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	127130.53 (10.78 %) (0.67 s.)	132943.96 (6.70 %) 2.3 s.)	140144.64 (1.65 %) (1.34 s.)	141867.55 (0.44 %) (0.61 s.)	142498.68 — (0.80 s.)
0.05_1.0	207051.88 (17.53 %) (0.66 s.)	216002.23 (13.97 %) 4.8 s.)	242152.42 (3.55 %) (1.35 s.)	247220.78 (1.53 %) (0.69 s.)	251084.80 — (10.17 s.)
0.10_1.0	281799.99 (18.36 %) (0.69 s.)	283534.93 (17.86 %) 3.84 s.)	334768.73 (3.02 %) (1.42 s.)	340447.87 (1.37 %) (0.71 s.)	345207.44 — (25.20 s.)
0.01_2.0	131474.54 (11.35 %) (0.71 s.)	139445.7 (5.97 %) 2.67 s.)	146791.68 (1.02 %) (1.27 s.)	147508.2 (0.54 %) (0.61 s.)	148309.81 — (1.12 s.)
0.05_2.0	223839.46 (16.75 %) (0.68 s.)	227474.32 (15.40 %) 3.8 s.)	258495.78 (3.86 %) (1.4 s.)	265138.64 (1.39 %) (0.66 s.)	268892.08 — (16.03 s.)
0.10_2.0	312117.02 (17.66 %) (0.68 s.)	312117.02 (17.66 %) 2.27 s.)	371343.21 (2.03 %) (1.36 s.)	374378.31 (1.23 %) (0.71 s.)	379060.66 — (36.26 s.)
0.01_8.0	194007.78 (3.35 %) (0.89 s.)	196152.4 (2.28 %) 3.43 s.)	198901.18 (0.91 %) (1.54 s.)	200086.16 (0.32 %) (0.7 s.)	200739.49 — (6.86 s.)
0.05_8.0	387905.35 (8.19 %) (0.71 s.)	387905.35 (8.19 %) 2.18 s.)	416586.53 (1.40 %) (1.36 s.)	420035.61 (0.59 %) (0.67 s.)	422532.48 — (44.44 s.)
0.10_8.0	591831.46 (10.48 %) (0.81 s.)	591831.46 (10.48 %) 4.9 s.)	655655.35 (0.83 %) (1.49 s.)	656404.25 (0.71 %) (0.65 s.)	661150.23 — (76.66 s.)

TAB. 5.14 – Résultats obtenus par les méthodes retenues sur les problèmes r20_220_40.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	341955.5 (14.78 %) (2.16 s.)	365217.16 (8.98 %) 8.12 s.)	397199.52 (1.01 %) (3.79 s.)	398394.57 (0.71 %) (1.79 s.)	401272.05 — (11.15 s.)
0.05_1.0	568837.69 (20.13 %) (2.1 s.)	568837.69 (20.13 %) 5.18 s.)	672777.97 (5.54 %) (3.95 s.)	699831.38 (1.74 %) (1.98 s.)	712272.47 — (149.27 s.)
0.10_1.0	788238.94 (21.16 %) (2.1 s.)	788238.94 (21.16 %) 5.35 s.)	944628.11 (5.52 %) (3.94 s.)	977508.32 (2.23 %) (1.96 s.)	999818.09 — (402.55 s.)
0.01_2.0	379373.33 (12.50 %) (2.14 s.)	387505.16 (10.63 %) 8.93 s.)	427037.78 (1.51 %) (3.96 s.)	430177.86 (0.79 %) (1.71 s.)	433613.17 — (16.66 s.)
0.05_2.0	672603.59 (17.49 %) (2.12 s.)	672603.59 (17.49 %) 5.08 s.)	788201.86 (3.31 %) (4.11 s.)	803092.29 (1.49 %) (1.99 s.)	815243.29 — (235.99 s.)
0.10_2.0	951391.56 (18.46 %) (2.09 s.)	951391.56 (18.46 %) 5.05 s.)	1129036.8 (3.23 %) (3.92 s.)	1152147 (1.25 %) (1.98 s.)	1166815.45 — (646.58 s.)
0.01_8.0	609173.26 (6.74 %) (2.94 s.)	609173.26 (6.74 %) 7.83 s.)	646408.31 (1.04 %) (4.79 s.)	650098.48 (0.48 %) (1.79 s.)	653260.44 — (122.48 s.)
0.05_8.0	1349868.6 (12.52 %) (2.48 s.)	1349868.6 (12.52 %) 7.14 s.)	1524174.8 (1.23 %) (4.5 s.)	1531157.9 (0.78 %) (1.89 s.)	1543233.25 — (409.03 s.)
0.10_8.0	2076398.1 (15.67 %) (2.47 s.)	2076398.1 (15.67 %) 6.34 s.)	2415057.7 (1.92 %) (4.38 s.)	2432985.1 (1.19 %) (2.01 s.)	2462471.13 — (730.18 s.)

TAB. 5.15 – Résultats obtenus par les méthodes retenues sur les problèmes r20_220_100.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	836208.06 (15.88 %) (5.62 s.)	836208.06 (15.88 %) 13.17 s.)	967221.35 (2.70 %) (9.44 s.)	987851.91 (0.62 %) (3.85 s.)	994108.85 — (65.11 s.)
0.05_1.0	1522379.9 (19.50 %) (4.87 s.)	1522379.9 (19.50 %) 13.16 s.)	1815739.9 (3.99 %) (8.8 s.)	1865487.7 (1.36 %) (4.06 s.)	1891387.46 — (2638.35 s.)
0.10_1.0	2163983.6 (20.70 %) (4.9 s.)	2163983.6 (20.70 %) 13.21 s.)	2615832.2 (4.14 %) (9.02 s.)	2684163.8 (1.63 %) (4.16 s.)	2728867.20 — (4636.68 s.)
0.01_2.0	1014053 (10.90 %) (5.86 s.)	1014053 (10.90 %) 13.39 s.)	1114462.5 (2.08 %) (9.81 s.)	1129597.6 (0.75 %) (3.66 s.)	1138154.39 — (203.13 s.)
0.05_2.0	2037142.3 (14.94 %) (4.81 s.)	2037142.3 (14.94 %) 12.71 s.)	2344171.7 (2.13 %) (9.01 s.)	2367344.1 (1.16 %) (3.95 s.)	2395216.16 — (2417.65 s.)
0.10_2.0	3069638.2 (16.58 %) (5.52 s.)	3069638.2 (16.58 %) 14.04 s.)	3616057.5 (1.74 %) (9.84 s.)	3636380.3 (1.18 %) (4.28 s.)	3680106.97 — (6725.01 s.)
0.01_8.0	1821861.9 (19.94 %) (5.08 s.)	1821861.9 (19.94 %) 83.46 s.)	2238494.1 (1.64 %) (9.07 s.)	2230227.5 (2.00 %) (3.76 s.)	2275900.38 — (506.63 s.)
0.05_8.0	4377705 (21.05 %) (5 s.)	4377705 (21.05 %) 37.52 s.)	5419094 (2.27 %) (9.44 s.)	5444073.2 (1.82 %) (4.5 s.)	5545498.87 — (970.90 s.)
0.10_8.0	6652608.6 (23.34 %) (5.01 s.)	6652608.6 (23.34 %) 28.43 s.)	8230206.4 (5.16 %) (9.45 s.)	8487480.1 (2.20 %) (4.81 s.)	8678431.23 — (2594.35 s.)

TAB. 5.16 – Résultats obtenus par les méthodes retenues sur les problèmes r20_220_200.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	120360.9 (11.60 %) (0.93 s.)	126993.48 (6.73 %) 3.66 s.)	133297.46 (2.10 %) (1.67 s.)	134811.21 (0.99 %) (0.76 s.)	136161 — (1.21 s.)
0.05_1.0	187944.12 (18.29 %) (1.11 s.)	194589.63 (15.40 %) 5.1 s.)	212833.09 (7.47 %) (1.91 s.)	223076.22 (3.02 %) (0.81 s.)	230036.78 — (22.14 s.)
0.10_1.0	251634.47 (19.08 %) (1.13 s.)	253818.13 (18.38 %) 4.83 s.)	297251.22 (4.41 %) (1.96 s.)	301626.49 (3.01 %) (0.84 s.)	310993.94 — (39.30 s.)
0.01_2.0	120676.67 (12.60 %) (1.04 s.)	129842.32 (5.96 %) 4.01 s.)	134653.69 (2.48 %) (1.8 s.)	136721.6 (0.98 %) (0.77 s.)	138085.81 — (1.78 s.)
0.05_2.0	195752.38 (17.01 %) (1.08 s.)	199411.99 (15.46 %) 6.9 s.)	222949.05 (5.48 %) (1.92 s.)	229900.78 (2.53 %) (0.83 s.)	235891.22 — (31.98 s.)
0.10_2.0	264926.01 (17.89 %) (1.14 s.)	264926.01 (17.89 %) 2.71 s.)	311725.78 (3.38 %) (2.01 s.)	311923.87 (3.32 %) (0.89 s.)	322655.77 — (53.36 s.)
0.01_8.0	154452.48 (4.41 %) (1.13 s.)	156649.2 (3.05 %) 4.35 s.)	160342.39 (0.76 %) (1.85 s.)	161014.46 (0.34 %) (0.72 s.)	161579.31 — (4.72 s.)
0.05_8.0	291540.67 (10.82 %) (1.08 s.)	296380.75 (9.34 %) 6.18 s.)	323343.51 (1.09 %) (1.85 s.)	324097.12 (0.86 %) (0.77 s.)	326930.46 — (72.84 s.)
0.10_8.0	421494.81 (14.55 %) (0.99 s.)	421494.81 (14.55 %) 3.49 s.)	484303.21 (1.82 %) (1.87 s.)	484829.25 (1.71 %) (0.8 s.)	493283.10 — (149.51 s.)

TAB. 5.17 – Résultats obtenus par les méthodes retenues sur les problèmes r20_320_40.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	303912.75 (13.77 %) (3.15 s.)	315258.5 (10.55 %) 11.41 s.)	339369.27 (3.71 %) (5.39 s.)	347858.11 (1.30 %) (2.22 s.)	352470.68 — (19.46 s.)
0.05_1.0	494916.43 (19.47 %) (3.34 s.)	494916.43 (19.47 %) 7.93 s.)	578478.85 (5.88 %) (5.58 s.)	603865.69 (1.75 %) (2.36 s.)	614649.06 — (243.22 s.)
0.10_1.0	663584.48 (21.18 %) (3.42 s.)	663584.48 (21.18 %) 8.19 s.)	800422.45 (4.92 %) (5.54 s.)	826488.98 (1.83 %) (2.48 s.)	841929.57 — (609.97 s.)
0.01_2.0	317485.25 (13.64 %) (3.18 s.)	320312.61 (12.87 %) 11.87 s.)	356045.8 (3.15 %) (5.33 s.)	363506.62 (1.12 %) (2.15 s.)	367643.94 — (33.83 s.)
0.05_2.0	548219.66 (19.21 %) (3.32 s.)	548219.66 (19.21 %) 7.68 s.)	649437.31 (4.30 %) (5.65 s.)	668663.97 (1.47 %) (2.42 s.)	678650.29 — (463.86 s.)
0.10_2.0	760933.38 (20.10 %) (3.22 s.)	760933.38 (20.10 %) 7.72 s.)	931595.24 (2.18 %) (5.67 s.)	936656.01 (1.65 %) (2.43 s.)	952437.17 — (1033.08 s.)
0.01_8.0	443240.08 (9.62 %) (3.33 s.)	443240.08 (9.62 %) 8.96 s.)	486262.64 (0.85 %) (5.49 s.)	485952.17 (0.92 %) (2.26 s.)	490467.96 — (133.80 s.)
0.05_8.0	946042.45 (11.26 %) (3.27 s.)	946042.45 (11.26 %) 8.38 s.)	1054449.3 (1.09 %) (5.42 s.)	1057568.6 (0.80 %) (2.21 s.)	1066109.72 — (857.63 s.)
0.10_8.0	1455191.9 (14.44 %) (3.42 s.)	1455191.9 (14.44 %) 8.32 s.)	1674004.8 (1.58 %) (5.69 s.)	1685117.3 (0.92 %) (2.38 s.)	1700915.13 — (1980.86 s.)

TAB. 5.18 – Résultats obtenus par les méthodes retenues sur les problèmes r20_320_100.

F_C	DA (P_1)	UPD libre	DA \Rightarrow KB	KB	CPLEX
0.01_1.0	665958.76 (18.29 %) (7.9 s.)	665958.76 (18.29 %) 19.42 s.)	798717.78 (2.00 %) (12.37 s.)	806185.75 (1.09 %) (4.59 s.)	815086.03 — (245.61 s.)
0.05_1.0	1164718.1 (21.82 %) (7.37 s.)	1164718.1 (21.82 %) 19.51 s.)	1421237.2 (4.61 %) (12.23 s.)	1464280.2 (1.72 %) (5.23 s.)	1489978.12 — (4799.79 s.)
0.10_1.0	1601224.4 (22.33 %) (7.25 s.)	1601224.4 (22.33 %) 18.21 s.)	1945184.3 (5.64 %) (12.17 s.)	2022443.1 (1.90 %) (5.23 s.)	2061668.57 — (8381.30 s.)
0.01_2.0	772869.29 (14.42 %) (7.87 s.)	772869.29 (14.42 %) 19.9 s.)	879906.63 (2.57 %) (12.55 s.)	894935.52 (0.91 %) (4.64 s.)	903176.39 — (384.59 s.)
0.05_2.0	1461128.2 (16.77 %) (7.64 s.)	1461128.2 (16.77 %) 20.11 s.)	1698812.4 (3.23 %) (12.5 s.)	1732680.5 (1.31 %) (4.76 s.)	1755696.49 — (6460.60 s.)
0.10_2.0	2132718.3 (17.16 %) (7.05 s.)	2132718.3 (17.16 %) 18.22 s.)	2518388.9 (2.18 %) (12.24 s.)	2542299.1 (1.25 %) (4.99 s.)	2574608.26 — (18660.55 s.)
0.01_8.0	1287187 (11.31 %) (8.2 s.)	1287187 (11.31 %) 21.92 s.)	1442080.9 (0.64 %) (12.81 s.)	1441101.9 (0.70 %) (4.49 s.)	1451377.47 — (1502.98 s.)
0.05_8.0	3114215.7 (16.63 %) (9.12 s.)	3114215.7 (16.63 %) 24.32 s.)	3707282.2 (0.75 %) (14.94 s.)	3705862.9 (0.79 %) (4.89 s.)	3735602.43 — (9678.37 s.)
0.10_8.0	4834935.3 (20.58 %) (8.92 s.)	4834935.3 (20.58 %) 29.76 s.)	6012815.2 (1.24 %) (14.94 s.)	6020937.1 (1.10 %) (5.67 s.)	6088503.71 — (13980.05 s.)

TAB. 5.19 – Résultats obtenus par les méthodes retenues sur les problèmes r20_320_200.