

Table des matières

Résumé	iii
Abstract	vii
Table des matières	x
Liste des tableaux	xii
Liste des figures	xv
Remerciements	xix
1 Introduction générale	1
1.1 Modélisation d'un système de gestion des ressources humaines	2
1.2 Décisions en gestion des horaires	3
2 Construction de quarts multi-jours dans un contexte multidépartement	8
2.1 Problématique de la thèse	8
2.2 Méthodologie et contributions	14
2.3 Générateur de données	19
3 Revue de la littérature	25
3.1 Problèmes de construction d'horaires de travail	29
3.2 Construction d'horaires dans un contexte monodépartement	37
3.3 Construction d'horaires dans un contexte multidépartement	50
4 Modélisation du problème	55
4.1 Modèle de construction d'horaires	55
4.2 Modèle de construction de quarts	57
4.3 Modèle de construction de blocs	59
4.4 Variante du modèle de construction de blocs	72
4.5 Modèle réduit de construction de blocs	73
4.6 Analyse comparative des formulations proposées	77
5 Approche de résolution basée sur la génération de colonnes	95
5.1 Notions préliminaires	95
5.2 Génération de colonnes suivie d'un Branch and Bound	99
5.3 Analyse de performance de la procédure de génération de colonnes	102
5.4 Procédure basée sur la génération de colonnes	102

6	Approche de résolution prétransfert	112
6.1	Problème agrégé	112
6.2	Problème désagrégé	118
6.3	Analyse de performance de l'approche de résolution prétransfert	133
7	Approche de résolution post-transfert	161
7.1	Problème agrégé séparable	161
7.2	Problème désagrégé	163
7.3	Analyse de performance de l'approche de résolution post-transfert	169
	Conclusion	186
A	Annexes	191
A.1	Choix de la méthode d'agrégation	191
A.2	Choix de la méthode de désagrégation	194
A.3	Comparaison entre les heuristiques de prétransfert et de post-transfert	196
	Bibliographie	210

Liste des tableaux

2.1	Complexité du jeu de données	21
4.1	Instances du JEU2 résolues à l’optimalité par les deux modèles implicites MCB1i et MCB1j : # d’instances et écarts des temps de calcul en secondes	84
4.2	Instances du JEU2 résolues à l’optimalité par le modèle MCB1i uniquement : # d’instances, temps de calcul du modèle MCB1i en secondes et gaps d’optimalité du modèle MCB1j en %	84
4.3	Instances du JEU2 non résolues à l’optimalité par les deux modèles MCB1i et MCB1j : # d’instances et gaps d’optimalité du modèle MCB1i puis MCB1j en %	85
4.4	Taille des modèles MCB11 et MCQ pour les classes d’instances du JEU2	86
4.5	Résultats obtenus par les modèles MCB11 et MCQ pour les instances du JEU2	87
4.6	Instances du JEU3 résolues à l’optimalité par les deux modèles implicites MCB2i et MCB2j : # d’instances et écarts des temps de calcul en secondes	90
4.7	Instances du JEU3 résolues à l’optimalité par le modèle MCB2i uniquement : # d’instances, temps de calcul du modèle MCB2i en secondes et gaps d’optimalité du modèle MCB2j en %	91
4.8	Instances du JEU3 non résolues à l’optimalité par les deux modèles MCB2i et MCB2j : # d’instances et gaps d’optimalité du modèle MCB2i puis MCB2j en %	91
4.9	Taille des modèles MCB11, MCB24 et MCQ pour les classes d’instances du JEU3	92
4.10	Résultats obtenus par les modèles MCB11, MCB24 et MCQ pour les instances du JEU3	93
5.1	Combinaisons de la procédure basée sur la génération de colonnes	103
5.2	Qualité relative des solutions obtenues par la procédure de génération de colonnes suivie d’un Branch and Bound	104
5.3	Temps de calcul en secondes de la procédure de génération de colonnes suivie d’un Branch and Bound	106
5.4	Taille des problèmes en nombres entiers résolus par GenColBB et par BB	108
5.5	Performances de la procédure basée sur la génération de colonnes et la procédure de séparation et d’évaluation progressive	109
6.1	Méthodes d’agrégation des données temporelles	134
6.2	Agrégation des réglementations	138
6.3	Taille des problèmes utilisés pour le choix de la méthode d’agrégation	139
6.4	Comparaison des différentes méthodes d’agrégation sur la base du coût du meilleur calendrier trouvé dans le voisinage de la solution agrégée	142
6.5	Comparaison des différentes méthodes d’agrégation en se basant sur le temps de calcul requis de l’étape d’agrégation	144

6.6	Configurations testées pour la résolution du problème désagrégé P restreint à \tilde{Q}	148
6.7	Comparaison des différentes méthodes de désagrégation en se basant sur la qualité des solutions obtenues	152
6.8	Répartition des temps de calcul des méthodes de désagrégation	154
6.9	Comparaison des procédures GenColBB et DecomPET pour le JEU1	160
7.1	Taille du problème désagrégé P restreint à \tilde{Q} pour les trois méthodes de désagrégation post-transfert	171
7.2	Coûts des calendriers désagrégés identifiés par les trois méthodes de désagrégation post-transfert	172
7.3	Répartition des instances en % selon la qualité relative des solutions obtenues par les trois méthodes de désagrégation post-transfert	173
7.4	Temps requis pour compléter l'étape de désagrégation par les trois méthodes de désagrégation post-transfert	174
7.5	Répartition des instances résolues avec une qualité acceptable en % selon le temps total requis pour compléter l'étape de désagrégation par les trois méthodes de l'heuristique post-transfert	176
7.6	Taille du problème agrégé pour les heuristiques de prétransfert et de post-transfert	178
7.7	Taille du problème désagrégé pour les heuristiques de prétransfert et de post-transfert	181
7.8	Comparaison globale entre les deux heuristiques de pré- et de post-transfert .	183
A.1	Gaps de CPLEX du problème agrégé \bar{P} restreint aux quarts sélectionnés par la génération de colonnes	192
A.2	Temps de calcul requis du problème désagrégé P restreint au voisinage $V(\bar{C})$ pour les six méthodes d'agrégation testées	193
A.3	Gaps de CPLEX du problème désagrégé P restreint aux quarts voisins sélectionnés par la génération de colonnes	194
A.4	Nombre moyen de variables de décision des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$	196
A.5	Nombre minimal de variables de décision des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$	197
A.6	Nombre maximal de variables de décision des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$	198
A.7	Coûts des calendriers calculés par les différentes méthodes de désagrégation	199
A.8	Gaps moyens de CPLEX des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$ en pourcentage	200
A.9	Gaps minimaux de CPLEX des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$ en pourcentage	201
A.10	Gaps maximaux de CPLEX des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$ en pourcentage	202
A.11	Nombre total d'heures transférées des calendriers calculés par les différentes méthodes de désagrégation	203
A.12	Temps de calcul des méthodes de désagrégation en secondes	204
A.13	Temps d'énumération de l'ensemble \tilde{Q} par les méthodes de désagrégation en secondes	205

A.14 Temps de la procédure d'élimination des transferts cycliques par les méthodes de désagrégation en secondes	206
A.15 Temps total de résolution de tous les sous-problèmes $P(w, d)$ définis par les méthodes de désagrégation en secondes	207
A.16 Taille des problèmes agrégés de prétransfert $\bar{P}_{Pré}$ et de post-transfert \bar{P}_{Post}	208
A.17 Résultats obtenus par la résolution des problèmes agrégés de prétransfert $\bar{P}_{Pré}$ et de post-transfert \bar{P}_{Post}	209

Liste des figures

1.1	Modèle générique de gestion des ressources humaines	3
2.1	Cadre conceptuel du problème traité	15
2.2	Courbes de demande de deux départements de l'instance D20E1000P3	23
2.3	Profils de demande d'un département	23
2.4	Arborescence des instances	24
4.1	Graphe de transport associé à l'exemple A	67
4.2	Arborescence des jeux de données	78
4.3	Taille des modèles implicites de construction de blocs pour les instances du JEU2	82
4.4	Effet de l'utilisation des blocs fictifs sur le modèle MCB1	83
4.5	Qualifications internes et externes des employés	88
4.6	Taille des modèles implicites réduits de construction de blocs pour les instances du JEU3	89
4.7	Effet de l'utilisation des blocs fictifs sur le modèle MCB2	90
5.1	Complexité des calculs de la procédure basée sur la génération de colonnes	107
6.1	Information sur les blocs externes agrégés	120
6.2	Graphe de transfert G_I^{tr} relatif à l'horizon de planification correspondant à l'exemple de la Figure 6.1	121
6.3	Construction du premier graphe de transfert	122
6.4	Tri des départements	125
6.5	Évaluation d'une méthode d'agrégation	135
6.6	Effet du profil et de la méthode d'agrégation sur la précision des courbes de demande	136
6.7	Indicateurs d'écart par instance et par méthode d'agrégation	137
6.8	Durées des blocs de début et de fin pour les quarts multidépartements	141
6.9	Évaluation d'une méthode de désagrégation	146
6.10	Taille du problème désagrégé P restreint à l'ensemble \tilde{Q}	151
6.11	Nombre moyen de variables de décision des sous-problèmes $P(w, d)$ restreints aux sous-ensembles $\tilde{Q}(w, d)$	151
6.12	Comparaison des deux procédures GenColBB et DecompET pour la résolution du problème désagrégé P	156
7.1	Méthodes de désagrégation pour l'heuristique de post-transfert	164

7.2	Résultats du problème agrégé pour les heuristiques de prétransfert et de post-transfert	179
7.3	Résultats du problème désagrégé pour les heuristiques de prétransfert et de post-transfert	182

Rapport-Gratuit.com

À ma mère
À la mémoire de mon père
À mon mari, ma soeur et mon frère

J'ai raté 9000 tirs dans ma carrière. J'ai perdu presque 300 matchs. 26 fois, on m'a fait confiance pour prendre le tir de la victoire et j'ai raté. J'ai échoué encore et encore et encore dans ma vie. Et c'est pourquoi j'ai réussi.

Michael Jordan

Remerciements

Je tiens à exprimer mes plus sincères remerciements à Mme Monia Rekik qui fut pour moi une directrice de recherche attentive et disponible tout au long de la thèse. Sa présence tant au niveau humain qu'au niveau professionnel m'a permis de progresser dans mes recherches. J'ai connu Mme Monia depuis mon projet de fin d'études. Elle m'a encadré dans mon cursus d'ingénieur, de mastère et de doctorat. Durant toutes ces années, à travers ses compétences, sa rigueur et sa clairvoyance, elle était une vraie mentor pour moi. C'est grâce à ses encouragements et son implication que j'ai pu avancer dans mes travaux et améliorer mes compétences de chercheuse. J'ai eu beaucoup de chance d'avoir rencontré madame Monia et de travailler avec elle.

Je tiens également à exprimer toute ma gratitude à M. François Soumis, codirecteur de ma thèse. Je tiens à exprimer tout mon respect et ma reconnaissance pour le rôle qu'il a joué dans l'élaboration de cette thèse. Par ses conseils, ses idées, son expérience et sa modestie, il est un exemple à suivre et une source d'inspiration pour moi.

J'adresse mes remerciements à M. Guy Desaulniers qui a contribué de près à la réalisation de cette thèse. Son implication, sa compétence, sa disponibilité et son aide précieuse m'ont été d'une grande utilité.

C'était un honneur et un plaisir de travailler avec vous trois.

Je remercie également Pr. Favez Boctor, Pr. Jean François-Cordeau et Pr. Louis-Martin Rousseau d'avoir accepté d'être membres de mon jury.

Mes remerciements à toute l'équipe Kronos Inc. pour son soutien financier et son implication dans la recherche et le développement.

Merci à tout le département Opérations et Systèmes de Décision, et plus spécifiquement M. Bernard Lamond et Mireille.

Merci à toute l'équipe du CIRRELT : Pierre, Louise, Alexis et Martine.

Mes remerciements à tous mes enseignants en Tunisie et au Canada. Une mention spéciale à Mme Atidel Ben Hadj Alouane et M. Chiheb Bouden.

Merci à tous mes amis. Un merci spécial à mes chers Ana, Salma, Lilia, Intissar et Luckny.

Merci à toute l'équipe de l'AÉDAUL. C'était un plaisir de travailler avec vous.

Merci à ma belle famille : Mouna, Issam, Alae, Ali, Hiba, Simou et mes beaux parents pour leurs encouragements.

Merci à ma chère sœur Manel et son mari Ahmed, mon neveu Adam qui était bébé lorsque j'ai commencé la thèse et ma nièce Éya qui est née au cours de la thèse, et à mon cher frère Bilel pour leurs encouragements.

Merci à mon mari qui a enduré tous les inconvénients liés à la vie d'un thésard. Merci Fayçal pour ta patience, ta compréhension, merci pour les fins de semaine travaillées, pour les sauts d'humeur et le stress. Merci pour le soutien et la sécurité que tu m'a offerts durant ma thèse.

Une pensée à toi. J'espère que tu seras fier de ta fille si tu nous regardes de l'au-delà.

Le plus grand merci à ma mère. Dans les moments les plus difficiles, à travers tes prières et tes mots, je trouvais la force de continuer. Parce que tu es une mère extraordinaire, je te dédis cette thèse.

Merci à Dieu.

Chapitre 1

Introduction générale

L'humain est la ressource de production centrale au sein de toute organisation. En dépit des avancées en robotique et de l'essor de l'automatisation, la composante humaine se distingue toujours par sa capacité à communiquer, à faire face à l'imprévu et à innover dans la conception et la gestion des systèmes de production. Un système de gestion de production nécessite l'utilisation de deux types de ressources : matérielles et humaines. Baptiste et collab. (2005) distinguent entre les approches *imbriquées* et *intégrées* de gestion de ces ressources. Les approches imbriquées séparent entre le déploiement matériel et humain en accordant la priorité à l'un des deux selon les objectifs de l'organisation. Dans un premier niveau, on commence par la gestion de la ressource la plus prioritaire. Dans le deuxième niveau, la ressource restante est allouée de façon à respecter les bornes définies par le niveau décisionnel supérieur. Les approches intégrées combinent les deux ressources dans un même modèle de gestion. En industrie manufacturière, on accorde souvent la priorité aux machines et on déploie les effectifs en conséquence. Dans le secteur tertiaire, lorsque le personnel constitue le facteur clé de production tel est le cas des supermarchés, des banques et des centres d'appel, il est prioritaire. Au contraire, lorsque la firme de service fait appel à des équipements coûteux en investissement et en fonctionnement comme les compagnies aériennes et les hôpitaux, les décisions de gestion des ressources matérielles sont prises en premier. Les approches intégrées sont encore limitées par la complexité des problèmes obtenus.

La construction des horaires du personnel est une partie intégrante de la gestion des ressources humaines. Elle permet à une organisation de construire un ensemble optimisé d'horaires de travail de ses employés. Cet ensemble doit permettre à la fois de répondre aux attentes des dirigeants et de respecter les clauses de la convention collective. Du point de vue des chefs d'entreprise, les horaires doivent permettre de satisfaire le besoin en main d'œuvre au moindre coût. Alors que du point de vue des salariés, ces horaires doivent permettre de respecter la qualité de travail et de vie. Face à ces objectifs conflictuels, le défi est de trouver le meilleur compromis entre les deux. Relever ce défi permet de générer des gains considérables pour les organisations.

L'introduction de l'horaire *flexible*, la *polyvalence* des employés, le développement du secteur de services, la complexité des conventions collectives et la conciliation travail et vie privée rendent la construction d'horaires une fonction à la fois complexe et à valeur ajoutée. De nos jours, l'offre des logiciels de gestion des ressources humaines est diversifiée et le besoin en algorithmes d'optimisation performants ne cesse de s'exprimer par les firmes qui proposent des solutions de gestion aux entreprises. Les modules d'optimisation ont recours à des techniques performantes telles que la programmation mathématique, la simulation et les heuristiques.

1.1 Modélisation d'un système de gestion des ressources humaines

Baptiste et collab. (2005) proposent un modèle générique d'un système de gestion de production que nous présentons dans cette section. Nous interprétons le schéma proposé par les auteurs au sens des ressources humaines en excluant les ressources matérielles. Ce modèle fait intervenir l'ensemble des intrants suivants : la demande, les données ressources et les gammes. La *demande* correspond à la quantité de travail à réaliser sur un horizon de planification. Les *données ressources* constituent la capacité de l'organisation en effectifs. Les *gammes* déterminent la quantité requise des ressources mobilisées sur des périodes de temps pour réaliser les activités de production. En considérant ces intrants, la décision aura comme extrants : un plan de production, une modification éventuelle des ressources et un plan de charge. Le *plan de production* exprime les quantités à produire en prenant compte des ressources déployées pour accomplir ce travail. Lorsque les objectifs de production ne sont pas atteints, des correctifs peuvent être appliqués en apportant des *modifications des données ressources* tel que l'embauche et la formation. Le *plan de charge* sert à décrire la consommation des ressources en effectifs durant l'horizon. La Figure 1.1 schématise le modèle décrit et focalise sur l'ensemble des composants considérés dans le cadre de cette thèse.

La gestion des ressources humaines fait intervenir trois types de décision selon l'étendue de l'horizon de planification et le niveau d'agrégation : les décisions stratégiques, tactiques et opérationnelles. À long terme, l'organisation doit planifier sa capacité globale en effectifs. Des décisions stratégiques sont prises par la direction portant sur la détermination de la taille et la composition de la main d'œuvre. Il s'agit essentiellement de définir la politique générale de recrutement, de promotion et de formation. À moyen terme, les cadres de l'organisation doivent prendre des décisions tactiques contraintes par le dimensionnement de la capacité stratégique. L'objectif est de choisir les options d'ajustement de la capacité aux variations de l'activité de l'organisation au moyen de certains recours tels que les heures supplémentaires et l'emploi de la main d'œuvre partielle. À court terme, les problématiques qui se posent sont principalement la gestion des horaires du personnel et l'ordonnancement de la production.

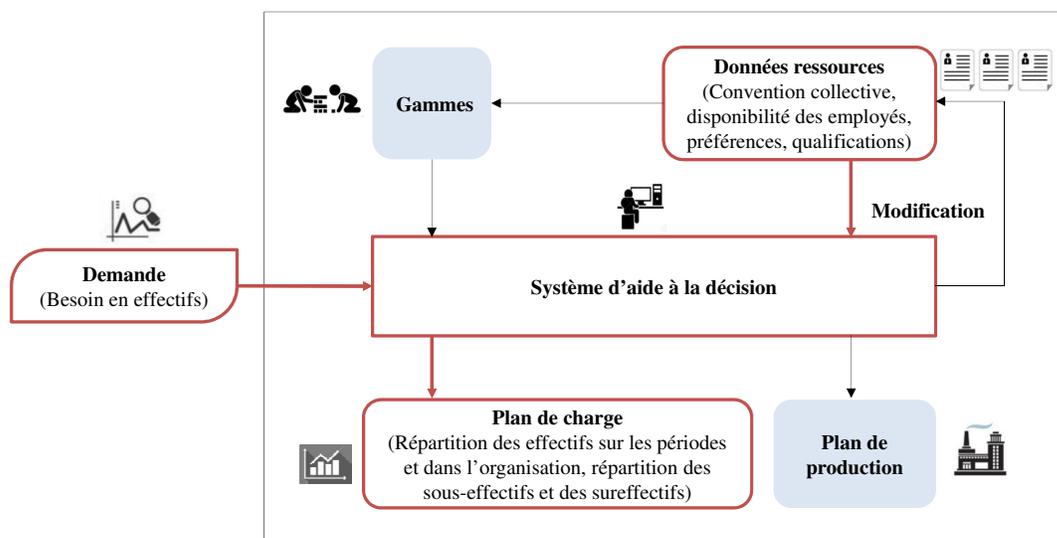


Figure 1.1 – Modèle générique de gestion des ressources humaines

Lorsque l’approche de gestion est imbriquée, ces deux problèmes sont traités séparément alors que lorsque l’approche est intégrée, l’ordonnancement et la gestion des horaires se font simultanément.

1.2 Décisions en gestion des horaires

Dans cette section, nous décrivons les problèmes de gestion d’horaires d’une façon sommaire. Nous détaillons davantage, dans le Chapitre 3, l’état de l’art dans le domaine. Tous les problèmes de construction d’horaires sont définis sur un *horizon de planification* divisé en des intervalles de temps généralement de durées égales. Cette durée correspond à la *période de discrétisation*. Pour chaque période de l’horizon, on a estimé le nombre d’employés minimal requis afin d’assurer un niveau de service satisfaisant. L’ensemble de ces données forment ce qu’on désigne par une *courbe de demande*. On distingue entre deux cas : la demande *cyclique* et *acyclique*. Une demande cyclique est une demande périodique qui se répète durant l’horizon de planification à toutes les semaines par exemple. Une demande acyclique fluctue sans un patron reconnaissable.

Dans les environnements de travail *figés*, tel est le cas des services administratifs, l’heure d’ouverture et de fermeture des bureaux correspond aux heures de début et de fin de travail des employés. Les fins de semaine, l’activité de l’organisation est interrompue et les employés ne travaillent pas. Lorsque l’organisation est en activité sept jours sur sept ou 24 heures sur 24, lorsque la demande fluctue considérablement durant la journée d’opération, on parle d’environnements *flexibles*. Dans les hôpitaux, les centres d’appel, la restauration, les supermarchés et d’autres firmes de service, l’horaire flexible permet un ajustement des

niveaux de la main d'œuvre à la demande.

La construction d'horaires est un processus de planification qui fait intervenir différents niveaux décisionnels. Ernst et collab. (2004a,b) décomposent le problème global en plusieurs niveaux et classent les travaux selon le ou les niveaux qu'ils traitent. Le problème global se compose principalement de trois niveaux : la modélisation de la demande, la construction des horaires anonymes et l'affectation des horaires aux employés.

1.2.1 Modélisation de la demande

La modélisation de la demande « *Demand modelling* » consiste à prévoir le besoin de l'organisation en effectifs. Ernst et collab. (2004b) distinguent trois types de demande : la demande en tâches, en activités et en quarts. Lorsque le travail est non interruptible, il s'agit de tâches. En transport aérien, un pilote d'avion qui commence un vol doit nécessairement le terminer. Il ne peut ni quitter temporairement le vol pour le reprendre ultérieurement, ni être remplacé par un autre pilote au cours du vol. En transport routier, un chauffeur n'est autorisé à quitter son travail que dans des points de relève bien déterminés. Dans une firme de service, l'employé doit terminer la prestation avec son client pour assurer la continuité du service. Une tâche est généralement définie par une heure de début, une durée, une fenêtre de temps durant laquelle la tâche doit être complétée et les compétences requises pour son exécution. Dans ce cas, on parle de « *task based demand* ». Lorsque le travail est interruptible, un employé peut interrompre l'activité qu'il est en train d'exécuter pour prendre sa pause ou pour quitter à la fin de son quart. Il peut soit reprendre cette activité ultérieurement, soit être remplacé par un autre employé. Dans ce cas, la demande est exprimée en un nombre minimal d'employés requis pour maintenir un niveau de service acceptable. On distingue entre deux cas : la demande exprimée en activités « *flexible demand* » et la demande exprimée en nombre de quarts « *shift based demand* ». Le premier cas est rencontré dans des firmes telles que les centres d'appel, les services de police et le personnel au sol dans les aéroports. Le deuxième cas est appliqué par exemple pour les infirmières et les ambulanciers. Dans le cadre de cette thèse, nous traitons uniquement le cas du travail interruptible lorsque la demande est exprimée en activités.

1.2.2 Construction des horaires anonymes

La construction des horaires anonymes « *Line of work construction* » se fait généralement en minimisant le coût de la main d'œuvre afin de satisfaire la demande prévue tout en respectant les réglementations de travail. L'horaire est dit *anonyme* tant qu'il n'est affecté à aucun employé spécifique de l'organisation. Une fois qu'il est affecté à un employé, il est dit *personnalisé*. Dans le cas d'un travail interruptible, on désigne par « *Tour scheduling* » la construction des horaires anonymes ou encore la construction simultanée de cycles et de quarts anonymes. Ce problème fait appel à deux niveaux décisionnels plus ou moins indépendants. Le

premier niveau est la construction de cycles anonymes « *Days-off scheduling* ». Il consiste à spécifier des séquences de jours de travail et de repos durant un horizon de planification, généralement de quelques semaines. Le deuxième niveau est la construction de quarts anonymes « *Shift scheduling* ». Il consiste à spécifier le début et la durée de travail durant un jour d'opération, et possiblement la position d'une ou de plusieurs pauses à l'intérieur de l'intervalle de travail. Baker (1976) était le premier à classer les travaux de construction d'horaires en trois catégories : la construction de quarts, la construction de cycles et la construction simultanée de cycles et de quarts. Dans ce qui suit, nous présentons chacun de ces problèmes.

1.2.3 Construction des cycles de travail

La construction de cycles consiste à spécifier les jours de travail et de repos. L'horizon de planification est généralement de quelques semaines discrétisé en des jours. Plusieurs semaines ou mois à l'avance, le nombre d'employés requis par jour est estimé et les cycles sont construits. Les employés peuvent planifier leur vie en conséquence. Les règles de travail les plus typiques qui permettent de définir l'ensemble admissible des cycles sont le nombre minimal et maximal de jours de travail par semaine et la proportion minimale de fins de semaine non travaillées par mois. Chaque cycle possède un coût comprenant le salaire et probablement des compensations pour les atteintes à la qualité de vie des employés. Ces compensations monétaires servent à dédommager un employé lorsqu'il reçoit un cycle de mauvaise qualité. Par exemple, l'employé travaille deux fins de semaine consécutives durant le mois alors qu'il devrait avoir au minimum une fin de semaine sur deux non travaillée. L'objectif du problème est de minimiser le coût total des cycles sélectionnés par le planificateur sous les contraintes de la convention collective et les contraintes de demande.

1.2.4 Construction des quarts de travail

La construction de quarts consiste à spécifier les heures de travail et de pause durant chaque jour travaillé. On distingue entre l'horizon de planification *continu* et *discontinu*. Lorsque les quarts ne peuvent pas chevaucher deux jours consécutifs, l'horizon est dit discontinu. Le problème de construction de quarts est résolu pour chaque jour séparément. Dans le cas contraire, le problème doit être résolu sur plusieurs jours à la fois et doit donc être défini sur un horizon *multi-jour*. Dans les deux cas, l'horizon de planification est généralement discrétisé en des périodes de 15, 30 ou 60 minutes selon la demande. Les règles de travail spécifient l'ensemble d'heures de début et de durées de travail journalier admissibles considérées par l'organisation. De plus, les pauses sont placées à l'intérieur des quarts d'une manière à respecter certaines règles. Des quarts longs peuvent contenir une pause de repas et deux pauses de café alors qu'un quart court peut ne contenir que la pause de repas. Le coût d'un quart est généralement proportionnel au nombre d'heures travaillées. L'objectif du problème est de minimiser le coût total des quarts sélectionnés par le planificateur sous les contraintes de la convention collective et les contraintes de demande.

1.2.5 Construction simultanée de cycles et de quarts de travail

Lorsqu'il s'agit d'un horizon continu, l'intégration de la construction de cycles et de quarts devient pertinente. Dans ce cas, le problème est défini sur un horizon multi-jour généralement discrétisé en des périodes de 15 ou 30 minutes. Il s'agit de spécifier à la fois les jours de travail et de repos, et les heures de travail et de pause durant chaque jour travaillé. L'objectif le plus couramment utilisé consiste à minimiser le coût total des séquences de quarts sélectionnées par le planificateur sous les contraintes d'admissibilité des cycles et des quarts et les contraintes de demande.

1.2.6 Affectation des horaires aux employés

L'affectation des horaires aux employés, « *Staff assignment* », se fait généralement en maximisant les préférences des employés pour les horaires sous certaines contraintes relatives à leurs disponibilités, leurs compétences, leurs séniorités, etc. La résolution de ce niveau décisionnel permet d'obtenir un ensemble d'horaires personnalisés. Lorsque la construction simultanée de cycles et de quarts est effectuée, on obtient un ensemble de cycles remplis par des quarts anonymes. Dans ce cas, ce niveau décisionnel se limite à une simple affectation de cet ensemble aux employés. Lorsque la construction de cycles et de quarts est effectuée séparément, on peut commencer par affecter les cycles anonymes aux employés. Ensuite, on passe à affecter les quarts aux jours travaillés de chaque employé. Les contraintes, souvent rencontrées, sont la durée de repos minimale entre deux quarts travaillés consécutivement, la durée minimale et maximale travaillée par employé et les patrons de changement de relève jour, soir et nuit. Par exemple, un employé affecté à un quart de type jour, ne peut passer qu'à un quart de type soir le jour suivant afin de respecter l'horloge biologique de cet employé.

1.2.7 Affectation des activités aux horaires

Un autre problème qui se pose est celui de l'affectation des activités aux horaires, « *Task assignment* ». Ce problème se pose uniquement dans un contexte multiactivité, c'est-à-dire que le travail est composé de plusieurs activités. Une activité est un type de travail interruptible. L'horaire devient alors entièrement défini, en spécifiant, en plus des séquences de jours travaillés et de repos et des intervalles de travail et de pause, la séquence d'activités qui le compose. Lorsqu'un employé change d'une activité à une autre, il s'agit d'une *transition*. Dans le cas où les transitions correspondent à une perturbation du travail et une perte de productivité, on impose une durée minimale de périodes travaillées consécutivement dans l'accomplissement d'une activité donnée. Ceci permet alors d'éviter des transitions fréquentes entre les activités durant un quart travaillé. Une durée maximale est également définie pour chaque activité afin d'imposer un plafond sur le nombre de périodes travaillées dans l'exécution d'une activité stressante. La demande est estimée pour chaque activité de l'organisation. Il s'agit de minimiser, par exemple, le nombre de transitions sous les contraintes d'admis-

sibilité des séquences d'activités composant les quarts et les contraintes de satisfaction des courbes de demande des activités. Dans un contexte multiactivité, on distingue trois types d'effectifs : *homogène*, *hétérogène* et *hiérarchique*. L'effectif est dit homogène lorsque tous les employés de l'organisation peuvent effectuer toutes les activités. Il est dit hétérogène lorsque les employés possèdent des *qualifications* pour les activités. Dans ce cas, un employé est dit qualifié pour une activité s'il possède les compétences requises lui permettant de l'exécuter. Le troisième type de la main d'œuvre hiérarchique est un cas spécifique du type hétérogène. Les employés sont regroupés par niveau ou *grade*. Un groupe d'employés d'un grade donné possède, en plus de ses propres qualifications, toutes les qualifications des grades qui lui sont inférieurs.

Nous verrons dans le Chapitre 3 que le processus de construction d'horaires ne requiert pas le traitement de tous ces niveaux décisionnels séparément. Étant donné que la décomposition décisionnelle résulte inévitablement en une dégradation de la qualité des solutions, la communauté scientifique s'oriente vers des approches intégrées. Étant donné la complexité accrue de telles approches, le défi est de proposer des méthodologies de résolution performantes.

Chapitre 2

Construction de quarts multi-jours dans un contexte multidépartement

Dans la section 2.1, nous présentons le problème de construction de quarts de travail personnalisés sur un horizon multi-jour et dans un contexte multidépartement. La terminologie et les notations requises pour la définition du problème y sont détaillées. Dans la section 2.2, nous décrivons la méthodologie de recherche permettant de résoudre le problème considéré et d'évaluer les résultats escomptés. Afin de procéder à la partie expérimentale, la génération des données est une étape importante permettant de proposer des scénarios qui s'ajustent bien à la réalité des environnements que nous visons. La section 2.3 est dédiée à la description du jeu de données qui sera utilisé dans l'intégralité de ce travail.

2.1 Problématique de la thèse

Le travail dans les grandes organisations requiert souvent une division en départements. Dans le jargon des entreprises, un *département* est un organe qui a pour mission l'accomplissement d'une fonction économique. Dans de tels environnements, la gestion de la main d'œuvre se fait en rattachant un ensemble d'employés à chacun des départements de l'entreprise. De plus, dans plusieurs organismes, les salariés, notamment ceux qui ont le plus de séniorité, possèdent suffisamment d'expérience pour travailler dans plusieurs départements. Selon la composition du travail, les goulots d'étranglements se déplacent d'un département à un autre. Un département est goulot lorsqu'il est hautement sollicité par les clients. Face à de telles situations, les dirigeants essaient d'équilibrer le fonctionnement de l'entreprise. Ceci se fait en transférant occasionnellement les employés ayant les qualifications requises pour travailler dans les départements goulots.

Cette thèse s'adresse à un problème générique de construction de quarts de travail personnalisés pour une main d'œuvre hétérogène, sur un horizon multi-jour et dans un contexte

multidépartement avec transfert d'employés. Chaque département de l'organisation est monoactivité. Ce problème peut être rencontré dans plusieurs applications réelles telles que les guichets bancaires, les hôpitaux publics, les centres d'appel, les grands magasins, etc.

2.1.1 Définitions, terminologie et notations

Soit une organisation constituée d'un ensemble de départements D et faisant appel à un ensemble d'employés E . Chaque employé e de l'organisation est rattaché à un département spécifique noté d_e , que l'on désigne par « *département d'origine* ». De plus, un employé e possède les qualifications nécessaires pour travailler dans d'autres départements appelés « *départements d'accueil* ». Cet ensemble de départements sera noté D_e . Lorsque l'employé travaille dans l'un de ces départements d'accueil d , on dit qu'il est transféré dans d . Ainsi, « *un transfert* » est défini comme étant une période travaillée par un employé donné en dehors de son département d'origine. Un « *bloc de transfert* » est défini comme étant un ensemble de périodes, à l'intérieur d'un quart, transférées successivement dans le même département d'accueil. Chaque département d utilise un ensemble d'« *employés internes* » noté E_d et un ensemble d'« *employés externes* » noté E_d^{ext} . L'ensemble des employés internes est constitué des employés rattachés au département en question et représente la capacité propre au département. L'ensemble des employés externes est constitué des employés qualifiés pour le département en question et qui proviennent des autres départements. Chaque employé travaille, de préférence, au sein de son département d'origine tant qu'on a besoin de lui pour accomplir sa fonction primaire. Il arrive qu'un employé soit redirigé occasionnellement vers l'un de ses départements d'accueil si on a besoin de lui pour accomplir l'une de ses fonctions secondaires. Cet employé constitue, dans ce cas, une ressource empruntée temporairement afin d'augmenter la capacité insuffisante du département goulot.

Soit I l'intervalle de temps durant lequel on se propose de construire les horaires multidépartements. Cet intervalle est appelé « *horizon de planification* ». Cet horizon est divisé en un ensemble de périodes de temps de durées égales à 15 minutes. Chaque période de l'horizon de planification est définie par son heure de début. Les périodes sont triées selon un ordre croissant de leurs heures de début. Une solution du problème sera donc constituée de blocs de travail débutant et finissant aux frontières de ces périodes. L'horizon de planification est de l'ordre de quelques jours. Soit J l'ensemble de ces jours. Chaque jour j est défini par l'ensemble des périodes le constituant. Cet ensemble est noté I_j . Il est important de souligner le fait que l'horizon de planification est continu. En d'autres termes, on ne peut pas diviser le problème de construction des quarts pour chaque jour séparément. En effet, dans les environnements de travail que l'on considère, la convention collective stipule qu'un quart peut commencer durant un jour donné et finir le jour suivant. Ceci est généralement rencontré dans les entreprises qui opèrent 24 heures par jour tels que les centres d'appel et les hôpitaux. En conséquence, le problème doit être traité d'une façon simultanée pour tous les jours

ensemble, ce qui augmente sa complexité. Dans ce qui suit, sans perte de généralité, l'horizon de planification est égal à une semaine et débute lundi à minuit. L'entreprise opère sept jours sur sept. On suppose que chaque employé possède une séquence de jours travaillés et de jours de congé assignée à priori. On note J_e l'ensemble des jours de travail pour un employé e .

Chaque département d possède une courbe de demande. Cette courbe fournit, pour chaque période i de l'horizon de planification, le nombre d'employés, noté $r_{d,i}$, requis pour maintenir un niveau de service visé. Dans notre cas, on suppose que la demande en main d'œuvre est *déterministe*. En d'autres termes, le nombre d'employés requis dans un département donné et durant une période de l'horizon est une constante connue avec certitude.

On suppose que l'entreprise dispose d'un ensemble de *types de quart*, noté T , qui sert à définir les règles de travail quant aux heures de début et aux durées admissibles des quarts de travail. Chaque type de quart t est défini par un ensemble d'heures de début S_t et un ensemble de durées admissibles L_t . L'ensemble L_t est composé de toutes les durées comprises entre une durée minimale l_t^{min} et maximale l_t^{max} . Étant donné que l'horizon de planification est mutli-jour, on notera $S_{j,t}$ l'ensemble des périodes de début admissibles le jour j obtenu en translatant les heures de début admissibles S_t à ce jour. On suppose que l'employé est payé au prorata du nombre d'heures travaillées. En conséquence, la restriction sur les durées des quarts permet, d'une part, d'assurer à chaque employé un nombre minimal d'heures travaillées et donc rémunérées et, d'autre part, d'interdire les quarts qui sont très longs et qui peuvent provoquer des problèmes physiques ou mentaux.

Le transfert des employés entre les départements est régi par un ensemble de règles fixées par la convention collective, appelées « *règles de transfert* ». Chaque employé doit passer au minimum une proportion T_{min} de sa charge horaire dans son département d'origine. De plus, un employé peut soit travailler son quart au complet dans son département d'origine ou dans l'un des départements d'accueil, soit commencer le quart dans son département d'origine et le terminer dans un département d'accueil, soit l'inverse. Donc, lorsque le quart est multidépartement, on suppose que le bloc de transfert peut soit commencer à n'importe quel moment du quart de travail et se terminer à la fin du quart, soit commencer au début du quart de travail et se terminer à n'importe quel moment du quart. Il est à noter qu'il existe une durée minimale de travail consécutif dans un département donné. Quoique les règles de transfert choisies soient restrictives, elles reflètent les pratiques réelles appliquées par les dirigeants. Il est vrai que le transfert permet d'éviter une perte de capacité du personnel, mais un transfert fréquent peut occasionner une perturbation du travail et une perte de productivité. D'un côté, les superviseurs préfèrent généralement diriger le même personnel à des fins d'évaluation et de formation. La plupart des employés préfèrent travailler avec la même équipe. Une baisse de productivité peut donc être observée lorsque l'employé reste en dehors de son département d'origine très longtemps ou lorsqu'il est transféré dans plu-

sieurs départements durant le même quart. D'un autre côté, le transfert peut avoir des effets positifs sur la motivation des employés souvent désireux de changer de temps en temps la nature du travail et de découvrir les autres départements de l'entreprise.

Pour chaque employé $e \in E$, pour chaque jour travaillé $j \in J_e$ de cet employé, on note $Q_{e,j,t}$ l'ensemble des quarts admissibles relativement à un type de quart $t \in T$. Dans ce qui suit, on se propose de définir formellement cet ensemble de quarts. Pour ce faire, on introduit la notion de « *bloc de travail* » qui désigne un ensemble de périodes travaillées consécutivement durant un quart dans un département donné. Chaque bloc est défini par son heure de début, sa durée, et le département au sein duquel le bloc de travail est exécuté. La durée d'un bloc doit être au minimum égale à B_{min} . En se rapportant aux règles de transfert, on peut distinguer deux cas selon que le quart soit formé d'un seul bloc de travail ou de deux blocs de travail.

- « *Quart monodépartement* » : il s'agit d'un quart q formé par un seul bloc de travail. L'employé le recevant reste dans le même département durant toute la durée du quart. Il est entièrement défini par son heure de début s_q , sa durée l_q et le département de sa réalisation d_q .
- « *Quart multidépartement* » : il s'agit d'un quart q formé par deux blocs de travail. L'employé le recevant se déplace entre son département d'origine et l'un de ses départements d'accueil. Dans ce cas, le quart q est entièrement défini par son heure de début s_q , la durée du bloc de début l_b , la durée du bloc de fin l_f , le département de la réalisation du bloc de début d_b et le département de la réalisation du bloc de fin d_f .

À des fins de modélisation, on définit le département fictif d_0 comme étant un département admissible à tous les employés de l'entreprise. Un quart monodépartement est un quart à deux blocs dont le bloc de début est une copie du quart et le bloc de fin est un bloc de durée nulle se déroulant dans le département fictif. Ainsi, tout quart sera désigné formellement par le quintuplet $(s_q, l_b, l_f, d_b, d_f)$. Un quart est dit personnalisé du moment qu'il est affecté à un employé individuellement. Un quart personnalisé de l'employé e le jour j est dit admissible relativement à un type de quart t si et seulement s'il commence à l'une des heures de début admissibles du type de quart t le jour j , si sa durée appartient à l'ensemble des durées admissibles du type de quart t , s'il est réalisé au complet dans un département admissible de l'employé e ou s'il est formé de deux blocs et réalisé dans deux départements admissibles dont l'un est le département d'origine d_e .

De plus, aucune période ne doit être travaillée durant un jour de repos d'un employé. En d'autres termes, si un quart reçu le jour j chevauche le jour suivant $j + 1$ et si c'est un jour de repos ($j + 1 \notin J_e$) alors ce quart n'est pas admissible et il ne faut pas le considérer. Il est à noter que les quarts qui chevauchent le jour suivant le dernier jour de l'horizon de

planification sont considérés comme étant admissibles. On note s_j la période de début du jour j .

En adoptant ce formalisme et en exprimant les données temporelles en fonction de la période de discrétisation de 15 minutes, l'ensemble des quarts personnalisés de l'employé e le jour j admissibles relativement au type de quart t est donné par l'expression suivante : $Q_{e,j,t} = \{(s_q, l_b, l_f, d_b, d_f) | s_q \in S_{j,t}; l_q = l_b + l_f \in L_t; l_b \geq B_{min}; l_f \in [B_{min}, +\infty[\cup \{0\}; s_q + l_q - 1 < s_{j+1}$ si $j + 1 \in J \setminus J_e; (d_b, d_f) \in (D_e \times \{d_e\}) \cup (\{d_e\} \times D_e)$ si $l_f > 0$ et $(d_b, d_f) \in (\{d_e\} \cup D_e) \times \{d_0\}$ si $l_f = 0\}$.

L'ensemble de tous les quarts admissibles Q peut être partitionné, selon la nature des quarts, en deux catégories : *internes* et *externes*.

- « Quarts internes » (Q^{int}) : ce sont des quarts à un seul bloc qui sont travaillés au complet dans le département d'origine.
- « Quarts externes » (Q^{ext}) : ce sont des quarts à un seul ou à deux blocs qui peuvent être travaillés au complet dans un département d'accueil (s'il s'agit d'un quart à un seul bloc), ou que l'un des deux blocs soit travaillé dans un département d'accueil et l'autre soit travaillé dans le département d'origine (s'il s'agit d'un quart à deux blocs).

Ainsi, les règles de la convention collective relatives à un quart en particulier sont celles liées à la définition de son heure de début, de sa durée et des transferts effectués à l'intérieur de ce quart s'il y a lieu. Étant donné que le problème traité s'adresse à la construction de quarts de travail sur un horizon multi-jour, outre les règles de définition des quarts personnalisés admissibles, il faut préciser les règles qui régissent l'admissibilité de la séquence de quarts assignés à un employé durant tout l'horizon. Le choix de ces restrictions se fait en essayant de garder le problème générique autant que possible. Ainsi, on distingue la règle de transfert, énoncée précédemment, qui impose une restriction sur le pourcentage maximal de périodes travaillées en dehors du département d'origine par rapport à la charge totale travaillée. De plus, on retient deux règles typiques régissant l'horaire au complet, dont le choix est justifié par le fait qu'elles soient rencontrées dans la majorité des environnements de travail. La première assure que chaque employé reçoit un bloc de repos de durée minimale S_{min} entre deux quarts travaillés consécutivement. La deuxième règle limite la charge totale travaillée pour chaque employé à une borne maximale notée L_{max} .

On suppose que chaque quart induit deux types de coûts différents : « le coût de travail » et « le coût de transfert ». Le coût de travail est calculé proportionnellement à la durée du quart. Soit c_e^l le coût en unités monétaires de la rémunération d'un employé donné e par période travaillée. Il est à noter que la définition du coût unitaire de travail peut varier d'un employé à un autre pour tenir compte de l'hétérogénéité de la main d'œuvre. En effet, ce coût est étroitement lié à des spécifications de l'employé à savoir sa séniorité, son expérience, son grade, sa

productivité, etc. Le coût de transfert, quant à lui, est calculé proportionnellement à la durée de l'intervalle de temps transféré. Soit $c_{e,d}^t$ le coût unitaire par période transférée de l'employé e dans son département d'accueil d . En effet, comme l'affirme Bechtold (1988), l'emploi de la main d'œuvre multidépartementale est généralement indésirable pour plusieurs raisons dont font partie la variation de la productivité des employés selon les départements et la diminution de la motivation des employés pour le travail suite à un changement de superviseur ou de membres d'équipes. Conséquemment, le coût de transfert est perçu dans le présent travail comme une pénalité qui mesure l'insatisfaction d'un employé, sa baisse de productivité, etc. Ainsi, une période transférée implique le même coût d'une période travaillée (coût de travail), en plus d'un coût additionnel de transfert. Le choix de définir une pénalité par employé par département d'accueil confère à la structure de coût que l'on considère une grande flexibilité pour s'ajuster aux objectifs des dirigeants. Par exemple, les coûts $(c_{e,d}^t)_{d \in D_e}$ permettent au planificateur d'attribuer une pénalité qui est d'autant plus grande que l'employé e baisse de productivité dans un département d'accueil d . Dans d'autres cas, chaque employé peut être amené à ordonner la liste de ses départements d'accueil selon un ordre décroissant de ses préférences d'y travailler. En conséquence, des pénalités $(c_{e,d}^t)_{d \in D_e}$ sont attribuées en tenant compte de cet ordre préférentiel (Al-Yakoob et Sherali, 2007). Dans cette thèse, de telles considérations sont envisagées lors de la génération des scénarios mais ne sont pas étudiées de plus près même si le problème, tel qu'il est défini, permet d'en tenir compte.

Une « sous-couverture » est obtenue dans un département donné d et durant une période i de l'horizon lorsque le besoin d'un employé pour cette période n'est pas satisfait. Pour chaque département d , on définit un coût unitaire d'une période de sous-couverture c_d^- . De par sa définition, ce coût dépend uniquement du département au sein duquel on fait face à la sous-couverture. Souvent, la sous-couverture dans l'un des départements est plus lourde à supporter pour la compagnie que dans d'autres. Par exemple, dans un hôpital, la sous-couverture dans le département d'urgence n'aura pas les mêmes conséquences que celles dans le département de kinésithérapie. De plus, le coût de sous-couverture ne dépend pas de la période de son apparition. Autrement dit, on suppose que la sous-couverture a le même impact peu importe le moment durant lequel on a un déficit en employés. Ceci dit, le problème tel qu'il sera modélisé dans la suite permet de considérer un coût de sous-couverture différent par période. De même, une « surcouverture » est obtenue dans un département donné d et durant une période de l'horizon i lorsque le nombre d'employés travaillant cette période dépasse la demande d'un employé. Cette surcouverture signifie qu'il existe un employé en service dans le département d qui est improductif durant la période i . Pour chaque département d , on définit un coût unitaire d'une période de surcouverture c_d^+ . La sous-couverture est généralement plus coûteuse que la surcouverture étant donné que le déficit en ressources occasionne des ventes perdues, une prestation de service de moindre qualité et donc une insatisfaction des clients.

2.1.2 Énoncé du problème

Dans le reste du document, on s'intéresse au problème de construction de quarts personnalisés sur un horizon multi-jour et dans un contexte multidépartement défini plus haut. Le cadre conceptuel est illustré par la Figure 2.1. Ce problème peut être énoncé comme suit : « Soit une compagnie utilisant des types de quart T , formée par un ensemble de départements D et employant une main d'œuvre E . Chaque département de la compagnie dispose d'un ensemble de ressources en personnel et peut emprunter occasionnellement des ressources provenant des autres départements, si celles-ci possèdent le savoir-faire requis pour y travailler. Étant donné que chaque ressource (employé) e reçoit au préalable un cycle de travail J_e , l'objectif du problème est d'assigner à chaque employé e de la compagnie, durant chaque jour travaillé de son cycle j , un quart, choisi parmi l'ensemble des quarts personnalisés admissibles $Q_{e,j} = \cup_{t \in T} Q_{e,j,t}$. Le choix des quarts pour tous les employés durant l'horizon de planification se base sur la minimisation d'une fonction de coût. Cette fonction de coût est une somme pondérée du coût total des périodes travaillées, du coût total des périodes transférées et du coût total des sous-couvertures et des surcouvertures. L'ensemble des quarts sélectionnés doit respecter les règles de la convention collective listées dans la section 2.1. »

La flexibilité porte sur la définition des heures de début et des durées des quarts et sur le placement des transferts à l'intérieur des quarts. Étant donné le haut degré de flexibilité considéré, l'énumération de tous les quarts explicites admissibles résulte en un nombre très grand de possibilités. La difficulté du problème est inhérente à ce nombre qui croît exponentiellement lorsque la taille des instances augmente. Ceci constitue la difficulté majeure lors de la résolution du problème. La problématique de la thèse consiste donc à concevoir des approches de résolution performantes.

2.2 Méthodologie et contributions

L'automatisation de la construction des horaires de travail devient une préoccupation dans plusieurs organisations. Si la main d'œuvre multidépartementale permet d'offrir aux décideurs des calendriers de travail qui répondent mieux à leurs objectifs, le partage des ressources en effectifs complique la gestion des temps notamment la construction d'horaires. Dans ce cadre, cette thèse vise essentiellement à :

- proposer et comparer plusieurs approches de modélisation implicites et explicites pour le problème étudié,
- et concevoir et comparer plusieurs approches de résolution heuristiques pour le problème étudié. Nous proposons deux approches de résolution : une approche dite de « *prétransfert* » et une approche dite de « *post-transfert* ». Une étude comparative entre

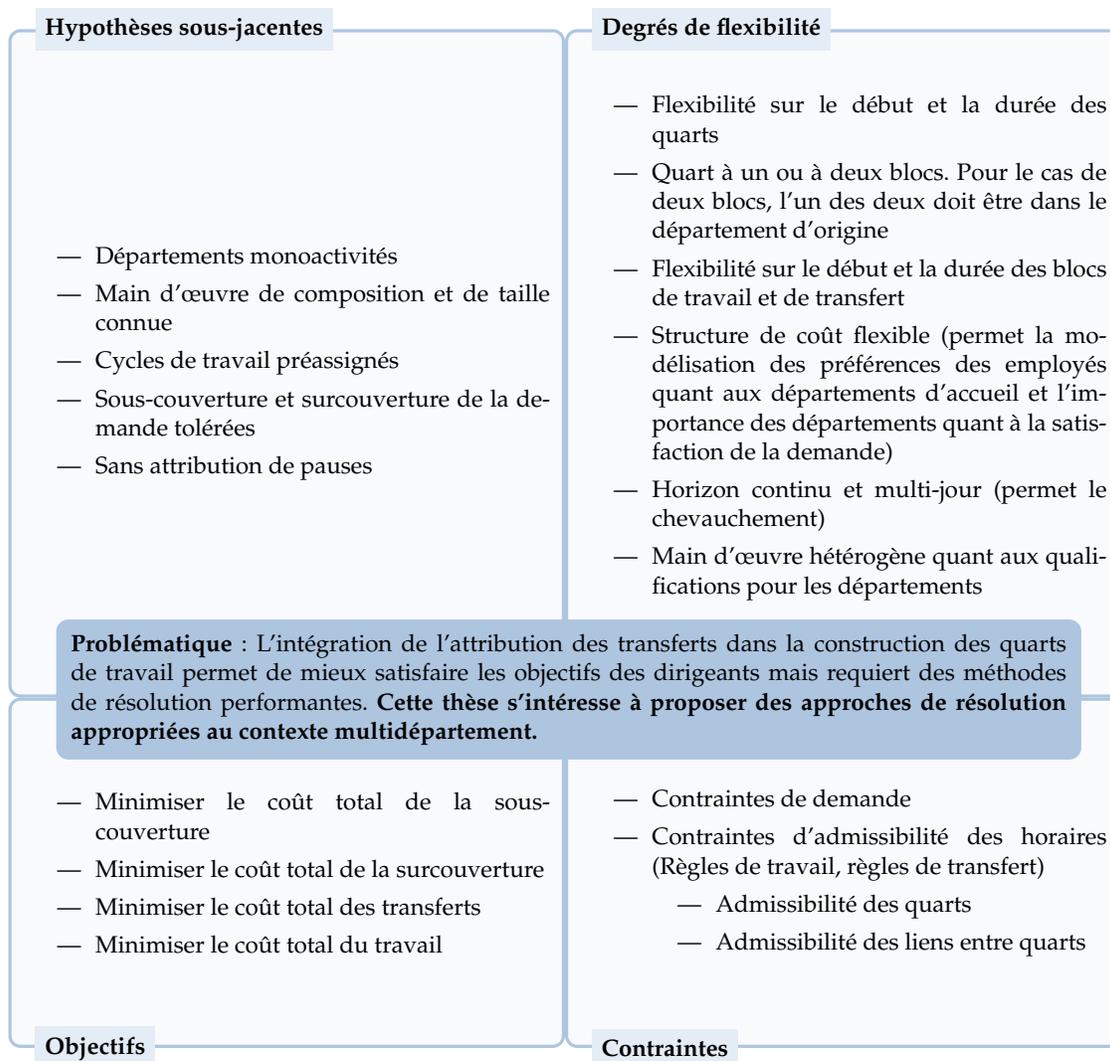


Figure 2.1 – Cadre conceptuel du problème traité

la qualité des calendriers obtenus est proposée afin d'orienter le choix des concepteurs des logiciels d'optimisation.

2.2.1 Approches de modélisation

Le premier objectif de recherche consiste à proposer différentes formulations du problème et à comparer la performance de ces formulations. Dans la littérature, il existe deux approches distinctes de modélisation : *explicite* et *implicite*. L'approche explicite consiste à énumérer l'ensemble des horaires de travail admissibles qui respectent les réglementations de travail. Une variable de décision est associée à chaque horaire. Un modèle de recouvrement généralisé initialement introduit par Dantzig (1954) est utilisé pour choisir le meilleur ensemble d'horaires. L'approche implicite permet de réduire le nombre de variables de décision en modélisant implicitement certaines formes de flexibilité. Plusieurs techniques sont utilisées

à savoir le recours à des problèmes de transport ayant une structure particulière (Bechtold et Jacobs, 1990; Rekik et collab., 2004, 2010) et le recours à la programmation par contraintes (Côté et collab., 2011b).

Dans le cadre de cette thèse, nous proposons différentes formulations implicites et explicites pour modéliser le problème de construction de quarts personnalisés sur un horizon multi-jour et dans un contexte multidépartement. Quoique le problème ne traite pas le niveau décisionnel de placement des jours de travail et de repos formant l'horaire, il permet de tenir compte de plusieurs réglementations considérées par un problème de construction simultané de cycles et de quarts. En effet, étant donné que le problème soit défini sur un horizon de plusieurs jours, il fait intervenir plusieurs règles de liaison entre les quarts formant l'horaire. Nous citons par exemple la charge totale de travail et la proportion minimale de travail sans transfert. Plusieurs autres aspects réalistes du problème sont considérés par nos formulations. La flexibilité porte sur les heures de début des quarts, les durées de travail, le placement des transferts et l'hétérogénéité de la main d'œuvre ayant des qualifications multiples et différentes. Une autre vocation de nos modèles est la construction d'horaires personnalisés. Ils permettent donc de définir des coûts de travail selon la séniorité des employés et des coûts de transfert selon les préférences des employés. De plus, ils permettent de tenir compte de la disponibilité de chaque employé individuellement sachant ses jours de repos. Plusieurs nouvelles formulations implicites sont proposées. Elles se basent sur le concept des contraintes forward et backward introduites par Bechtold et Jacobs (1990). Nous généralisons ce concept au contexte multidépartement. En se basant sur un ensemble de tests, une analyse comparative des performances permettra d'identifier les formulations les plus faciles à résoudre. Nos tests montrent que les formulations sont performantes pour la résolution directe du problème pour des instances de petite taille. Pour les instances de grande taille, nous proposons des approches de résolution heuristiques.

2.2.2 Approche de résolution prétransfert

Pour résoudre le problème global, il faut spécifier pour chaque employé et durant chaque période de l'horizon si 1) l'employé est en repos, 2) l'employé travaille dans son département d'origine ou 3) l'employé travaille dans un département d'accueil spécifique. Le problème global se ramène donc à résoudre deux niveaux décisionnels : 1) la spécification des périodes travaillées et 2) l'affectation des départements à ces périodes.

L'approche de résolution prétransfert est une heuristique à deux niveaux : le premier niveau est une étape d'« agrégation » et le deuxième niveau est une étape de « désagrégation ». Le premier niveau résout un problème approximatif que l'on désigne par le problème « agrégé ». Ce problème a la vocation d'intégrer les deux niveaux décisionnels ensemble évitant ainsi la perte de qualité provoquée par la décomposition décisionnelle. Cependant, cette intégration aboutit à un problème impossible à résoudre directement. Pour diminuer sa complexité,

le problème approximatif est défini en se basant sur des données temporelles de précision moindre. La période de discrétisation lors de l'étape d'agrégation est alors plus grossière que celle du problème initial. Nous proposons une heuristique basée sur la génération de colonnes pour résoudre ce problème obtenant ainsi un ensemble de quarts personnalisés agrégés.

Le deuxième niveau permet d'affiner la solution agrégée en l'ajustant au problème original. Dans cette étape, nous définissons un problème « désagrégé ». Ce problème a la vocation d'utiliser les données temporelles initiales du problème global. Par contre, il examine un ensemble restreint de quarts admissibles. La détermination de cet ensemble restreint se fait au biais d'une remise en question partielle des décisions entreprises au niveau agrégé. Le défi consiste donc à choisir l'information à retenir à partir de la solution agrégée afin de délimiter l'ensemble restreint des quarts désagrégés de la meilleure façon possible. Toutefois, le problème désagrégé est impossible à résoudre directement compte tenu de la taille des organisations traitées. Nous définissons ainsi une heuristique de décomposition espace-temps pour résoudre ce problème. Une heuristique d'horizon roulant est proposée combinée avec une séparation selon la composante départementale. Pour chaque fenêtre de temps et chaque département, un sous-problème est défini. Il permet de sélectionner, parmi l'ensemble restreint de quarts désagrégés débutant à l'intérieur de la fenêtre, un ensemble de quarts permettant de combler les besoins du département considéré en utilisant toute la main d'œuvre interne disponible et en réservant certains employés externes. Les sous-problèmes restreints à une fenêtre donnée peuvent être résolus selon des règles de précedence précises. La résolution se fait séquentiellement d'une fenêtre de temps à l'autre. Différentes configurations sont conçues pour 1) l'information agrégée retenue et 2) la méthode de résolution d'un sous-problème donné. Des tests expérimentaux permettent d'identifier la meilleure configuration selon la taille des organisations.

Cette nouvelle approche ouvre la perspective de la construction des horaires dans des organisations de grande taille, et ceci en intégrant plusieurs niveaux décisionnels du processus de construction d'horaires : la construction des quarts, la liaison entre les quarts, l'affectation des quarts multi-jours aux employés et l'affectation des départements aux horaires personnalisés. Tous ces niveaux intégrés ensemble aboutissent à des problèmes très difficiles à résoudre même dans des organisations de petite taille. Notre heuristique constitue donc un outil puissant, non seulement pour l'intégration décisionnelle du processus de construction, mais aussi pour la gestion centralisée des horaires dans les grandes organisations. Les tests expérimentaux montrent que la construction simultanée des quarts mutli-jours personnalisés et l'affectation des transferts permet une amélioration nette des performances si l'approche de résolution heuristique permet de bien gérer la complexité du problème.

2.2.3 Approche de résolution post-transfert

L'approche de résolution post-transfert est plus intuitive. Comme l'approche de résolution prétransfert, cette approche est une heuristique à deux étapes : agrégée et désagrégée. Elle diffère principalement de l'approche de résolution prétransfert par la décomposition des deux niveaux décisionnels : la spécification des périodes travaillées et l'affectation des départements à ces périodes.

Dans l'étape d'agrégation, les transferts des employés entre les départements sont interdits. C'est alors le premier niveau décisionnel de spécification des périodes travaillées qui est traité à cette étape. Un sous-problème associé à chaque département est résolu séparément. Cette étape sert ainsi à prendre les décisions relatives aux heures de début et aux durées de travail tout en considérant exclusivement les quarts internes agrégés. Ce choix est justifié par le fait que l'horaire de l'employé sera constitué principalement par des blocs de travail internes qui doivent représenter au minimum un pourcentage T_{min} de son horaire total.

L'étape de désagrégation concerne le deuxième niveau décisionnel d'affectation des départements à l'intérieur des quarts formant la solution agrégée. Notons que la décomposition décisionnelle entre la spécification des périodes travaillées et le placement des départements n'est pas absolue du moment que l'on révisera partiellement les décisions sur les périodes travaillées, en plus du placement à postériori des blocs de transfert. Nous définissons un ensemble restreint de quarts construits par une perturbation des heures de début et des durées agrégées et un placement à postériori des départements à l'intérieur de ces plages horaires. Nous formulons le problème de deux manières différentes : la première manière utilise une formulation explicite de quarts et la deuxième manière utilise cette formulation suivie par une autre formulation implicite. Nous résolvons le problème de deux manières différentes : une procédure de séparation et d'évaluation progressive et notre procédure basée sur la génération de colonnes. Nous effectuons une analyse comparative des performances afin de retenir la méthode la plus performante pour configurer l'approche de résolution post-transfert. Finalement, nous comparons entre les deux approches de résolution prétransfert et post-transfert.

Notre approche de résolution post-transfert offre un outil très pratique pour l'ajustement dynamique des horaires. Supposons que le directeur d'une organisation procède à la construction des horaires de ses employés quelques semaines en avance pour leur permettre de planifier leur vie privée. Ces horaires sont construits en se basant sur des prévisions de demande. Quelques jours avant, la demande est plus précise (réception de commandes fermes, réservations des clients, arrivée de nouveaux patients hospitalisés, etc). Notre approche permet donc de réviser les horaires déjà construits en perturbant légèrement les heures de début et les durées, et en permettant de remettre en question toutes les décisions quant à la composition de l'horaire en termes de départements. On peut donc demander à une infirmière de

commencer une heure plus tôt le travail et de la transférer dans un département différent de son département d'origine. Une autre situation peut se présenter lorsqu'un employé est obligé de s'absenter pour des raisons de santé. L'approche permet de transférer un employé inactif dans son département d'origine pour remplacer l'employé absent. Notre heuristique de résolution post-transfert montre que, dans le contexte multidépartement, le transfert a posteriori est plus adéquat pour approcher un problème de « *rescheduling* » que pour approcher un problème de construction d'horaires.

2.3 Générateur de données

Dans cette section, nous passons à décrire les instances utilisées dans la partie expérimentale de la thèse. Nous avons construit un générateur de données capable de simuler les problèmes rencontrés par les gestionnaires d'entreprises. Nous considérons un jeu de données formé par 48 instances qui diffèrent par leur degré de complexité. L'objectif est de questionner les performances et les limitations de nos approches de résolution.

2.3.1 Description des instances

Dans ce qui suit, nous décrivons la structure de données formant une instance. Chaque instance est constituée d'un ensemble de départements D , un ensemble de cycles S , un ensemble d'employés E , et un ensemble de types de quart T .

Chaque département $d \in D$ est caractérisé par :

- un ensemble d'employés internes (E_d),
- un coût d'une heure de sous-couverture ($4 c_d^-$) (rappelons que le coût c_d^- est défini par période de 15 minutes, il en est de même pour le reste des coûts),
- un coût d'une heure de surcouverture ($4 c_d^+$),
- et une courbe de demande qui spécifie, pour chaque période de l'horizon de planification $i \in I$, le nombre d'employés requis ($r_{d,i}$).

Chaque cycle (ou encore séquence de travail) $c \in C$ est caractérisé par un ensemble de jours de travail et de jours de repos durant l'horizon de planification.

Chaque employé $e \in E$ est caractérisé par :

- un cycle qui lui est assigné et qui détermine ses jours de travail (J_e),
- son département d'origine (d_e),
- un ensemble de départements d'accueil (D_e),
- un coût de rémunération horaire ($4 c_e^l$),
- et un coût qui mesure la pénalité d'une heure transférée vers chaque département d'accueil $d \in D_e$ ($4 c_{e,d}^t$).

Chaque type de quart $t \in T$ est caractérisé par :

- un ensemble d’heures de début possibles (S_t)
- et un ensemble de durées possibles (L_t).

2.3.2 Procédure de génération des instances

Pour toutes les instances générées, nous avons considéré un horizon de planification d’une semaine. Il est discrétisé en 672 périodes de durées égales à 15 minutes. La durée de repos minimale séparant deux quarts travaillés durant deux jours consécutifs est égale à $S_{min} = 14h$. Les cycles déjà assignés aux employés sont obtenus en attribuant deux jours de repos consécutifs durant la semaine. Nous avons fixé la charge horaire hebdomadaire maximale à $L_{max} = 40h$ pour chaque employé (en moyenne huit heures par jour). Chaque employé doit passer au minimum $T_{min} = 70\%$ de son horaire dans son département d’origine. Un bloc de travail (interne ou externe) ne peut pas durer moins qu’une heure ($B_{min} = 1h$).

Pour tous les départements de l’organisation, le coût unitaire par heure de sous-couverture est égal à 20\$ et celui de surcouverture est égal à 10\$. Le taux de rémunération horaire est égal à 15\$ pour tous les employés. Les coûts de transfert sont générés comme suit. Nous supposons que chaque employé possédant n départements d’accueil est amené à ordonner l’ensemble de ses départements selon un ordre croissant de ses préférences. Le département d’accueil qui est le plus préférable pour l’employé coûte 1\$, le 2^{ème} coûte 2\$ et ainsi de suite jusqu’au $n^{\text{ème}}$ qui coûte n \$ par heure de transfert.

Les instances générées considèrent un seul type de quart dont les durées admissibles sont entre 7h et 9h inclusivement. L’ensemble des débuts admissibles permet au planificateur de construire des quarts de jour, de soir et de nuit comme suit :

- les quarts de jour débutent aux 15 minutes entre 8h00 et 12h00,
- les quarts de soir débutent aux 15 minutes entre 16h00 et 20h00,
- et les quarts de nuit débutent aux 15 minutes entre 0h00 et 4h00.

Nous avons généré 12 classes d’instances. Chaque classe correspond à une organisation qui est caractérisée par un ensemble de départements et un ensemble d’employés. Ce choix de classification est expliqué par le fait que la taille de l’organisation est déterminante de la complexité du problème. Chaque classe sera notée selon le nombre de départements et le nombre d’employés. Par exemple, la classe D20E600 contient 20 départements et 600 employés. Le Tableau 2.1 donne le nombre total de quarts internes $|Q^{int}|$ et externes $|Q^{ext}|$ pour chaque classe d’instances.

Les qualifications des employés pour les départements sont générées aléatoirement selon une distribution normale dont la moyenne et l’écart type sont indiqués dans le Tableau 2.1.

Donc, si n_{Qua}^e représente le nombre de qualifications de l'employé e , on note :

$$\mu_{Qua} = \frac{\sum_{e \in E} n_{qua}^e}{|D||E|}$$

$$\sigma_{Qua} = \frac{1}{|D|} \sqrt{\frac{\sum_{e \in E} (n_{qua}^e - \mu_{qua})^2}{|E|}}$$

Prenons l'exemple de l'organisation D20E800. Chaque employé est qualifié en moyenne pour travailler dans 6.76 départements (soit 34%). Ce nombre varie entre 1 (aucun département d'accueil) et 19 (18 départements d'accueil). L'écart type de ce nombre de qualifications est égal à 3.65 départements (18 %). Nous avons choisi des environnements de travail pour lesquels chaque employé est qualifié en moyenne pour 38% des départements avec un écart type moyen de 18%. Ces paramètres peuvent être ajustés dans le générateur permettant ainsi une analyse de sensibilité si le décideur souhaite estimer le gain perçu en augmentant le nombre de qualifications de ses employés.

Classe	# Qualifications		# Quarts	
	μ_{Qua} (%)	σ_{Qua} (%)	$ Q^{int} $	$ Q^{ext} $
D2E20	62.50	21.65	43,692	549,635
D3E20	45.00	15.90	43,692	776,607
D5E20	35.00	15.33	43,692	1,663,141
D5E50	35.60	20.12	109,230	4,315,625
D5E70	34.86	15.38	153,060	5,815,856
D5E200	34.60	15.19	437,196	16,255,764
D10E200	35.25	17.03	437,196	56,296,115
D10E300	33.23	17.74	655,794	77,587,167
D10E400	35.23	19.34	874,392	112,517,893
D20E600	34.61	19.04	1,311,588	395,667,119
D20E800	33.82	18.25	1,748,784	513,656,851
D20E1000	34.13	18.94	2,185,980	648,826,824

Tableau 2.1 – Complexité du jeu de données

Pour toutes les organisations, la répartition des cycles sur les employés favorise l'utilisation des séquences ayant des samedis et des dimanches travaillés. Prenons l'exemple des séquences s_1 et s_6 dont les deux jours consécutifs de repos sont lundi et mardi pour s_1 et samedi et dimanche pour s_6 . Ces séquences correspondent respectivement aux proportions moyennes de 21% et 5% du total des cycles assignés aux employés. La moyenne est calculée sur l'ensemble de toutes les organisations (avec un écart type proche de 0% dans les deux cas). Ceci est pour générer des courbes de demande simulant des environnements pour lesquels la demande est importante à la fin de la semaine (magasins de détail, centres d'appel, restaurants, etc).

Pour chaque organisation, nous avons généré quatre profils de demande. Chaque profil est caractérisé par un degré de variabilité de la demande. Les profils sont triés de 1 à 4 selon un ordre croissant de cette variabilité. La demande associée au $i + 1^{\text{ème}}$ profil présente plus de variabilité que la demande associée au $i^{\text{ème}}$ profil. Plus précisément, nous proposons une procédure qui permet de générer des courbes de demande respectant les propriétés suivantes : le 1^{er} profil est caractérisé par une variabilité moyenne à toutes les 8 heures, le 2^{ème} à toutes les 4 heures, le 3^{ème} à toutes les 2 heures et le 4^{ème} à toutes les heures. Dans ce qui suit, nous décrivons la procédure d'obtention de ces profils de demande.

Nous commençons par diviser chaque jour de l'horizon en un nombre de parties de durées égales. Nous divisons la journée en 3, 6, 12 et 24 parties pour les profils 1, 2, 3 et 4 respectivement. Pour chaque partie de la journée p , pour chaque département d , nous générons un pourcentage moyen $\mu_{d,p}$ et un écart type $\sigma_{d,p}$ pour répartir les employés internes sur les parties de la journée dans leur département d'origine. Nous avons adopté des courbes de demande qui suivent une saisonnalité horaire durant les différents jours de l'horizon. En d'autres termes, pour un département donné, les hausses de demande se répètent plus ou moins durant les mêmes périodes de la journée. Par exemple dans la restauration, si le département « réception de la marchandise » est sollicité plus en début de journée, le département « nettoyage » est sollicité plus en fin de journée. Pour chaque jour de l'horizon j , pour chaque partie de la journée p , pour chaque département d , nous générons un nombre aléatoire $r_{d,p,j}$ entre 0 et 1 et nous utilisons la loi normale inverse de moyenne $\mu_{d,p}$ et d'écart type $\sigma_{d,p}$ pour déduire le pourcentage d'employés travaillant durant cette partie p du jour j dans leur département d'origine d .

Pour chaque profil, nous considérons des quarts qui commencent à tous les débuts des parties de journée perturbés de plus ou moins 15 minutes et qui durent exactement 8 heures. Nous résolvons un modèle d'affectation qui associe à chaque employé un et un seul quart sous les contraintes d'admissibilité des horaires et les contraintes des pourcentages générés. Nous relaxons les contraintes de pourcentage en admettant certains écarts et en pénalisant ces écarts dans la fonction objectif. Les coûts des quarts sont générés aléatoirement. Nous déduisons les courbes de demande correspondantes. Finalement, nous avons créé des perturbations aléatoires pour former artificiellement des hausses dans certains départements et des baisses dans d'autres. Il est à noter que pour éviter de biaiser les expérimentations, nous avons perturbé légèrement les courbes calculées permettant ainsi de rendre les instances difficiles à résoudre.

La Figure 2.2 illustre les courbes de demande de deux départements d'une instance donnée. Nous remarquons que les baisses de demande dans l'un des départements sont parfois accompagnées par des hausses de demande dans l'autre, tel est le cas de la fin du dimanche (dernier jour de l'horizon).

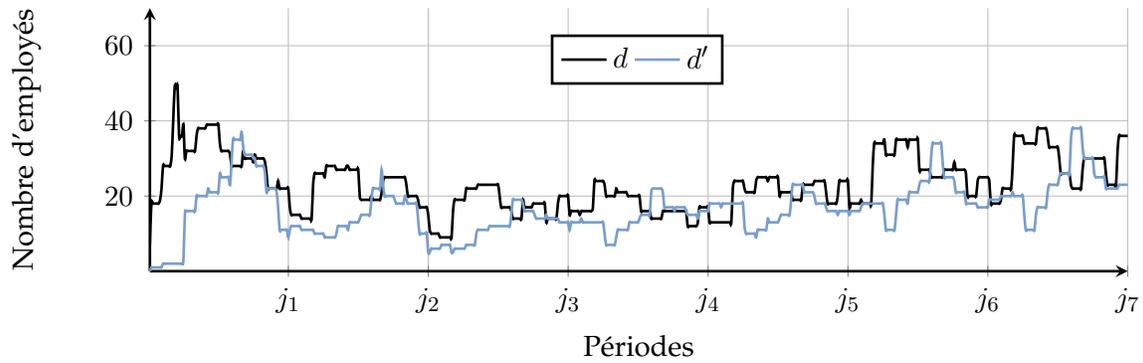


Figure 2.2 – Courbes de demande de deux départements de l'instance D20E1000P3

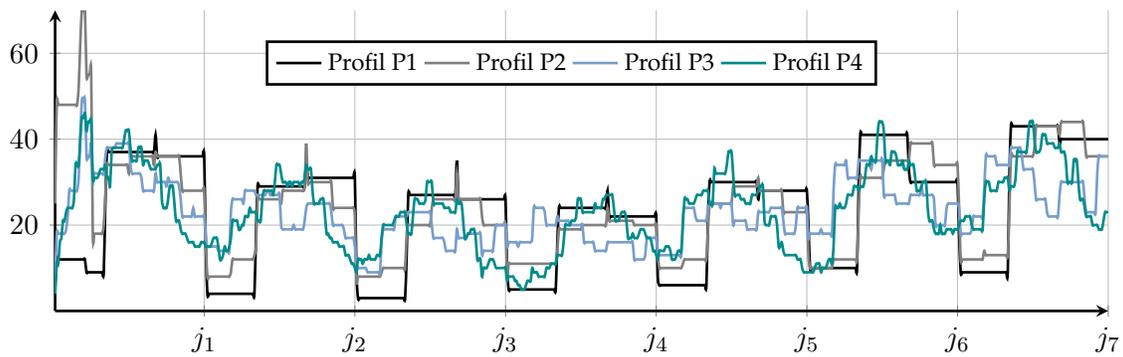


Figure 2.3 – Profils de demande d'un département

La Figure 2.3 illustre les courbes de demande d'un département selon les 4 profils. Nous remarquons que la variabilité de la demande augmente avec le profil.

Nous obtenons ainsi 48 instances distinctes tel qu'illustré par la Figure 2.4. L'instance D20E1000P3 correspond par exemple à l'organisation de 20 départements et 1000 employés et ayant une demande qui suit le profil P3.

Dans toute la suite, nous nous basons sur le jeu de données, ainsi généré, dans notre analyse expérimentale des approches de résolution qui seront proposées.

Jeu de données : 12 organisations et 4 profils de demande par organisation

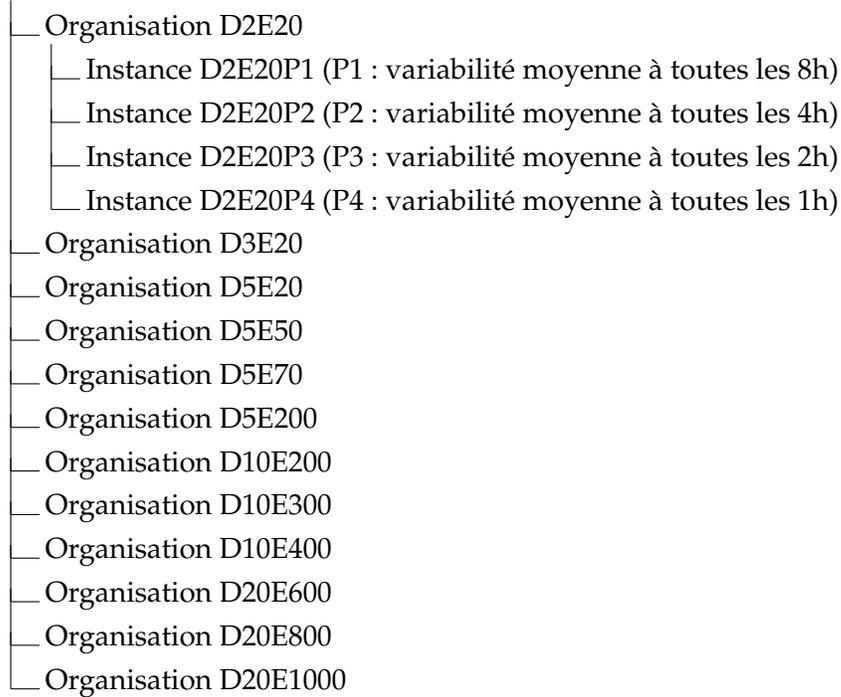


Figure 2.4 – Arborescence des instances

Chapitre 3

Revue de la littérature

Le domaine de la construction d'horaires a suscité l'intérêt de la communauté scientifique. Plusieurs travaux ont été élaborés et ont couvert de multiples applications réelles. La revue de la littérature de [Van den Bergh et collab. \(2013\)](#) est à notre connaissance la plus récente dans le domaine. Les auteurs classent les problèmes de construction d'horaires selon les quatre axes suivants : (1) la nature du problème, (2) les contraintes et les mesures de performance, (3) les méthodes de résolution et l'incorporation de l'incertitude et (4) le domaine d'application et la portée pratique des travaux.

La nature du problème inclut trois critères de classification : les caractéristiques du personnel, le type de décision traité et la flexibilité des quarts. Les auteurs soulignent que les chercheurs considèrent une main d'œuvre souvent à temps complet, parfois à temps partiel et rarement par intérim. Les auteurs identifient aussi les travaux considérant les qualifications des employés. De plus, la demande en effectifs peut être exprimée en nombre d'employés individuels ou par équipe selon le secteur. La demande est spécifiée par équipe notamment dans le domaine du transport. La composition de l'équipage d'un avion par exemple doit être conforme à des spécifications réglementées par le secteur. Les auteurs classent les travaux selon le type de décision considéré comme la construction des cycles de travail, l'affectation des activités et l'affectation de certaines ressources matérielles. La flexibilité portant sur la définition des quarts de travail tient compte du chevauchement entre jours et des départs et durées de travail.

Les contraintes et les mesures de performance concernent la partie conceptuelle du problème. Certains travaux considèrent des contraintes de demande dures alors que d'autres relaxent ces contraintes en pénalisant les variables d'écart correspondantes dans la fonction objectif. Les auteurs proposent une classification précise en divisant les travaux en quatre catégories selon que la sous-couverture soit permise ou non et que la surcouverture soit permise ou non. Un autre critère concerne les qualifications des employés. Alors que la majorité des travaux considèrent les qualifications des employés comme étant un intrant fixe du pro-

blème, certains remettent en question ces qualifications moyennant des coûts relatifs à la formation du personnel ou à la baisse de productivité. Plus encore, certains travaux considèrent les qualifications comme une décision prise par le modèle (un extrait du problème). Les auteurs identifient également les travaux qui considèrent le placement des pauses et soulignent le fait que la littérature récente néglige de plus en plus cet aspect. Un autre critère très important pour les décideurs est la structure de la fonction économique. Outre la masse salariale, d'autres coûts peuvent être considérés comme les indemnités des jours de fin de semaine travaillés, les bonus pour les employés les plus qualifiés, les pénalités des temps improductifs, etc. Les auteurs définissent aussi les contraintes d'équilibrage (*balancing constraints*) comme étant l'ensemble des contraintes servant à assurer l'équité entre les employés. On peut citer par exemple la répartition de la charge horaire travaillée et la répartition des fins de semaine travaillées sur l'ensemble des employés de l'organisation. Il existe des travaux qui introduisent des contraintes pour satisfaire équitablement les préférences des employés. Le dernier critère concerne les règles de la convention collective et la disponibilité des ressources matérielles. Les articles sont classés selon les spécificités relatives aux contraintes d'admissibilité des horaires tel que le nombre minimal de jours de repos par semaine, le nombre maximal d'heures travaillées par semaine, le nombre de machines disponibles, etc.

Pour les méthodes de résolution les plus populaires, on trouve la décomposition de Benders et la génération de colonnes lorsque le problème est formulé en un programme en nombres entiers, la programmation par contraintes, la simulation, les files d'attente, etc. L'incorporation de l'incertitude implique trois types : l'incertitude de la demande, de l'occurrence de la demande et de la capacité. Prenons comme exemple un centre d'appel. La variation aléatoire du temps requis pour servir un client constitue l'incertitude de la demande. La variation de l'instant de réception des appels constitue l'incertitude de l'occurrence de la demande. Finalement, la disponibilité des employés n'est pas garantie à cause des absentéismes et des retards. Dans ce cas, on parle d'incertitude de capacité.

Les auteurs divisent les travaux selon le domaine d'application en 6 classes : le secteur des services (incluant la santé, les centres d'appel, etc), le secteur du transport (aérien, ferroviaire, routier, etc), général (restaurants, supermarchés, festivals, parkings, etc), le secteur industriel, le secteur de la distribution (magasins de détail) et le secteur militaire. La portée pratique des travaux examine l'utilisation de données théoriques ou pratiques dans l'étude expérimentale. De plus, en cas de données pratiques, les auteurs vérifient si l'approche proposée est implémentée dans l'organisation qui a fourni ces données.

Van den Bergh et collab. (2013) affirment que la communauté scientifique doit s'orienter vers les axes de recherche suivants :

- l'intégration des niveaux décisionnels relatifs à la prévision de demande, l'embauche et le licenciement, l'affectation des employés aux machines, la disponibilité des res-

- sources utilisées par le personnel et le transfert entre différentes localisations,
- la prise en considération de variantes plus réalistes du problème à savoir le placement de pauses, les qualifications des employés, la main d'œuvre à temps partiel et par intérim, les préférences des employés, etc,
 - l'introduction de travaux théoriques pour l'étude de la complexité du problème en fonction de l'utilisation de contraintes dures et souples dans les modèles utilisés,
 - la prise en compte de l'incertitude nécessitant l'utilisation de la programmation stochastique, l'analyse de la robustesse des calendriers obtenus dans un contexte déterministe le cas échéant en perturbant les intrants du problème, la reconstruction des horaires en se basant sur de nouvelles informations plus précises sur la demande ou la disponibilité des ressources humaines et matérielles (*rescheduling*),
 - l'adéquation des travaux basés sur des données réelles avec l'infrastructure informatique des organisations afin de permettre leur implémentation pratique,
 - et la comparaison avec les travaux existants dans la littérature pour évaluer la performance des approches proposées.

Ernst et collab. (2004b,a) proposent une revue plus ancienne de la littérature portant sur les problèmes de construction d'horaires. Les auteurs décomposent le problème global en trois niveaux décisionnels principaux : la modélisation de la demande, la construction des horaires anonymes et l'affectation des horaires aux employés. La construction des horaires anonymes inclut à son tour 3 niveaux élémentaires : la construction de cycles anonymes, la construction de quarts anonymes et l'affectation des quarts aux cycles. Lorsque le travail est décomposé en différents types, un niveau décisionnel s'ajoute qui est l'affectation des activités. La résolution directe du problème global n'est pas possible à cause de sa difficulté élevée. La résolution se fait donc en décomposant le problème global en différents sous-problèmes. Chaque sous-problème traite une combinaison des niveaux décisionnels identifiés précédemment. Ce processus décisionnel débute ainsi par la modélisation de la demande et se termine par la construction d'un horaire personnalisé pour chaque employé de l'organisation.

Dans le présent chapitre, nous proposons une nouvelle classification qui peut être perçue comme une généralisation du travail d'Ernst et collab. (2004b) à un contexte plus large qui est le contexte multidépartement introduit dans la Section 2.1 du chapitre précédent. Un nouveau niveau décisionnel s'ajoute qui correspond à l'affectation des départements aux périodes travaillées. Dans le reste du chapitre, on considère les acronymes suivants :

- Contexte multiactivité (MA pour dire *Multiple Activities*) : le travail dans l'organisation est composé de plusieurs activités. Une activité correspond à une unité élémentaire de travail qui consomme des ressources en effectifs et du temps. Prenons l'exemple d'un magasin de détail du prêt-à-porter. Parmi les activités qui peuvent être exécutées par les employés, on peut citer les actions suivantes : « accueillir les clients et leur annoncer

les promotions », « s'occuper de la caisse », « s'occuper des réclamations et des retours » et « réapprovisionner les rayons ».

- Contexte monoactivité (SA pour dire *Single Activity*) : le travail est composé d'une seule activité. On peut citer l'exemple des cabines de péage dans les autoroutes. La seule activité qu'il faut assurer est de recevoir le paiement des conducteurs. De plus, le contexte monoactivité peut être utilisé pour résoudre le problème de construction d'horaires dans un contexte multiactivité. En effet, le problème dans un contexte multiactivité peut être séparé en deux étapes. La première étape agrège la demande de toutes les activités et ramène le problème au contexte monoactivité. La deuxième étape correspond à l'affectation des activités aux périodes travaillées.
- Contexte multidépartement (MD pour dire *Multiple Departments*) : l'organisation du travail dans les grandes organisations requiert une division en différents départements. Un département constitue une section ou une unité de travail plus ou moins indépendante du reste de l'organisation. Reprenons l'exemple du magasin de détail. Imaginons que ce magasin peut vendre, en plus du prêt-à-porter, de l'électroménager et de l'agroalimentaire. Dans ce cas, il est divisé en différentes sections : « vêtements pour femmes », « vêtements pour hommes », « électroménagers », « agroalimentaire », « entrepôt », « point de cueillette » et « facturation ». Dans le département « électroménagers », les employés peuvent exécuter les activités suivantes : « recevoir les clients et les guider dans leur recherche », « traiter les réclamations et les problèmes techniques » et « réapprovisionner les rayons ». Dans ce cas, on dit que le département « électroménagers » est multiactivité. Si nous prenons l'exemple du département « facturation », la seule activité exécutée correspond à « s'occuper de la caisse ». Dans ce cas, le département « facturation » est dit monoactivité.
- Contexte monodépartement (SD pour dire *Single Department*) : l'intégralité du travail de l'organisation est rassemblée dans un seul département. Reprenons l'exemple du magasin du prêt-à-porter et supposons que ce magasin est petit. Il peut donc être géré sans le décomposer en sections. Le contexte monodépartement peut être exploité aussi pour résoudre le problème de construction d'horaires dans un contexte multidépartement. En effet, le problème dans un contexte multidépartement peut être séparé en 2 étapes. La première étape agrège la demande de tous les départements et ramène le problème au contexte monodépartement. La deuxième étape correspond à l'affectation des départements aux périodes travaillées.

La taxinomie que nous proposons divise l'ensemble des travaux examinés dans cette thèse en quatre classes :

- Classe SDSA : cet acronyme désigne un contexte monodépartement et monoactivité. L'organisation est formée par un seul département qui est monoactivité.

- Classe SDMA : cet acronyme désigne un contexte monodépartement et multiactivité. L'organisation est formée par un seul département qui est multiactivité.
- Classe MDSA : cet acronyme désigne un contexte multidépartement et monoactivité. L'organisation est formée par plusieurs départements qui sont tous monoactivités.
- Classe MDMA : cet acronyme désigne un contexte multidépartement et multiactivité. L'organisation est formée par plusieurs départements et l'un au moins de ces départements est multiactivité.

Dans le reste de ce chapitre, nous proposons une revue de la littérature portant sur des travaux relativement récents qui relèvent de la construction d'horaires. Pour une étude approfondie sur la complexité des ces problèmes, nous référons à [Brucker et collab. \(2011\)](#).

3.1 Problèmes de construction d'horaires de travail

La présente section décrit les niveaux décisionnels, les formes de flexibilité et les environnements de travail considérés dans le domaine de la construction d'horaires de travail en se référant à la littérature.

3.1.1 Niveaux décisionnels

Rappelons que tous les problèmes de construction d'horaires débutent par la modélisation de la demande. Cette étape consiste à prévoir le besoin de l'organisation en main d'œuvre afin d'atteindre un niveau de service donné. La demande en bien ou en service est d'abord estimée en utilisant des techniques de prévision. Cette demande prévue constitue les intrants qui sont convertis ensuite en une demande en effectifs. [Ertogral et Bamuqabel \(2008\)](#) modélisent la demande en agents dans un centre d'appel. Les intrants correspondent aux taux d'arrivée des appels. Ces taux sont supposés constants à l'intérieur d'une heure, varient entre les heures de la journée et suivent le même profil durant les jours de la semaine. Les auteurs utilisent les files d'attente et un modèle de simulation afin de déterminer le nombre requis d'agents pour chaque heure de la journée. Ce nombre doit permettre de répondre aux appels moyennant un temps d'attente maximal de 20 secondes dans au moins 80% des cas. L'extrait de cette étape correspond aux courbes de demande en agents durant chaque heure de la semaine.

Une fois le besoin en effectifs déterminé, on passe à construire un horaire admissible pour chaque employé de l'organisation. Lorsque les horaires construits sont anonymes, une affectation à posteriori de ces horaires aux employés est nécessaire. Dans le cas contraire, on parle de la construction d'horaires personnalisés. Ce problème permet de différencier les employés en fonction de certaines caractéristiques. Lorsque la main d'œuvre est hétérogène ou encore *flexible*, cette différenciation permet de tenir compte des qualifications des employés pour le travail à accomplir. Lorsque la main d'œuvre est homogène, la différenciation se

fait par exemple en fonction des préférences des employés pour les équipes de travail, les jours de repos, etc. Purnomo et Bard (2007) considèrent un problème de construction simultanée de cycles et de quarts personnalisés pour des infirmières. Il existe trois quarts de huit heures (des quarts de jour D, de soir E et de nuit N) et deux quarts de 12 heures (des quarts d'avant-midi AM et d'après-midi PM). Les règles de travail sont classées en deux types : dures et souples. Les règles souples correspondent à des profils de travail indésirables. Les auteurs identifient deux situations à éviter. La première situation arrive lorsqu'une infirmière travaille un jour situé entre deux jours de repos ou lorsqu'elle est assignée à un jour de repos situé entre deux jours travaillés. La deuxième situation arrive lorsqu'une infirmière reçoit différents quarts durant des jours de travail consécutifs (par exemple D/E/D). Le coût de violation d'un horaire personnalisé contenant a violations est égal à 2^{a-1} . Les auteurs mentionnent que ces coûts peuvent être exprimés par les infirmières afin de quantifier leurs propres préférences selon le degré de gravité des violations.

Construction d'horaires dans un contexte monodépartement

Dans un contexte monodépartement, le problème de construction d'horaires peut être énoncé comme suit : « Durant un horizon de planification donné, on cherche à construire un ensemble d'horaires qui permet de minimiser un coût relatif à la main d'œuvre sous les contraintes de demande et les règles de la convention collective ». La convention collective est un accord entre l'employeur et ses employés relatif aux conditions de travail et aux garanties sociales. Un horaire est *admissible* si et seulement s'il respecte les règles de la convention collective.

La définition de l'horaire dépend du problème traité. Il peut être réduit à une séquence de jours de travail et de repos. On parle de la construction de cycles. Il peut aussi être réduit à l'heure de début et la durée de travail durant un jour de l'opération et possiblement le placement d'une ou de plusieurs pauses. On parle de la construction de quarts. Lorsque les cycles et les quarts sont construits simultanément, on parle de la construction simultanée de cycles et de quarts. Dans tous ces problèmes, l'horaire peut être anonyme ou personnalisé. Dans un contexte multiactivité, un horaire est entièrement défini en spécifiant en plus de la séquence des jours de travail et de repos et des heures travaillées et de pause, la séquence d'activités qui le compose. Si ce niveau décisionnel est traité simultanément avec la construction des quarts personnalisés par exemple, on parle du problème de construction de quarts personnalisés dans un contexte multiactivité. Lorsque ce niveau décisionnel est traité séparément, on parle du problème d'affectation des activités (Lequy et collab., 2012a).

Construction d'horaires dans un contexte multidépartement

Dans un contexte multidépartement, le problème de construction d'horaires peut être énoncé comme suit : « Étant donné une organisation formée par un ensemble de départements, on cherche à construire un ensemble d'horaires permettant de couvrir les courbes de demande

de tous les départements tout en respectant les règles de travail et de transfert ».

Comme pour le contexte monodépartement, le contexte multidépartement implique la construction de cycles, de quarts, ou de cycles et de quarts simultanée avec transfert d'employés entre départements. En permettant de transférer les employés entre les départements, un nouveau niveau décisionnel doit être traité. Il correspond à l'affectation des départements aux périodes travaillées. Il faut donc décider durant chaque période travaillée, si l'employé sera transféré ou non et, dans l'affirmative, dans quel département d'accueil il sera transféré. Lorsque le département est multiactivité, il s'agit de décider aussi de l'activité que l'employé va réaliser.

À notre connaissance, tous les travaux rencontrés dans la littérature excluent la résolution directe du problème et adoptent une approche de résolution utilisant des techniques de décomposition afin de faire ressortir des sous-problèmes intermédiaires qui sont plus faciles à résoudre. Différentes approches ont été proposées pour modéliser et résoudre le problème. On va les exposer ultérieurement d'une façon plus détaillée. Dans le cas général, le problème est scindé en deux étapes : la construction des horaires en agrégeant la demande de tous les départements ensemble et l'affectation des départements aux horaires construits précédemment. Cette affectation se fait généralement en minimisant les transferts.

Bechtold (1988) propose des formulations impliquant plusieurs objectifs conflictuels de l'emploi de la main d'œuvre multidépartementale. L'auteur affirme que, d'une part, le transfert permet d'éviter les temps improductifs et, d'autre part, peut provoquer certaines complications. L'implication de plusieurs superviseurs peut créer des problèmes administratifs et des problèmes de motivation du personnel. La productivité de l'employé peut varier selon le département. Lorsqu'il s'agit d'une version multi-site du problème, des coûts additionnels peuvent être induits par le transport entre les sites. Lorsque les employés manipulent de l'argent (tels que les caissiers dans les banques et les magasins de vente), des problèmes de sécurité peuvent être liés au transfert. Bechtold (1988) souligne ainsi le compromis entre les avantages et les inconvénients de la main d'œuvre multidépartementale. Il considère trois objectifs triés selon différentes listes de priorité. Ces objectifs sont la minimisation du temps improductif, la minimisation du nombre d'employés transférés et la minimisation du nombre total d'employés.

Outre les contraintes de demande qui sont spécifiées pour chaque département, et éventuellement pour chaque activité de ce département s'il est multiactivité, des règles s'ajoutent possiblement pour définir les transferts admissibles. Le travail de Bard et Wan (2008) couvre deux niveaux décisionnels habituellement négligés : l'affectation de chaque employé à son département d'origine et le respect de certaines restrictions de transfert entre les départements. Le problème correspond à une application réelle dans un centre de traitement et de distribution du Service postal des États-Unis. La localisation des stations de travail (WSG)

et les motifs de supervision autorisent la main d'œuvre interne d'une station donnée à être transférée à un ensemble restreint de stations. Un employé est rattaché à un WSG (*home base*) lorsqu'il passe la durée de temps la plus élevée de son horaire de travail dans ce WSG. Une contrainte impose que le temps que l'employé passe dans n'importe quel WSG doit être au plus égal au temps qu'il passe dans son WSG d'origine.

Dans ce qui suit, on étudie une notion très importante dans tous les problèmes de construction d'horaires pour tous les contextes cités précédemment qui est la « *flexibilité* ».

3.1.2 Formes de flexibilité

La pertinence du problème de construction des horaires découle de la flexibilité de l'environnement de travail. D'un côté, cette flexibilité offre une certaine souplesse pour la construction des horaires. En effet, dans certains environnements « *figés* », tel est le cas des administrations publiques, l'ouverture des bureaux se fait généralement entre 8h00 et 17h00 durant les jours de semaines uniquement du lundi au vendredi. À l'opposé, dans les environnements « *flexibles* », le travail peut débuter à différents moments de la journée, avoir plusieurs durées, couvrir des jours de fins de semaine, etc.

D'un autre côté, une grande flexibilité mène à des problèmes plus difficiles à résoudre. Des travaux de recherche ont été faits pour proposer des approches permettant de profiter des avantages que procure la flexibilité en outrepassant la complexité qui en découle. Dans ce qui suit, on présente certains exemples choisis pour chacun des contextes SDSA, SDMA, MDSA et MDMA.

Formes de flexibilité dans un contexte monodépartement

Pour le contexte SDSA, les formes de flexibilité retrouvées sont relatives à une littérature plus ou moins récente portant sur les quarts, les cycles ou la combinaison des deux. La flexibilité qui porte sur les quarts a été largement étudiée dans les travaux les moins récents. Prenons l'exemple de la flexibilité portant sur les pauses. On distingue deux cas, « *monopause* » et « *multipause* ». Bechtold et Jacobs (1990); Rekik et collab. (2004); Addou et Soumis (2007) s'intéressent au cas monopause et le traitent au biais de modèles spécifiques. Dans ces travaux, un type de quart est défini par un début, une durée et une fenêtre de temps durant laquelle la pause peut débuter. Ainsi, un quart admissible est obtenu en plaçant la pause dans la fenêtre de temps du type de quart qui lui correspond.

Plus tard, Rekik et collab. (2010) introduisent le nouveau concept de « *pauses décomposables* » pour permettre une flexibilité plus élevée sur la définition des pauses. Les auteurs traitent un problème de construction de quarts anonymes multipauses. Une pause décomposable peut être divisée en plusieurs sous-pauses. La somme des durées des sous-pauses est égale à la durée de la pause. Pour éviter que les sous-pauses soient de durée très courte ou très longue,

chaque pause décomposable est caractérisée par des restrictions sur la durée minimale et maximale des sous-pauses qui lui sont associées. De plus, un quart peut recevoir au plus une pause décomposable. Les auteurs ont montré les mérites des pauses décomposables par rapport aux pauses standards. Ils comparent la taille de la main d'œuvre nécessaire en utilisant trois pauses de durées fixes et en utilisant une pause décomposable. Ils réussissent à générer un gain considérable en nombre d'employés pour certaines instances qui atteint jusqu'à 8.7%.

Parallèlement à ces travaux, d'autres formes de flexibilité portant sur les cycles sont étudiés. Alfares (2003) traite un problème de construction de cycles anonymes. Chaque cycle est défini sur une semaine par quatre jours de travail et trois jours de repos. Il existe deux types de restrictions. Le premier type porte sur le placement des jours de repos durant la semaine. Ces jours doivent être égaux à 3 et deux au moins parmi les trois doivent être consécutifs. Le deuxième type porte sur l'attribution d'une proportion minimale de fins de semaine non travaillées (*weekend work frequency constraints*). L'auteur considère deux cas qui diffèrent selon que la fin de semaine doit être complètement non travaillée ou peut être partiellement non travaillée. L'utilisation des fins de semaine partiellement non travaillées est une alternative permettant d'augmenter la flexibilité du placement des jours de repos par rapport au cas standard utilisant uniquement des fins de semaine complètement non travaillées.

Pour les problèmes de construction simultanée de cycles et de quarts, la flexibilité peut porter par exemple sur les heures de début des quarts. Rekik et collab. (2004) utilisent le concept de la *bande* d'heures de début des quarts (*start-time band*). Cette bande est un intervalle de temps à l'intérieur duquel les heures de début de travail peuvent varier d'un jour à l'autre pour un horaire donné. Elle constitue ainsi une solution intermédiaire entre un cycle rempli par des quarts de travail débutant nécessairement à une position figée de la journée d'opération et entre un cycle rempli par des quarts pouvant commencer à n'importe quelle période de la journée.

Plus tard, Seçkiner et collab. (2007) utilisent le concept de la « *semaine de travail comprimée* » adopté par certaines organisations pour résoudre un problème de construction simultanée de cycles et de quarts. Quoique la demande soit spécifiée par jour, nous considérons qu'il s'agit d'une construction simultanée de cycles et de quarts parce que l'approche proposée permet de décider, non seulement des cycles, mais aussi des durées des quarts. Une semaine de travail comprimée consiste à allonger les durées de travail quotidiennes pour bénéficier d'un plus grand nombre de jours de repos. Un cycle est formé par des quarts de durées égales durant tous les jours de travail. Deux versions sont utilisées et comparées : la première suppose que tous les cycles utilisent la même durée et une autre variante plus flexible considère que les cycles utilisent différentes durées. Les employés recevant les quarts longs ont droit à plus de jours de repos. Si le cycle est formé par des durées de travail de 8h, l'employé doit recevoir deux jours de repos par semaine. S'il est formé par des durées de travail

de 9h, il doit recevoir trois jours de repos. Finalement, s'il est formé par des durées de travail de 10h, il doit recevoir quatre jours de repos. C'est Hung (2006) qui a introduit le concept de la semaine de travail comprimée ignoré précédemment dans la littérature pour résoudre un problème de construction de cycles anonymes. L'auteur souligne les gains que peut procurer l'utilisation de la semaine de travail comprimée sur les coûts de la main d'œuvre, en plus de la construction de cycles plus attractifs pour les employés.

Pour le contexte SDMA, la flexibilité porte, en plus de toutes les formes précédemment étudiées, sur les durées des activités. En effet, une pratique consiste à définir une durée minimale et maximale requises pour l'exécution des activités. Au lieu que la durée d'une activité soit constante, elle peut varier dans un intervalle spécifique en respectant ces restrictions (Lequy et collab., 2012b). Dans ce contexte, la flexibilité peut porter aussi sur la main d'œuvre. Cette main d'œuvre est flexible (*multi-skill*) si elle possède les qualifications requises pour exécuter différentes activités. Ertogral et Bamuqabel (2008) considèrent des employés dans des centres d'appel possédant des qualifications pour servir les clients en plusieurs langues. Pour étudier l'effet de la main d'œuvre flexible, ils ont considéré deux scénarios : le premier scénario utilise exclusivement des employés qualifiés pour une seule langue et le deuxième scénario introduit les employés qualifiés pour les deux langues. Les résultats expérimentaux affirment que l'utilisation des employés flexibles peut diminuer considérablement la taille de la main d'œuvre.

Formes de flexibilité dans un contexte multidépartement

Le contexte multidépartement hérite toutes les formes de flexibilité caractérisant le contexte monodépartement. Les nouvelles formes de flexibilité qui sont spécifiques à ce contexte portent essentiellement sur la définition des transferts admissibles.

Afin de permettre le transfert des employés entre les départements, Bechtold (1988) précise que 1) le travail exécuté dans différents départements doit être de même nature exigeant ainsi des qualifications équivalentes ou que 2) les employés doivent être formés pour travailler dans différents départements exigeant des expertises distinctes. Comme pour le cas du contexte monodépartement et multiactivité, la main d'œuvre est considérée flexible si elle peut travailler dans plus qu'un département permettant ainsi le partage de ressources. Citons par exemple les travaux de Bard et Wan (2008) pour le déplacement entre stations de travail.

La deuxième forme de flexibilité porte sur la nature des transferts. Le transfert est plus ou moins flexible si les règles imposent qu'il soit accompli durant toute la journée de travail. Il s'agit donc d'un transfert de quart au complet. En d'autres termes, l'employé ne peut changer de département de travail durant son cycle que d'un jour à l'autre (Mabert et Raedels, 1977; Bechtold, 1988; Al-Yakoob et Sherali, 2007). Au contraire, Bard et Wan (2008) consi-

dèrent un haut degré de flexibilité portant sur les transferts. Les employés peuvent se déplacer à n'importe quelle période de la journée et pour une durée quelconque. Aucune restriction n'est donc imposée sur le début et la durée des blocs de transfert. Nous proposons dans cette thèse une situation intermédiaire. L'employé a la possibilité de passer le quart au complet dans l'un de ses départements admissibles ou de se déplacer une fois de son département d'origine vers l'un des départements d'accueil ou vice-versa donnant lieu à un quart à deux blocs. Une restriction sur la durée minimale des blocs est imposée. Le transfert peut se faire à n'importe quelle période dans la mesure du respect de la durée minimale des blocs avant et après le déplacement entre les départements.

3.1.3 Environnement de travail

Il existe deux environnements de travail : continu et discontinu. L'environnement est dit continu lorsque les entreprises opèrent sept jours par semaine, 24 heures par jour. Ce cas est rencontré fréquemment dans le secteur des services tels que les centres d'appel, les cabines de péage autoroutier, l'assistance aéroportuaire, les guichets des gares, etc. La complexité qui en découle est que le problème n'est pas séparable par jour. Il doit donc être résolu pour tous les jours de l'horizon simultanément. Parmi les travaux qui relèvent de la construction d'horaires dans un environnement continu, on cite Baker et Magazine (1977); Rekik et collab. (2004); Brusco et Jacobs (1998); Brusco et Johns (2011).

L'environnement de travail dans le domaine de la construction des horaires peut être cyclique ou acyclique. Dans le cas cyclique, la demande en main d'œuvre est caractérisée par un profil qui se répète à tous les cycles à l'intérieur de l'horizon de planification. Des horaires cycliques ou rotatifs sont construits durant un intervalle de temps et sont assignés aux employés d'une manière répétitive à l'intérieur de l'horizon d'un intervalle à l'autre. Purnomo et Bard (2007) traitent un problème de construction simultanée de cycles et de quarts personnalisés pour des infirmières. L'approche est générique et peut être appliquée à d'autres cas. Le problème possède certaines spécificités. D'abord, il s'agit d'un problème tactique (à moyen terme). On cherche par exemple à construire les horaires des infirmières pour les quatre semaines à venir. Les horaires à construire sont cycliques. Autrement dit, ils se répètent périodiquement tant que la demande en effectifs présente les mêmes variations d'un intervalle de planification à l'intervalle suivant.

Musliu et collab. (2002) traitent un problème de construction d'horaires rotatifs anonymes. Tous les employés possèdent le même horaire de base mais les horaires diffèrent par leur initialisation. Les auteurs soulignent que les horaires rotatifs ont un impact positif sur le rendement, la santé et la satisfaction des employés. Le nombre d'employés est connu. L'horaire est construit généralement sur une base hebdomadaire ou à la quinzaine de jours. Trois types de quart sont considérés : de jour, de soir et de nuit. Les courbes de demande donnent le nombre d'employés requis pour chaque type de quart et durant chaque jour. Les auteurs

proposent une approche de résolution à quatre étapes. Ils commencent par choisir les durées des blocs de travail. Un bloc de travail est constitué d'un nombre de jours consécutif de travail. Ensuite, les blocs de travail sélectionnés précédemment sont ordonnés. La séquence obtenue permet de maximiser le nombre de fins de semaine non travaillées, de bien distribuer ces fins de semaine pour qu'elles soient éloignées et non consécutives et de maximiser le nombre de fins de semaine non travaillées prolongées. La troisième étape consiste à énumérer toutes les affectations admissibles de quarts de jour, de soir et de nuit pour chaque bloc de jours travaillés. Il est interdit de passer d'un quart de nuit à un quart de jour, d'un quart de nuit à un quart de soir et d'un quart de soir à un quart de jour. Il existe également un nombre minimal et maximal de quarts de jour travaillés consécutivement. La même contrainte s'applique pour les quarts de soir et de nuit. La dernière étape correspond à l'affectation des séquences de quart aux blocs travaillés de manière à couvrir les contraintes de demande.

Une autre classification peut être élaborée en divisant les travaux de construction des horaires de travail selon les qualifications de la main d'œuvre. Nous identifions trois types de main d'œuvre : *hiérarchique*, *non hiérarchique* et *mixte*. Le premier type consiste à partitionner les employés en catégories et de trier ces catégories selon un ordre croissant de leurs compétences. Ainsi, une catégorie d'employés ne peut exécuter le travail requis pour une autre catégorie que si elle est plus compétente (Billionnet, 1999; Seçkiner et collab., 2007).

La classification des employés selon leur catégorie se fait en se basant sur leurs qualifications. Supposons par exemple qu'il existe deux types de travail : S1 et S2. Il existe donc trois catégories d'employés : les employés qualifiés pour le type S1 uniquement (W1), ceux qualifiés pour le type S2 uniquement (W2) et ceux qualifiés pour les deux types S1 et S2 ensemble (W3). Il existe deux sous-ensembles de main d'œuvre hiérarchique : W3 et W1 puis W3 et W2. Les employés de catégorie W3 peuvent donc exécuter le travail des employés de catégories W1 et W2. Si la main d'œuvre renferme les catégories W1 et W2 alors elle est dite non hiérarchique car il n'existe aucune relation d'hiérarchie entre ces deux catégories. Si la main d'œuvre renferme les catégories W1, W2 et W3 alors la main d'œuvre est mixte. Les auteurs définissent une matrice des compétences, appelée « *worker type-skill matrix* », permettant de construire des horaires pour une main d'œuvre mixte.

En conclusion, les travaux sur la construction d'horaires évoluent principalement selon les deux axes suivants : l'augmentation de la flexibilité et la complication des réglementations. Le premier axe s'explique par les gains considérables des coûts de la main d'œuvre lorsque le planificateur dispose d'une flexibilité élevée pour la définition des horaires. Le deuxième axe correspond au cadre législatif et réglementaire qui ne cesse d'évoluer pour garantir la qualité de vie des employés et la conciliation travail et vie personnelle. Ces axes amènent de nouveaux défis pour la communauté scientifique. Une autre constatation correspond à un intérêt grandissant aux problèmes d'horaires impliquant des jours multiples. En effet, de telles approches permettent d'examiner la qualité des horaires sur un horizon moyennement

long pour pouvoir tenir compte de l'historique des quarts personnalisés et éviter le partage inéquitable de la charge de travail entre les employés. Finalement, les travaux qui relèvent d'une littérature récente s'intéressent de plus en plus au contexte multidépartement avec partage de ressources entre départements, divisions, sites, postes de travail, etc.

Dans le reste, passons à classer l'ensemble des travaux rencontrés selon les quatre contextes que nous avons défini : SDSA, SDMA, MDSA et MDMA. Pour chaque contexte, nous listons les travaux selon un ordre croissant du ou des niveaux décisionnels traités et nous accordons une grande importance aux approches de modélisation et de résolution utilisées.

3.2 Construction d'horaires dans un contexte monodépartement

La construction des horaires de travail dans un contexte monodépartement considère les différents départements de l'entreprise comme des entités indépendantes. La construction des horaires de travail dans un tel contexte consiste à traiter le problème séparément pour chaque département. Dans ce qui suit, nous étudions les travaux qui traitent le problème de construction des horaires de travail dans un département donné, c'est à dire dans un contexte monodépartement. Ces travaux sont divisés en deux classes distinctes selon que le département soit monoactivité ou multiactivité : SDSA et SDMA.

3.2.1 Contexte monodépartement et monoactivité

Les premiers travaux de construction des horaires de travail s'adressent au contexte monodépartement et monoactivité. La communauté scientifique s'est orientée plus tard vers les autres classes permettant de capter des aspects plus réalistes du problème de construction des horaires. Par conséquent, l'ensemble des travaux appartenant aux classes SDMA, MDSA et MDMA sont imprégnés des approches de modélisation et de résolution initialement proposées pour la classe SDSA. Pour cette classe, nous distinguons principalement trois types de problèmes : la construction des cycles, la construction des quarts et la construction simultanée des cycles et des quarts.

Construction des cycles

La construction des cycles anonymes consiste à spécifier une séquence de jours de travail et de repos durant un horizon de planification généralement de quelques semaines. Lorsque cette construction se fait simultanément en spécifiant les employés recevant les cycles, il s'agit d'une construction de cycles personnalisés.

Baker et Magazine (1977) examinent un problème de construction de cycles anonymes dans un environnement de travail continu (sept jours de travail par semaine) et cyclique. La demande est déterminée par deux paramètres N et n avec N le nombre d'employés minimal requis durant les jours de semaine et n le nombre d'employés minimal requis durant les

jours de fin de semaine. La taille de la main d'œuvre minimale est d'abord calculée à partir d'une formule spécifique selon la stratégie de gestion du personnel. Ensuite, un algorithme constructif permet de spécifier les cycles.

Dans un cadre similaire, Narasimhan (1997) applique la même approche de résolution qui calcule la taille de la main d'œuvre minimale puis construit les cycles. Ces cycles doivent vérifier les deux réglementations suivantes : 1) chaque employé doit recevoir deux jours de repos par semaine contenant au moins une proportion A sur B de fin de semaines non travaillées et 2) chaque employé peut recevoir au plus cinq jours de travail consécutifs.

Alfares (1998) considère le problème de construction de cycles anonymes dans un environnement cyclique où deux jours de repos par semaine doivent être placés. L'auteur résout le problème en deux étapes. La main-d'œuvre minimale requise est d'abord calculée. Ensuite, un modèle linéaire continu est résolu fournissant une solution optimale entière.

Alfares (2003) traite un problème de construction de cycles anonymes en appliquant le concept de la semaine de travail comprimée. Il définit ainsi un problème de construction de semaine de travail flexible comprimée. L'auteur propose de nouveaux modèles en nombres entiers. Il définit une procédure de résolution à deux étapes. La première étape sert à calculer le nombre minimal d'employés requis. La deuxième étape consiste à résoudre une version modifiée des modèles. Cette version est obtenue en incorporant une contrainte additionnelle sur la taille de la main d'œuvre calculée lors de la première étape. L'auteur montre à partir de 202 instances que la procédure de résolution à deux étapes nécessite en moyenne un temps de calcul neuf fois moins élevé que la résolution directe des modèles.

Plus tard, Elshafei et Alfares (2008) utilisent la programmation dynamique pour résoudre le problème de construction de cycles anonymes. Les jours de l'horizon sont modélisés par les étapes et les cycles admissibles sont modélisés par les états. L'objectif est de minimiser le coût total de la main d'œuvre. La contribution principale de ce travail réside dans la définition du coût qui dépend de la séquence de jours de travail et de repos. Par exemple, le coût d'affecter un jour de fin de semaine travaillé à un cycle diffère selon que ça soit le premier ou le deuxième jour de fin de semaine travaillé. Cette structure de coût peut être modélisée par l'approche explicite comme un problème en nombres entiers alors que l'approche implicite ne permet pas de modéliser de telles structures. Les auteurs affirment que la programmation dynamique se prête bien avec ce type de problème. Elle permet de construire le cycle de travail jour par jour en pénalisant le travail durant les jours de fin de semaine selon la séquence partielle déjà construite.

Plus tard, Costa et collab. (2006) traitent le problème de construction de cycles personnalisés. L'environnement de travail est acyclique. La demande fluctue d'un jour à l'autre. La taille de la main d'œuvre est connue et le nombre de jours travaillés est assigné a priori à chaque employé. Les auteurs proposent des algorithmes polynomiaux pour la résolution du

problème.

Construction des quarts

La construction des quarts anonymes consiste à spécifier le début et la durée de travail durant une journée d'opération, et possiblement le placement d'une ou de plusieurs pauses. Lorsque cette construction se fait simultanément en spécifiant les employés recevant les quarts, il s'agit d'une construction de quarts personnalisés.

L'ensemble des travaux de construction de quarts anonymes implique deux approches de modélisation différentes : explicite et implicite. L'approche explicite se base sur une formulation de couverture généralisée (*Generalized set covering*). C'est Dantzig (1954) qui était le premier à introduire cette formulation pour la construction de quarts anonymes. L'auteur énumère tous les quarts admissibles qui respectent les réglementations. Il associe une variable de décision à chaque quart admissible. Cette variable représente le nombre d'employés recevant le quart. L'objectif est de minimiser le nombre total d'employés sous les contraintes de demande. Dantzig (1954) a appliqué ce modèle pour la construction de quarts anonymes à des employés dans des cabines de péage. La résolution de la relaxation linéaire produit généralement des solutions entières. Dans le cas contraire, une heuristique est utilisée pour arrondir les valeurs fractionnaires à l'entier le plus proche. La solution ainsi obtenue n'est pas forcément réalisable.

La formulation de type couverture généralisée a été utilisée par plusieurs chercheurs, d'où l'émergence d'une approche de modélisation dite explicite. Comme son nom l'indique, cette approche énumère l'ensemble de tous les horaires admissibles et associe une variable de décision explicite à chaque horaire. Supposons que l'approche est utilisée pour formuler un problème de construction de quarts anonymes. Supposons de plus que l'organisation utilise plusieurs heures de début, plusieurs durées et plusieurs placements de pause. Une variable de décision doit être définie pour chaque possibilité de choisir une heure de début, une durée et un placement de pause admissible. Il est clair donc que l'approche implique un nombre élevé de variables de décision qui augmente dramatiquement avec la flexibilité. Par conséquent, bien que cette approche ait la vocation de pouvoir modéliser toutes les formes de flexibilité, elle engendre des modèles de grande taille difficiles à résoudre. L'aspect de résolution du problème représente dans ce cas le défi à soulever en proposant des approches exactes ou heuristiques identifiant de bonnes solutions dans des temps raisonnables.

Une autre manière pour outrepasser la difficulté inhérente à la taille des formulations explicites est de modéliser certaines formes de flexibilité d'une manière implicite. L'approche implicite offre une alternative plus viable lorsqu'elle est capable de modéliser les formes de flexibilité qui se présentent. Bechtold et Jacobs (1990) sont les premiers à introduire un modèle implicite intéressant par rapport au modèle de couverture généralisée pour la construc-

tion de quarts anonymes avec placement de pauses. Au lieu d'énumérer des quarts, les auteurs énumèrent des types de quart. Un type de quart est défini par son début, sa durée et une fenêtre de temps (*break window*) durant laquelle la pause peut être attribuée. Au lieu d'énumérer toutes les possibilités de placement de pauses, les auteurs considèrent deux types de variables : les variables de types de quart et les variables de pauses. Ils introduisent aussi des contraintes qui lient les variables de pauses aux variables de types de quart. Ces contraintes de type « *forward* » et « *backward* » permettent de lier les pauses aux types de quart qui lui sont admissibles. Une pause est admissible pour un type de quart si elle appartient à la fenêtre de pause de ce type de quart. Des contraintes additionnelles égalisent le nombre de types de quart et le nombre de pauses sélectionnés dans la solution optimale. La reconstitution de quarts explicites se fait par une simple affectation a posteriori des pauses sélectionnées aux types de quarts sélectionnés qui lui sont admissibles. Le modèle de [Bechtold et Jacobs \(1990\)](#) pour le placement implicite de pauses suppose l'absence d'extra-chevauchement (*extraordinary overlap*). Les auteurs définissent un extra-chevauchement par le fait que l'intervalle de pauses d'un type de quart donné soit strictement inclus dans l'intervalle de pauses d'un autre type de quart. La résolution du problème se fait par une procédure de séparation et d'évaluation progressive avec plans de coupes. [Bechtold et Jacobs \(1996\)](#) démontrent l'équivalence entre leur modèle implicite et le modèle explicite de couverture généralisée sous l'hypothèse de contiguïté des intervalles de pauses et d'absence d'extra-chevauchement. Tous ces concepts seront détaillés dans le Chapitre 4.

L'hypothèse restrictive de non extra-chevauchement a été relaxée plus tard par [Addou et Soumis \(2007\)](#). Les auteurs démontrent que, même en présence d'extra-chevauchement, l'ajout d'un ensemble restreint de contraintes au modèle de [Bechtold et Jacobs \(1990\)](#) rend la formulation implicite applicable. Il s'agit d'ajouter uniquement une seule contrainte par extra-chevauchement présent. Plus encore, ils démontrent qu'en absence d'extra-chevauchement, aucune des contraintes *forward* et *backward* n'est redondante. Par contre, en présence d'extra-chevauchement et suite à l'ajout des contraintes de [Addou et Soumis \(2007\)](#), certaines des contraintes *forward* et *backward* deviennent redondantes. Les auteurs proposent alors un ensemble restreint de contraintes spécifiques puis démontrent que la réduction proposée débouche en un ensemble minimal de contraintes.

D'autres travaux considèrent un problème plus flexible pour la construction des quarts anonymes avec placement de pauses multiples. [Rekik et collab. \(2010\)](#) utilisent le concept de contraintes *forward* et *backward* pour modéliser deux formes de flexibilité : les pauses décomposables et les restrictions sur la durée de travail consécutif. Les quarts ainsi obtenus sont caractérisés par une bonne distribution des périodes travaillées et de pause. Les auteurs soulignent que le concept des pauses décomposables qu'ils introduisent permet un gain en nombre d'employés de 6% pour des profils réels de demande. Pour des profils légèrement perturbés, le gain atteint 8.7%.

Quoique l'approche implicite soit plus difficile que l'approche explicite de point de vue modélisation, son avantage majeur consiste à une réduction significative de la taille des modèles. Ceci permet alors d'utiliser des méthodes de résolution exactes plutôt que des heuristiques. Néanmoins, certains environnements de travail sont très complexes. Même pour les problèmes de construction de quarts anonymes, la résolution exacte demeure difficile et requiert l'emploi de techniques adaptées. Rekik et collab. (2008) considèrent un environnement continu hautement flexible et proposent deux méthodes différentes pour résoudre un problème de construction de quarts anonymes avec pauses multiples. La continuité de l'environnement requiert la résolution du problème sur plusieurs jours. La première méthode proposée est exacte. C'est une généralisation du concept de branchement local proposé par Fischetti et Lodi (2003). Le concept de branchement local utilise des coupes (*local branching cuts*) permettant d'explorer en priorité un sous-arbre de branchement potentiellement prometteur. La deuxième méthode est basée sur une heuristique d'horizon roulant adaptée à un environnement d'opération continu. Les tests expérimentaux faits sur des instances réelles fournies par une agence de contrôleurs aériens permettent de conclure que le branchement local proposé domine souvent le Branch-and-Bound de CPLEX. L'heuristique d'horizon roulant est la moins performante pour les instances plus ou moins faciles. Pour les instances plus difficiles, CPLEX et la méthode de branchement local ne peuvent pas identifier une solution entière. Dans ce cas, l'heuristique d'horizon roulant devient pertinente du moment qu'elle permet de fournir de bonnes solutions en un temps raisonnable.

Notons que la difficulté de représenter implicitement certaines données du problème fait en sorte que l'approche explicite reste pertinente et plus capable de couvrir un ensemble plus large d'applications.

Construction simultanée des cycles et des quarts

La séparation entre la construction des quarts et la construction des cycles a pour objectif principal de simplifier le problème global. Néanmoins, cette séparation altère la qualité des horaires obtenus. Les chercheurs ont donc essayé d'intégrer les deux problèmes et de les traiter en un seul problème de construction simultanée de cycles et de quarts. La définition de l'horaire devient à la fois une spécification des jours travaillés et de repos généralement durant une semaine et des périodes travaillées et de pause durant chaque jour de travail. Cette intégration devient très pertinente lorsqu'il s'agit d'environnements d'opération continus utilisant des quarts chevauchant entre les jours consécutifs.

La majorité des travaux récents dans un contexte SDSA s'adresse à la construction simultanée des cycles et des quarts et hérite les propriétés du problème de construction des quarts et du problème de construction des cycles. La majorité des travaux rencontrés emploient une approche de modélisation implicite (Topaloglu et Ozkarahan, 2003; Bard et collab., 2003; Rekik et collab., 2004; Brusco et Johns, 2011). Ces travaux diffèrent notamment par les formula-

tions implicites utilisées. Dans un cadre plus général, certains travaux récents emploient des approches hybrides qui combinent l'approche explicite et implicite dans le même modèle (Ni et Abeledo, 2007). Dans ce qui suit, nous détaillons ces travaux.

Une pratique courante dans les problèmes de construction simultanée de cycles et de quarts consiste à limiter la flexibilité afin de réduire la complexité du problème. Nous citons par exemple la détermination des heures de début des quarts formant un même cycle. Un problème soulevé dans la littérature s'intéresse à la sélection appropriée de ces heures de début. Les travaux effectués pour la sélection de ces heures de début sont généralement de nature séquentielle. D'abord, un sous-ensemble restreint d'heures de début est retenu. Ensuite, les cycles sont construits en se basant uniquement sur le sous-ensemble identifié à l'étape précédente.

Rekik et collab. (2004) proposent des modèles implicites pour le problème de construction simultanée de cycles et de quarts anonymes avec pauses et bandes d'heures de début. Les auteurs adaptent le concept de contraintes forward et backward à une large panoplie d'environnements d'opération. Ils proposent une étude approfondie de l'origine de ces contraintes et des conditions nécessaires à leur application. Les auteurs prouvent que sous les hypothèses d'absence d'extra-chevauchement et de contiguïté des fenêtres de pauses, les contraintes forward et backward sont des coupes de réalisabilité de Benders suffisantes pour garantir la réalisabilité de problèmes de transport ayant une structure particulière. La preuve établie permet, dans le cas que les contraintes forward et backward ne sont pas suffisantes, de proposer une décomposition de Benders qui peut être appliquée au cas général. Les auteurs développent une reformulation basée sur des contraintes forward et backward adaptées qui permet de réduire significativement la taille du problème.

Plus tard, Brusco et Johns (2011) considèrent un problème en nombres entiers, noté RSTP, qui se base sur une formulation implicite pour la construction simultanée de cycles et de quarts dans un environnement continu et employant une main d'œuvre hétérogène. La contribution de leur travail consiste à proposer une procédure qui intègre la sélection des temps de début et la construction des cycles anonymes. Cette procédure, notée TSCP, utilise la recherche taboue pour la sélection des temps de début potentiellement bons et une méthode de génération de plans de coupes (*cutting plan*) pour la construction des cycles. La TSCP est comparée à la résolution directe du RSTP en utilisant CPLEX, puis à une heuristique à deux étapes proposée par Brusco et Jacobs (1998), et ceci en se basant sur des instances pour des employés d'aéroport précédemment traités par Brusco et Jacobs (1998). Les auteurs concluent que lorsque les profils de demande sont cohérents durant les jours de la semaine d'opération, toutes les méthodes sont performantes. Pour les auteurs, des profils de demande cohérents signifient une tendance similaire de la demande au fil des jours de la semaine. Par contre, lorsqu'une incohérence des courbes de demande d'un jour à l'autre de la semaine d'opération est observée, la nouvelle méthode TSCP devient plus performante

que la résolution exacte du RSTP et que l'heuristique de Brusco et Jacobs (1998). Les auteurs concluent qu'une perspective future de leurs travaux serait de développer une procédure de résolution exacte pour le RSTP. L'heuristique TSCP est utilisée pour identifier une solution réalisable. Ensuite, CPLEX l'utilisera comme solution initiale lui permettant éventuellement d'identifier rapidement une solution optimale pour le problème.

Bard et collab. (2003) traitent un problème de construction simultanée de quarts et de cycles anonymes dans un centre de traitement et de distribution du Service postal des États-Unis. Il s'agit d'un problème de planification à long terme permettant de calculer la taille et la composition optimale d'une main d'œuvre homogène. L'horizon de planification est formé d'une semaine discrétisée en des périodes de 30 minutes. Le problème est résolu sur une semaine et les horaires, ainsi obtenus, seront affectés individuellement aux employés d'une manière cyclique d'une semaine à l'autre. L'horizon de planification est discontinu car les quarts considérés ne chevauchent pas des jours consécutifs. Le problème traite d'une manière simultanée les niveaux décisionnels relatifs à la sélection des heures de début et des durées de travail, au placement de pause dépendamment de la durée des quarts et au placement de deux jours de repos à l'intérieur de la semaine. Il existe deux types d'employés : à temps complet et à temps partiel. Les quarts constituant un même horaire doivent avoir la même heure de début et la même durée. Lorsque la durée du quart excède les six heures, une pause repas de 30 minutes doit être affectée. Un ratio minimal qui mesure le nombre d'employés à temps complet par rapport au nombre d'employés à temps partiel doit être respecté. Étant donné que l'effectif à temps partiel est payé moins cher que l'effectif à temps complet, l'incorporation de ce ratio dans le modèle permet de ne pas dépasser la borne supérieure sur la taille de l'effectif à temps partiel selon la convention collective. La formulation implicite proposée permet le placement d'une pause repas à l'intérieur des quarts et le placement de deux jours de repos par semaine. Le modèle implicite proposé est basé principalement sur deux concepts. Le premier concept correspond à l'utilisation des contraintes de type forward et backward pour lier les variables de type de quarts aux variables de pauses admissibles (Bechtold et Jacobs, 1990). Un type de quart est défini par son heure de début et sa durée. Le deuxième concept correspond à la définition de deux bornes inférieures sur le nombre d'employés recevant chaque type de quart permettant de garantir la réalisabilité des cycles (Burns et Carter, 1985).

Une fois le modèle en nombres entiers est résolu par CPLEX, Bard et collab. (2003) procèdent à la construction a posteriori des horaires. Deux post-traitements sont effectués selon un ordre aléatoire pour placer les pauses à l'intérieur des quarts et les jours de repos à l'intérieur des cycles. Dans la partie expérimentale, les auteurs considèrent plusieurs scénarios. Les auteurs concluent que lorsque le ratio diminue, c'est-à-dire lorsque la borne supérieure de la main d'œuvre partielle augmente, le coût de la main d'œuvre diminue et vice-versa. De plus, lorsque les deux jours de repos doivent être consécutifs, le coût de la main d'œuvre augmente

considérablement de 7.6%. Lorsque les heures de début d'un horaire peuvent varier dans un intervalle de temps délimité au lieu qu'elles soient figées, les expérimentations montrent qu'il y a une amélioration légère du coût de la main d'œuvre.

Ni et Abeledo (2007) proposent une approche de résolution pour le problème de construction simultanée de cycles et de quarts. Les auteurs affirment que le modèle est une approche intermédiaire qui combine les concepts de modélisation implicite et explicite. Deux ensembles sont initialement énumérés : les quarts admissibles et les profils de début admissibles. Un profil de début est défini par l'ensemble des débuts des quarts composant l'horaire. Dans le modèle, les quarts et les profils de début sont énumérés explicitement alors que la reconstitution de l'horaire explicite se fait implicitement au biais de contraintes qui lient les quarts aux profils de début. Deux types de variables de décision sont considérés : les variables de quarts (qui désignent le nombre d'employés travaillant un quart) et les variables de cycles ayant un profil de début donné (qui désignent le nombre d'employés associés à ce cycle). La reconstitution des horaires explicites se fait au biais d'un post-traitement qui affecte les quarts aux heures de début avec une simple règle du « premier arrivé premier servi ». Cette approche permet de représenter le modèle dans un contexte continu ou discontinu lorsque les règles sur les quarts et les temps de début sont très compliquées. Une méthode de Branch-and-Price est proposée pour résoudre le problème d'une manière efficace. Cette approche est testée sur les instances les plus larges retrouvées dans la littérature et donne de bonnes performances. Cette méthode peut être généralisée pour des scénarios plus complexes traitant le placement de pauses dans les quarts.

Burke et collab. (2010) traitent un problème de construction simultanée de quarts et de cycles personnalisés pour des infirmières. Les auteurs proposent une approche hybride combinant la programmation en nombres entiers et la recherche dans des voisinages multiples (*Variable Neighbourhood Search, VNS*). L'environnement d'opération est caractérisé par un nombre élevé de réglementations. Les quarts considérés sont de type jour, soir et nuit. Les auteurs visent alors à placer ces quarts durant les jours de la semaine de manière à respecter la demande et l'admissibilité des cycles. Un problème en nombres entiers est proposé impliquant des objectifs multiples. Ce problème permet de modéliser toutes les contraintes dures et un sous-ensemble de contraintes souples portant sur certaines règles de travail. Une fois le problème en nombres entiers résolu, l'heuristique VNS est utilisée pour améliorer la solution trouvée. Cette heuristique se concentre sur la satisfaction des contraintes souples exclues du problème en nombres entiers. L'approche est comparée à un algorithme génétique commercial et à une heuristique VNS classique en utilisant des instances réelles. Les résultats montrent que l'approche hybride utilisant la programmation entière et la VNS permet d'améliorer la performance. Les auteurs concluent que cette méthodologie peut être généralisée à d'autres problèmes d'allocation de ressources impliquant un nombre élevé de contraintes.

3.2.2 Contexte monodépartement et multiactivité

L'ensemble des travaux effectués dans le contexte multiactivité visent à construire des horaires remplis par des séquences admissibles d'activités. Dans ce contexte, les travaux rencontrés peuvent être divisés en trois ensembles : l'affectation des activités aux quarts (Jin, 2009; Lequy et collab., 2012a,b), la construction de quarts avec placement des activités (Demasse et collab., 2005; Côté et collab., 2011a; Quimper et Rousseau, 2010; Restrepo et collab., 2012) et la construction simultanée de quarts et de cycles avec placement des activités (Dahmen et Rekik, 2015).

Affectation des activités aux quarts

Bard et Wan (2006) considèrent un problème d'affectation des activités aux horaires de travail dans un centre de traitement et de distribution du Service postal des États-Unis. La composition et la taille de la main d'œuvre est connue. Ce niveau décisionnel a été traité précédemment par Bard et collab. (2003) pour déterminer un effectif permanent permettant de rencontrer les objectifs stratégiques et budgétaires du centre. La main d'œuvre est homogène. Chaque employé possède un horaire déterminé a priori (Bard et collab., 2003). Cet horaire est constitué des jours de travail et de repos et des périodes de travail et de pause. Le problème consiste donc à spécifier la composition des horaires en termes d'activités. La sous-couverture n'est pas permise alors que la surcouverture est permise occasionnant ainsi un temps d'inactivité. L'environnement d'opération est continu. L'objectif est de trouver l'affectation qui minimise le coût total des transitions entre les stations de travail WSG. Les auteurs démontrent que le problème est NP-difficile. Ils proposent une formulation en nombre entiers basée sur la définition d'un réseau de flot multiproduit adapté au problème. Un produit correspond à un employé. Les sommets du graphe correspondent aux combinaisons (WSG,période). Le coût des arcs correspond aux coûts de passage d'un WSG à un autre. Le problème obtenu correspond à trouver le flot de coût minimum pour chaque employé en couvrant les contraintes de demande. La résolution du problème fait intervenir trois approches heuristiques : une heuristique gloutonne, une heuristique de post-traitement combinée avec une décomposition temporelle et une heuristique basée sur la recherche Tabou. Les résultats expérimentaux montrent que la résolution exacte du problème en utilisant CPLEX est la meilleure option pour le cas des plus petites instances alors que la décomposition temporelle du problème s'avère très performante et permet d'obtenir des solutions de bonne qualité lorsque la taille des instances augmente. Pour les grandes instances, lorsque la résolution n'est plus possible par le reste des approches, la recherche Tabou devient la meilleure option.

Jin (2009) considère un problème d'affectation des activités et des tâches aux quarts de travail. Étant donné un ensemble de quarts personnalisés assignés à une main d'œuvre hétérogène, le problème consiste à affecter les tâches et les activités aux périodes travaillées tout en

tenant compte des qualifications des employés. Chaque tâche doit commencer à l'intérieur d'une fenêtre de temps spécifique, possède une durée constante et ne peut pas être interrompue. Il existe des relations de précédence entre les tâches. À l'opposé, une activité peut être interrompue, elle peut débiter à n'importe quelle période de l'horizon et possède une durée variable. L'auteur développe une heuristique de résolution à deux étapes. La première étape consiste à résoudre un problème approximatif d'affectation des tâches et des activités. Dans la deuxième étape, la position des tâches obtenue à la première étape est retenue et un problème d'affectation des activités est résolu. Jin (2009) propose trois formulations pour le problème résolu à la première étape. Ces formulations sont à objectifs multiples. Dans tous les cas, le premier objectif consiste à minimiser la sous-couverture des tâches et le deuxième objectif consiste à minimiser la sous-couverture des activités. D'autres objectifs sont traités séparément dans chaque formulation pour donner une grande flexibilité au problème de la deuxième étape : 1) la répartition de la surcouverture sur les périodes et les activités, 2) la maximisation des qualifications des employés inactifs et 3) la répartition des qualifications des employés inactifs entre les périodes.

Lequy et collab. (2012b) traitent un problème similaire à celui de Jin (2009). Les auteurs proposent une formulation en nombres entiers et une heuristique de résolution à deux étapes pour résoudre le problème. La première étape consiste à résoudre un problème approximatif en nombres entiers pour l'affectation des tâches tout en tenant compte des activités. La deuxième étape consiste à résoudre un problème d'affectation simultanée des activités et des tâches. Cette étape remet en question le placement des tâches déterminé à la première étape. Quatre stratégies sont proposées et diffèrent selon le degré de flexibilité portant sur la réoptimisation du placement des tâches. Des instances sont générées aléatoirement. Les tests expérimentaux montrent que la stratégie offrant le degré de flexibilité le plus élevé domine les autres moyennant des temps de calcul relativement plus élevés.

Lequy et collab. (2012a) traitent un problème similaire à celui de (Jin, 2009; Lequy et collab., 2012b) en absence des tâches. Il s'agit donc d'un problème d'affectation des activités aux quarts. Trois modèles en nombres entiers sont proposés. Pour tous ces modèles, la fonction objectif est une somme pondérée de trois termes : le coût total de sous-couverture, de surcouverture et des transitions. Les auteurs introduisent deux concepts : la pièce de travail (*piece of work*) et le bloc d'activité (*activity block*). La pièce de travail est un ensemble consécutif de périodes travaillées. Le bloc d'activité est une partie de la pièce de travail qui consiste à affecter une activité spécifique durant l'intervalle de temps couvert par ce bloc. La première formulation est un modèle de flot multiproduit (*multi-commodity network flow model*). Chaque produit correspond à une pièce de travail. La structure du graphe associé au modèle vérifie la propriété suivante : chaque chemin partant de l'origine et atteignant la destination correspond à une séquence d'activités admissible pour la pièce de travail. Deux types de variables sont associés aux arcs du graphe : les variables de bloc et les variables

de transition. Une variable de bloc est associée à chaque bloc d'activité admissible et une variable de transition correspond à une liaison entre deux blocs d'activités consécutifs. Le deuxième modèle basé sur les blocs (*block model*) est une reformulation du premier modèle en supprimant les variables de transition réduisant ainsi la taille du problème. La troisième formulation possède une structure qui se prête à la résolution par la génération de colonnes (*column generation model*). Cette formulation utilise des variables de quarts remplis par des séquences d'activités admissibles. Ces modèles sont résolus par des méthodes heuristiques : un Branch-and-Bound, un Branch-and-Bound modifié (truncated branch-and-bound) et une heuristique basée sur la génération de colonnes. Une heuristique d'horizon roulant est utilisée pour résoudre les instances de grande taille. Des solutions de très bonne qualité sont obtenues en une durée de deux heures pour des instances définies sur une semaine impliquant jusqu'à 100 quarts par jour et 15 activités. Ces résultats sont obtenus par l'heuristique basée sur la génération de colonnes combinée avec l'heuristique d'horizon roulant.

Construction de quarts multiactivités

Plusieurs approches basées sur la programmation par contraintes ont traité le problème de construction de quarts dans un contexte multiactivité. Demassey et collab. (2005) résolvent le problème intégré de construction de quarts et d'affectation des activités. Les auteurs forment le problème par un modèle de couverture généralisée. La relaxation linéaire du problème est résolue à l'aide d'une procédure de génération de colonnes. À chaque itération de la génération de colonnes, un problème de satisfaction de contraintes est utilisé pour générer les quarts admissibles remplis par des activités qui sont introduits dans le problème maître restreint. Dans le même contexte, Quimper et Rousseau (2010) utilisent les langages réguliers et les grammaires non contextuelles pour modéliser les règles définissant les séquences admissibles d'activités composant les quarts. Une heuristique de recherche dans un grand voisinage (*large neighbourhood search*) est proposée. Les opérateurs de voisinage sont basés sur l'utilisation d'un automate étendu et de grammaires non contextuelles.

Côté et collab. (2007) proposent un modèle en nombres entiers qui est basé sur les langages réguliers afin de formuler les règles d'admissibilité des séquences d'activités formant les quarts. Plus récemment, Côté et collab. (2011a) traitent un problème de construction de quarts personnalisés dans un contexte multiactivité. Les auteurs proposent un modèle en nombres entiers qui utilise les grammaires non contextuelles, introduites précédemment, afin de modéliser les réglementations. Ce modèle décrit implicitement, pour chaque employé, l'ensemble de ses quarts admissibles remplis par les activités. Récemment, Côté et collab. (2013) proposent un algorithme de Branch-And-Price pour résoudre le problème de construction de quarts personnalisés dans un contexte multiactivité. Le problème maître restreint est une formulation de type couverture généralisée. Les auteurs utilisent le modèle en nombres entiers basé sur les grammaires (*MIP grammar*) pour formuler les sous-problèmes

associés aux employés.

À notre connaissance, Côté et collab. (2011b) sont les premiers à proposer une approche implicite, très intéressante par rapport à l'approche explicite, pour la construction de quarts de travail anonymes dans un contexte multiactivité. L'approche implicite est utilisée afin de modéliser la flexibilité portant sur la composition des quarts en activités. Comme précédemment, les grammaires non contextuelles sont utilisées pour modéliser les contraintes d'admissibilité des quarts multiactivités. Les expérimentations montrent que leur formulation permet de résoudre des problèmes de grande taille (jusqu'à 10 activités) en des temps très courts.

Plus tard, Salvagnin et Walsh (2012) proposent une approche hybride qui utilise la programmation en nombres entiers et la programmation par contraintes. L'approche est basée sur le concept de la réduction orbitale (*orbital shrinking*) proposée par Fischetti et Liberti (2012). La plupart des réglementations sont modélisées par les langages réguliers sauf pour un ensemble spécifique de contraintes de cardinalité (*cardinality constraints*). Ceci est pour éviter la complexité provoquée par l'incorporation de telles contraintes. Quoique le modèle basé sur les grammaires de Côté et collab. (2011b) est de loin meilleur que le modèle de Salvagnin et Walsh (2012) pour l'ensemble des instances considérées, l'approche de décomposition proposée par les auteurs devient prometteuse lorsque les grammaires ne sont plus viables pour modéliser certaines règles s'exprimant par des contraintes de cardinalité.

Plus tard, Restrepo et collab. (2012) proposent une approche basée sur la génération de colonnes pour résoudre le problème de construction de quarts anonymes dans un contexte multiactivité. Le problème maître est un modèle explicite. Le sous-problème est un problème de plus court chemin avec contraintes de ressources. Les auteurs testent leur approche sur des instances réelles. Il s'agit d'un problème de construction de quarts pour des agents de sécurité et des caissiers dans un parking à la ville de Bogota. Leur approche a permis à la firme de comparer plusieurs scénarios pour améliorer leurs stratégies de pilotage des ressources humaines. Les auteurs concluent que lorsque la longueur de la période de discrétisation diminue, la taille du sous-problème limite la performance de l'approche de résolution.

Construction simultanée de cycles et de quarts multiactivités

Plus récemment, Dahmen et Rekik (2015) considèrent un problème de construction de quarts de travail personnalisés sur un horizon multi-jour et pour une main d'œuvre hétérogène. L'environnement d'opération est continu. Les auteurs supposent que les cycles de travail sont connus a priori. Nous classons ce travail parmi les problèmes de construction simultanée de cycles et de quarts étant donné que plusieurs réglementations lient entre les quarts, et qui doivent être incorporé habituellement dans ce type de problème, sont considérées par les auteurs. L'objectif est de construire les quarts personnalisés remplis par des activités et

de lier entre ces quarts pour obtenir des horaires définis sur un horizon multi-jour. Les employés possèdent des qualifications pour les activités. Le problème consiste à minimiser le coût total de sous-couverture et de surcouverture en respectant des contraintes horizontales de liaison entre quarts telles que la contrainte budgétaire et la charge de travail maximal par employé. Le problème est formulé en un modèle de couverture généralisée. Cette formulation explicite est résolue en utilisant une heuristique hybride combinant la recherche Tabou avec le Branch-and-Bound de CPLEX. Les tests expérimentaux effectués sur un ensemble large d'instances générées aléatoirement montrent que l'heuristique permet d'identifier des solutions proches de l'optimum en des temps très courts.

Eitzen et collab. (2004) traitent le problème de construction simultanée de cycles et de quarts personnalisés dans un contexte multiactivité pour une main d'œuvre hétérogène. Certaines formes de flexibilité sont ignorées dans ce travail. Les auteurs utilisent une formulation de couverture généralisée et proposent différentes stratégies de résolution. Une analyse comparative est effectuée. Ce travail est une application réelle à un problème de construction d'horaires dans une centrale d'énergie électrique en Australie.

Récemment, une série de travaux s'adressent au problème de construction simultanée de cycles et de quarts personnalisés dans un contexte multiactivité (Restrepo et collab., 2015a,b, 2016). Toutes les formes de flexibilité sont considérées. Restrepo et collab. (2016) développent une approche de Branch-and-Price pour résoudre le problème. Deux modèles sont proposées : un modèle basé sur les quarts et un modèle basé sur les horaires. Dans le premier modèle, les colonnes correspondent aux quarts. Les cycles admissibles remplis par les quarts sont modélisés implicitement dans le problème maître en utilisant des contraintes additionnelles. Les sous-problèmes utilisent les grammaires non contextuelles pour construire les quarts lors de la génération de colonnes. Dans le second modèle, les colonnes sont des horaires (*tours*). Les sous-problèmes sont résolus en utilisant une procédure à deux étapes. La première étape vise à déterminer la composition des quarts en termes d'activités. La deuxième étape lie entre les quarts afin d'obtenir les cycles en utilisant un problème de plus court chemin avec contraintes de ressources. Les instances les plus larges comportent cinq activités et l'horizon de planification est d'une semaine. Les auteurs montrent que la deuxième formulation, basée sur les horaires, implique un saut d'intégrité plus petit et permet d'obtenir des solutions de très bonne qualité avec un gap d'optimalité de 1%. Restrepo et collab. (2015a) proposent une approche combinant la décomposition de Benders et la génération de colonnes pour résoudre le même problème. Les résultats expérimentaux montrent que l'approche permet d'identifier des solutions de très bonne qualité pour des instances impliquant jusqu'à 10 activités.

Restrepo et collab. (2015b) traitent le problème de construction simultanée de cycles et de quarts anonymes. L'environnement d'opération est discontinu. La particularité de ce travail consiste à incorporer l'incertitude portant sur la demande. Les auteurs utilisent la pro-

grammation stochastique avec recours en deux étapes (*Two-stage stochastic programming with recourse*). Les décisions de la première étape correspondent à la construction des horaires (*tours*) alors que les décisions de la deuxième étape correspondent à l'affectation des activités et des pauses aux quarts de travail. Les résultats expérimentaux montrent que l'utilisation de l'approche stochastique permet d'éviter des coûts additionnels.

En conclusion, différentes approches de modélisation et de résolution ont été proposées pour le contexte monodépartement. Ces approches sont basées sur la programmation en nombres entiers, les modèles de flot, la programmation par contraintes, les langages réguliers, etc. Pour certains cas très rares, la résolution donne naissance à des algorithmes polynomiaux. Dans le reste des cas, des techniques élaborées sont évoquées comme la génération de colonnes, la génération de plans de coupes, le Branch-And-Bound, le Branch-And-Price, la décomposition de Benders, les heuristiques de recherche dans le voisinage, etc. La nouvelle tendance consiste à essayer de combiner certaines de ces techniques ensemble.

3.3 Construction d'horaires dans un contexte multidépartement

La construction d'horaires dans un contexte multidépartement consiste à déterminer les horaires des employés en permettant le transfert entre les départements. Combiner la construction d'horaires de plusieurs départements dans le même processus d'optimisation fait en sorte que la taille du problème devient très grande. Tous les travaux rencontrés dans la littérature écartent la résolution directe du problème et procèdent par la décomposition heuristique du problème. Dans ce qui suit, nous présentons d'une manière détaillée ces travaux.

3.3.1 Contexte multidépartement et monoactivité par département

L'ensemble des travaux rencontrés dans la littérature dans un contexte MDSA implique deux problèmes : la construction des cycles anonymes multidépartements et la construction simultanée des cycles et des quarts personnalisés multidépartements.

Construction des cycles anonymes multidépartements

À notre connaissance, [Mabert et Raedels \(1977\)](#) sont les premiers à s'adresser à un contexte multidépartement. Les auteurs traitent un problème de construction de cycles anonymes dans un contexte multidépartement. Les auteurs considèrent le cas réel d'une banque (Purdue National Bank) formée de huit agences. Chaque agence est monoactivité. Tous les caissiers peuvent travailler dans n'importe quelle agence. La main d'œuvre est à temps-partiel. Les cycles admissibles sont ceux obtenus en plaçant deux ou trois jours de travail à l'intérieur d'une semaine ouvrable de cinq jours. Aucune restriction sur les transferts n'est considérée. L'affectation des agences aux jours travaillés se fait sans aucune restriction. Le problème est formulé comme un modèle de couverture généralisée. L'objectif est de minimiser le nombre

total de cycles anonymes sélectionnés et le nombre total de jours travaillés. La demande est spécifiée par jour de travail et par agence bancaire.

Deux heuristiques de résolution sont proposées pour résoudre le problème difficile dont la taille augmente exponentiellement avec le nombre d'agences. Les deux heuristiques décomposent le problème en deux niveaux décisionnels : la construction des cycles anonymes et le placement des agences à l'intérieur des jours travaillés. La construction des cycles anonymes se fait en agrégeant la demande de toutes les agences ensemble. La différence entre les deux heuristiques est que la première résout un modèle de couverture généralisé alors que la deuxième utilise une heuristique constructive. Le placement des agences à l'intérieur des jours travaillés se fait en minimisant les transferts ou encore les déplacements. Nous utilisons dans ce cas le terme déplacement étant donné que le caissier n'est pas rattaché à une agence en particulier. La deuxième heuristique permet de diminuer les temps de calcul par rapport à la première. La stratégie de gestion de la main d'œuvre dans la banque utilisait exclusivement des caissiers à temps complet sans transferts entre les agences. L'emploi des caissiers à temps partiel qui peuvent être déplacés entre les agences a permis d'économiser environ 30000 par an à la banque.

Comme pour le contexte monodépartement, les travaux explicites ont été suivis par des travaux implicites afin de réduire la taille des modèles obtenus. Plus tard, *Bechtold (1988)* a repris le travail de *Mabert et Raedels (1977)* et a proposé une approche implicite pour la résolution du même problème. Les tests expérimentaux montrent que l'approche implicite domine l'approche explicite. Le modèle implicite, noté BIF, divise les employés en deux types : les employés SD et les employés MD. Les employés SD désignent les employés travaillant dans le même département tout au long de la fenêtre de temps. Les employés MD travaillent dans plus qu'un département durant la fenêtre. Alors que le nombre d'employés de type SD est spécifié par département en se basant sur une demande désagrégée de ce département, le nombre d'employés MD se base sur une demande agrégée pour tous les départements ensemble. Plusieurs formulations implicites sont proposées permettant de considérer des objectifs différents. Le modèle est résolu. L'affectation a posteriori de la main d'œuvre multidépartementale aux différents départements se fait d'une manière heuristique.

Construction simultanée des cycles et des quarts personnalisés multidépartements

Al-Yakoob et Sherali (2007) s'adressent à un problème de construction simultanée de quarts et de cycles personnalisés dans un contexte multidépartement. Un département est une station-service. Les auteurs définissent une approche de résolution à deux étapes. La première étape vise à partitionner l'ensemble des employés et des stations en des sous-ensembles mutuellement disjoints. Ce problème est formulé comme un modèle linéaire en nombres entiers. La fonction objectif maximise les préférences des employés pour les stations et les heures de travail. À la deuxième étape, un sous-problème est associé à chaque partition éta-

blie à l'étape précédente. L'ensemble des sous-problèmes sont résolus séparément. Un sous-problème correspond à la construction simultanée de cycles et de quarts personnalisés dans un contexte multidépartement. Le partitionnement permet donc de réduire la taille du problème global. Al-Yakoob et Sherali (2008) propose une heuristique basée sur la génération de colonnes pour résoudre le problème intégré.

3.3.2 Contexte multidépartement et multiactivité par département

Plus tard, Bard et Wan (2008) s'adressent à un problème de construction simultanée de cycles et de quarts personnalisés dans un contexte multidépartement. La contribution principale de ce travail couvre deux niveaux décisionnels habituellement négligés : l'affectation de chaque employé à son département d'origine et le respect de certaines restrictions de transfert entre les départements, notés ici WSG. Il existe certaines règles sur les transferts entre certains départements à cause de leur localisation et des objectifs de supervision. Un employé est rattaché à un WSG lorsqu'il passe l'intervalle de temps le plus élevé de son horaire de travail dans ce WSG. En absence de restrictions sur les transferts, le problème peut être traité en deux phases d'une manière optimale. Dans une première phase, la demande est agrégée pour tous les WSG du centre. La composition et la taille de la main d'œuvre est déterminée à l'aide d'un problème de construction simultanée de cycles et de quarts anonymes avec placement de pause. On se ramène au contexte monodépartement (Bard et collab., 2003). Dans une deuxième phase, un problème d'affectation des WSG aux périodes travaillées est résolu et un horaire rempli est construit pour chaque employé du centre (Bard et Wan, 2006).

Cette décomposition n'est plus optimale lorsqu'il y a des restrictions sur les transferts entre départements. En présence de restrictions sur les transferts, les auteurs proposent deux approches heuristiques : séquentielle et itérative. L'approche séquentielle repose sur la définition d'un graphe particulier dont les nœuds sont les WSG et les arcs sont les déplacements permis entre les WSG. Les auteurs commencent par partitionner l'ensemble des nœuds en des sous-ensembles mutuellement disjoints. Une partition est obtenue en regroupant tous les WSG vérifiant la propriété d'absence de restrictions sur les transferts. Les partitions sont par la suite triées selon un ordre spécifique. Finalement, les sous-problèmes associés aux partitions sont résolus selon l'ordre de priorité identifié précédemment. L'approche itérative fait intervenir deux étapes principales. La première étape consiste à relaxer les restrictions sur les transferts et à résoudre un problème global de construction simultanée de cycles et de quarts anonymes en agrégeant la demande de tous les WSG ensemble. La solution obtenue sera un intrant pour la deuxième étape qui utilise un problème en nombres entiers. Ce problème consiste à l'affectation simultanée des WSG aux périodes travaillées et des WSG d'origine aux horaires personnalisés en tenant compte des restrictions sur les transferts. Ce problème n'est pas forcément réalisable à cause de la relaxation des restrictions au niveau de la première étape. Autrement dit, la solution agrégée de la première étape n'est pas for-

cément suffisante pour couvrir la demande désagrégée de la deuxième étape. Afin d'assurer la réalisabilité du problème d'affectation simultanée des WSG et des WSG d'origine, la sous-couverture est acceptée et minimisée dans la fonction objectif. Lorsque la sous-couverture est non nulle, un nouveau problème est résolu en considérant deux options. La première option correspond à réitérer la procédure à deux étapes en considérant la demande incrémentée de la sous-couverture obtenue. Le processus itératif s'arrête lorsqu'une solution réalisable soit obtenue. La deuxième option correspond à égaliser la demande à la sous-couverture et à faire appel à l'approche séquentielle pour rajouter le nombre d'employés suffisant à couvrir les écarts. Les auteurs démontrent que le problème de la deuxième étape est NP-difficile. Ils proposent un algorithme de fixation de variables (*Variable fixing algorithm*) pour le résoudre.

Les auteurs concluent que l'approche itérative combinée avec l'algorithme de fixation de variables et l'utilisation a posteriori de l'approche séquentielle pour couvrir les écarts de demande est la meilleure alternative. Trois classes d'instances sont considérées et chaque classe contient quatre scénarios faisant intervenir le même nombre de WSG et différent par les interdictions de transfert entre les WSG. La première classe d'instances contient trois WSG, et une fois résolue, utilise un peu plus de 200 employés. La deuxième contient quatre WSG et un peu plus de 70 employés et la troisième contient cinq WSG et presque 50 employés. En moyenne, des solutions de bonne qualité sont obtenues dans un intervalle de 30 minutes.

Les approches de résolution des problèmes de construction d'horaires dans un contexte multidépartement ont presque toutes un schéma de décomposition à deux étapes. Nous distinguons deux cas. Le premier, celui adopté par Mabert et Raedels (1977); Bechtold (1988); Bard et Wan (2008) qui divisent le problème en deux étapes : une étape d'agrégation et une étape de désagrégation. La première étape se base sur l'agrégation des courbes de demande de tous les départements en une seule, ramenant ainsi le contexte multidépartement au contexte monodépartement. Ceci permet alors à la première étape d'obtenir des horaires agrégés pour l'ensemble des départements. La deuxième étape consiste à affecter les départements aux horaires préétablis. Des post-traitements sont utilisés. Le deuxième cas, celui de (Al-Yakoob et Sherali, 2007) traite le problème différemment en partitionnant l'ensemble des départements et des employés en des sous-problèmes mutuellement exclusifs et ceci en maximisant les préférences des employés. Ensuite, chacun des sous-problèmes est à son tour un problème de construction d'horaires dans un contexte multidépartement de taille plus petite.

Pour conclure, cette revue nous a permis d'examiner des axes prometteurs de recherche pour la thèse. D'une part, elle montre que le contexte multidépartement est peu exploré par rapport au contexte monodépartement qui a été dûment étudié dans la littérature. Ce constat a donc motivé l'élaboration de cette thèse. Tel que précisé par Van den Bergh et collab. (2013), les organisations ont besoin d'outils permettant un haut niveau d'intégration décisionnelle et l'incorporation d'aspects plus réalistes du problème. D'autre part, elle fait apparaître que la résolution du problème de construction d'horaires dans le contexte multidépartement est

généralement traitée par une heuristique à plusieurs étapes. Ceci a donc motivé notre travail pour la conception d'approches de résolution performantes supportant des organisations de taille très grande non encore rencontrées dans la littérature.

Chapitre 4

Modélisation du problème

4.1 Modèle de construction d'horaires

Le modèle de construction d'horaires est un modèle *explicite* basé sur la formulation de couverture généralisée proposée par Dantzig (1954). On commence par l'énumération exhaustive de tous les horaires admissibles H . À chaque horaire énuméré $h \in H$, on associe une variable de décision binaire X_h qui vaut 1 si l'horaire est sélectionné et 0 sinon. L'ensemble des horaires construits doivent, d'une part respecter les règles de travail et de transfert, et d'autre part couvrir les courbes de demande des départements. Les règles de travail et de transfert sont explicitement considérées lors de l'énumération des horaires admissibles.

Pour chaque département d et chaque période i , on définit une contrainte de demande qui impose que l'offre en effectifs soit supérieure ou égale à la demande. Étant donné que la taille et la composition de la main d'œuvre dans cette thèse sont un intrant fixe, il arrive que l'offre ne suffise pas à couvrir la demande d'une ou plusieurs périodes. À l'opposé, les employés peuvent être disponibles mais inactifs. On définit ainsi, pour chaque département d et chaque période i , une variable de sous-couverture $U_{d,i}$ et de surcouverture $O_{d,i}$ que l'on pénalisera dans la fonction objectif.

Étant donné que nous traitons le cas d'horaires personnalisés, l'ensemble H est formé par l'union des horaires admissibles personnalisés de tous les employés ($\cup_{e \in E} H_e$). Une *contrainte d'affectation* est considérée dans le modèle pour sélectionner un et un seul horaire pour chaque employé. Dans ce qui suit, on se propose de définir formellement l'ensemble des horaires admissibles personnalisés H_e de l'employé e .

Rappelons qu'un quart q est modélisé par le quintuplet $(s_q, l_b, l_f, d_b, d_f)$. Le quart q est composé de deux blocs : un bloc de début dans le département d_b qui commence à s_q et se termine à $s_q + l_b - 1$ et un bloc de fin dans le département d_f qui commence à $s_q + l_b$ et se termine à $s_q + l_b + l_f - 1$. De plus, la durée du quart q est notée l_q . Notons $l_{q,d}^{tr}$ le nombre de périodes transférées dans le département d'accueil $d \neq d_e$ selon le quart q . Ce nombre peut être cal-

culé comme suit : $l_{q,d}^{tr} = l_b \mu_{d,d_b} + l_f \mu_{d,d_f}$ avec $\mu_{d,d'}$ un paramètre qui vaut 1 si $d = d'$ et 0 sinon.

On peut diviser l'ensemble des règles de travail et de transfert en deux types : *ponctuelles* et *horizontales*. Les règles ponctuelles servent à définir les quarts admissibles d'un jour donné alors que les règles horizontales servent à définir les liens admissibles entre quarts sur les différents jours de l'horizon.

Les règles ponctuelles de travail sont celles qui définissent les heures de début et les durées de travail. Elles sont définies par les types de quart de l'organisation. Les règles ponctuelles de transfert sont celles qui définissent la composition des quarts en départements. Elles imposent qu'un quart soit formé d'au plus deux blocs, que la durée d'un bloc soit au moins supérieure à un seuil B_{min} et que dans le cas d'un quart à deux blocs, au moins l'un des deux soit interne. L'ensemble des règles ponctuelles de travail et de transfert permettent de définir l'ensemble des quarts admissibles personnalisés de l'employé e le jour j désigné par $Q_{e,j} = \cup_{t \in T} Q_{e,j,t}$ avec T l'ensemble des types de quart. Pour la définition de l'ensemble $Q_{e,j,t}$, on réfère à la Section 2.1.1.

Les règles horizontales de travail sont la règle de séparation minimale entre quarts reçus durant deux jours de travail consécutifs (S_{min}) et la règle de la charge horaire maximale (L_{max}). La règle horizontale de transfert est la règle du pourcentage minimal de travail dans le département d'origine (T_{min}). L'ensemble des règles horizontales de travail et de transfert permettent de définir l'ensemble des horaires admissibles $H_e = \{(q_j)_{j \in J_e} \in \prod_{j \in J_e} Q_{e,j} | s_{q_{j+1}} - (s_{q_j} + l_{q_j}) \geq S_{min}, \forall (j, j+1) \in J_e^2; \sum_{j \in J_e} l_{q_j} \leq L_{max}; \sum_{j \in J_e} l_{q_j} - \sum_{d \in D_e} \sum_{j \in J_e} l_{q_j,d}^{tr} \geq T_{min} \sum_{j \in J_e} l_{q_j}\}$.

La fonction objectif est une somme pondérée de quatre termes. En premier lieu, on minimise le coût total de sous-couverture. L'objectif ultime dans les organisations est de construire des calendriers de travail qui permettent d'atteindre la «sous-couverture zéro». Les sous-couvertures d'effectifs créent des goulots, dégradent la qualité de service et provoquent des pertes de marché. En deuxième lieu, on minimise le coût total de surcouverture. La surcouverture est un gaspillage de ressources car l'employé sera payé pour des heures d'inactivité. En troisième lieu, on minimise le coût total de transfert. Dans les organisations, le recours fréquent à la main d'œuvre multidépartementale provoque la variation de la productivité des employés entre les départements, diminue la motivation du personnel à cause du changement du superviseur et de l'équipe de travail, complique la gestion administrative et cause une gêne de travail (Bechtold, 1988). En quatrième lieu, on minimise le coût total de travail. Il s'agit de la masse salariale calculée proportionnellement aux heures travaillées. Nous définissons les coefficients γ^- , γ^+ , γ^{tr} et γ^l pour pondérer les objectifs correspondant au coût total de sous-couverture, de surcouverture, de transfert et de travail respectivement.

Le coût de l'horaire $h = (q_j)_{j \in J_e}$ de l'employé e est formé de deux termes : coûts de travail et

de transfert. Notons par $l_h = \sum_{j \in J_e} l_{q_j}$ le nombre de périodes travaillées et $l_{h,d}^{tr} = \sum_{j \in J_e} l_{q_j,d}^{tr}$ le nombre de périodes transférées dans le département d selon l'horaire h . Le coût de travail est égal à $c_h^l = c_e^l l_h$ avec c_e^l le taux de rémunération d'une période travaillée de l'employé e . Le coût de transfert est égal à $c_h^{tr} = \sum_{d \in D_e} c_{e,d}^{tr} l_{h,d}^{tr}$ avec $c_{e,d}^{tr}$ le coût d'une période transférée de l'employé e dans son département d'accueil d . Rappelons que c_d^- et c_d^+ désignent le coût d'une sous-couverture et d'une surcouverture dans le département d . Soit $\delta_{h,d,i}$ le paramètre de couverture qui vaut 1 si à la période i l'employé recevant l'horaire $h \in H_e$ sera dans le département d et 0 sinon.

Le modèle de construction d'horaires de travail sur un horizon multi-jour et dans un contexte multidépartement est comme suit :

Modèle (MCH)

$$\min \gamma^- \sum_{d \in D} \sum_{i \in I} c_d^- U_{d,i} + \gamma^+ \sum_{d \in D} \sum_{i \in I} c_d^+ O_{d,i} + \gamma^{tr} \sum_{e \in E} \sum_{h \in H_e} c_h^{tr} X_h + \gamma^l \sum_{e \in E} \sum_{h \in H_e} c_h^l X_h \quad (4.1)$$

$$\text{s.c.} \sum_{e \in E} \sum_{h \in H_e} \delta_{h,d,i} X_h + U_{d,i} - O_{d,i} = r_{d,i}, \quad \forall d \in D, i \in I \quad (4.2)$$

$$\sum_{h \in H_e} X_h = 1, \quad \forall e \in E \quad (4.3)$$

$$X_h \in \{0, 1\}, \forall e \in E, h \in H_e \quad (4.4)$$

$$U_{d,i}, O_{d,i} \geq 0, \forall d \in D, i \in I \quad (4.5)$$

La fonction objectif (4.1) minimise le coût total de sous-couverture, de surcouverture, de transfert et de travail. L'ensemble des contraintes est formé par les contraintes de demande (4.2), les contraintes d'affectation (4.3), les contraintes d'intégralité (4.4) et les contraintes de non négativité (4.5).

La formulation explicite de construction d'horaires MCH fait intervenir un nombre de variables de décision très élevé du moment qu'elle procède à l'énumération exhaustive de tous les horaires admissibles. Nous concluons qu'il serait pertinent de chercher à proposer des formulations permettant de réduire ce nombre.

4.2 Modèle de construction de quarts

Le modèle de construction de quarts est un modèle *hybride* qui énumère explicitement tous les quarts admissibles et modélise implicitement les liens admissibles entre les quarts formant un même horaire. On commence par l'énumération exhaustive de tous les quarts admissibles Q . À chaque quart énuméré $q \in Q$, on associe une variable de décision binaire X_q qui vaut 1 si le quart q est sélectionné et 0 sinon.

L'ensemble des quarts construits doivent, d'une part respecter les règles de travail et de

transfert, et d'autre part couvrir les courbes de demande des départements de l'organisation. Les règles de travail et de transfert ponctuelles sont explicitement considérées lors de l'énumération des quarts admissibles. Les règles de travail et de transfert horizontales sont modélisées par un ensemble de contraintes spécifiques : la contrainte de séparation minimale, la contrainte de travail maximal et la contrainte de pourcentage de non transfert minimal.

Étant donné que nous traitons le cas de quarts personnalisés, l'ensemble Q est formé par l'union des quarts personnalisés de tous les employés durant tous les jours travaillés ($\cup_{e \in E} \cup_{j \in J_e} Q_{e,j}$). Une *contrainte d'affectation* est considérée dans le modèle pour sélectionner un et un seul quart pour chaque employé e durant chaque jour travaillé $j \in J_e$.

Le coût du quart q de l'employé e est formé de deux termes : coûts de travail ($c_q^l = c_e^l l_q$) et de transfert ($c_q^{tr} = \sum_{d \in D_e} c_{e,d}^{tr} l_{q,d}^{tr}$). Soit $\delta_{q,d,i}$ le paramètre qui vaut 1 si à la période i l'employé recevant le quart q sera dans le département d et 0 sinon.

Le modèle de construction de quarts de travail sur un horizon multi-jour et dans un contexte multidépartement est comme suit :

Modèle (MCQ)

$$\min \gamma^- \sum_{d \in D} \sum_{i \in I} c_d^- U_{d,i} + \gamma^+ \sum_{d \in D} \sum_{i \in I} c_d^+ O_{d,i} + \gamma^{tr} \sum_{e \in E} \sum_{j \in J_e} \sum_{q \in Q_{e,j}} c_q^{tr} X_q + \gamma^l \sum_{e \in E} \sum_{j \in J_e} \sum_{q \in Q_{e,j}} c_q^l X_q \quad (4.6)$$

$$\text{s.c.} \sum_{e \in E} \sum_{j \in J_e} \sum_{q \in Q_{e,j}} \delta_{q,d,i} X_q + U_{d,i} - O_{d,i} = r_{d,i}, \quad \forall d \in D, i \in I \quad (4.7)$$

$$\sum_{q \in Q_{e,j}} X_q = 1, \quad \forall e \in E, j \in J_e \quad (4.8)$$

$$\sum_{q' \in Q_{e,j+1}} s_{q'} X_{q'} - \sum_{q \in Q_{e,j}} (s_q + l_q) X_q \geq S_{min}, \quad \forall e \in E, (j, j+1) \in J_e^2 \quad (4.9)$$

$$\sum_{j \in J_e} \sum_{q \in Q_{e,j}} l_q X_q \leq L_{max}, \quad \forall e \in E \quad (4.10)$$

$$\sum_{j \in J_e} \sum_{q \in Q_{e,j}} (l_q - \sum_{d \in D_e} l_{q,d}^{tr}) X_q \geq T_{min} \sum_{j \in J_e} \sum_{q \in Q_{e,j}} l_q X_q, \quad \forall e \in E \quad (4.11)$$

$$X_q \in \{0, 1\}, \quad \forall e \in E, j \in J_e, q \in Q_{e,j} \quad (4.12)$$

$$U_{d,i}, O_{d,i} \geq 0, \quad \forall d \in D, i \in I \quad (4.13)$$

La fonction objectif (4.6) minimise le coût total de sous-couverture, de surcouverture, de transfert et de travail. L'ensemble des contraintes est formé par les contraintes de demande (4.7), les contraintes d'affectation (4.8), les contraintes de travail et de transfert horizontales ((4.9) - (4.11)), les contraintes d'intégralité (4.12) et les contraintes de non négativité (4.13). Les

contraintes de travail et de transfert horizontales sont : la contrainte de séparation minimale (4.9), la contrainte de travail maximal (4.10) et la contrainte de non transfert minimal (4.11).

4.3 Modèle de construction de blocs

Le modèle de construction de blocs est un modèle *implicite* qui énumère explicitement tous les blocs admissibles, modélise implicitement les liens admissibles entre les blocs formant un même quart et modélise implicitement les liens admissibles entre les quarts formant un même horaire.

Le principe du modèle de construction de blocs est d'éviter l'énumération exhaustive des quarts en les définissant d'une manière implicite. La définition implicite des quarts repose sur un concept classique qui a été largement utilisé dans la littérature pour le contexte monodépartement et qui est relatif à l'utilisation des contraintes *forward* et *backward*.

Le modèle de construction de blocs proposé s'applique uniquement lorsque l'ensemble des durées admissibles d'un type de quart soit constitué d'un ensemble consécutif de multiples de la période de discrétisation.

Rekik et collab. (2004) ont démontré que la réalisabilité de certains problèmes de transport particuliers peut être garantie par un ensemble restreint de contraintes. La structure particulière que doit respecter le problème de transport se base sur deux propriétés : la «contiguïté» et l'«absence d'extra-chevauchement». Un problème de transport est défini par un graphe biparti $G(O, D, A)$ avec O l'ensemble des nœuds d'«offre», D l'ensemble des nœuds de «demande» et A l'ensemble des arcs du graphe. La propriété de contiguïté suppose l'existence d'une relation d'ordre total sur les nœuds de demande. Cette propriété est respectée lorsque l'ensemble des nœuds de demande lié à chaque nœud d'offre est consécutif selon la relation d'ordre définie. Ainsi, chaque nœud d'offre $o \in O$ est lié à un intervalle de nœuds de demande consécutifs noté $D(o) \subseteq D$. L'absence d'extrachevauchement implique qu'il n'existe aucune paire de nœuds d'offre (o_1, o_2) tel que l'ensemble des nœuds de demande auquel le nœud o_1 est relié soit strictement inclus dans l'ensemble auquel le nœud o_2 est relié ($D(o_1) \not\subseteq D(o_2)$).

4.3.1 Principe de base

Afin de décrire le modèle de construction de blocs de travail, on utilise les termes suivants :

- *Segment k* : désigne un intervalle de temps formé par un ensemble de périodes travaillées consécutivement dans un même département. Il est défini par son début s_k et sa durée l_k . De plus, on notera e_k sa fin. On distingue deux types de segments : segments de début et de fin. Ces segments servent à modéliser le déplacement de l'employé durant son quart d'un département à l'autre. Lorsqu'il s'agit d'un segment de

début, on utilisera la notation b pour dire qu'il s'agit de la partie du quart se déroulant avant le déplacement. De même, lorsqu'il s'agit d'un segment de fin, on utilisera la notation f pour dire qu'il s'agit de la partie du quart se déroulant après le déplacement.

- *Paire segment-département* ou bloc (k, d) : désigne un segment k travaillé dans le département d . On distingue deux types de blocs : blocs de début (associé à un segment de début) et de fin (associé à un segment de fin). Un bloc est dit interne lorsqu'il est exécuté dans le département d'origine. Il est externe lorsqu'il est exécuté dans l'un des départements d'accueil.
- Segment *fictif* k : désigne un segment k de durée nulle $l_k = 0$.
- Bloc fictif (k, d_0) : désigne un segment fictif accompli dans le département fictif d_0 .

L'idée de base est de modéliser le quart par le biais d'un bloc de début et d'un bloc de fin qui lui est admissible. La liaison entre blocs de début et de fin admissibles se fait implicitement à l'aide d'un ensemble de contraintes spécifiques. L'ajout du bloc fictif sert uniquement à modéliser les quarts monodépartements. Ainsi, un quart à deux blocs est obtenu en concaténant un bloc de début et un bloc de fin non fictif alors qu'un quart monodépartement est obtenu en concaténant un bloc de début et un bloc de fin fictif.

Les blocs de début et de fin admissibles selon les règles ponctuelles de travail et de transfert sont énumérés comme suit. Pour chaque employé e , pour chaque jour travaillé j et pour chaque type de quart t , on commence par énumérer explicitement les débuts et les durées admissibles. Ceci résulte en un ensemble de quarts « vides ». Du moment qu'on ne spécifie pas encore la séquence de départements remplissant un quart, il sera considéré comme vide. La composition des quarts vides est déterminée de la manière suivante. L'idée est de diviser chaque quart vide en deux segments distincts : un segment de début et un segment de fin. La division du quart peut se faire au début de n'importe quelle période à l'intérieur du quart à condition de respecter la restriction sur la durée minimale des blocs de travail. Les segments, ainsi obtenus, forment l'ensemble des segments admissibles de l'employé e le jour j relativement au type de quart t . Ensuite, chaque segment vide peut être rempli par un département admissible de l'employé donnant ainsi une paire segment-département ou encore un bloc. Les blocs obtenus forment l'ensemble des blocs personnalisés de l'employé e le jour j admissibles relativement au type de quart t . Un quart est entièrement défini par un bloc de début et un bloc de fin qui lui est admissible compte tenu des règles ponctuelles de travail et de transfert. On verra que l'incorporation d'un ensemble restreint de contraintes permet de lier chaque bloc de début à l'ensemble des blocs de fin qui lui correspondent pour enfin former un quart « rempli » admissible.

Cet ensemble de contraintes peut être divisé en deux types :

- les contraintes forward et backward, en plus d'un ensemble additionnel de contraintes d'égalité, pour lier entre les segments de début et de fin admissibles selon les règles ponctuelles de travail (heures de début et durées),

- et d'autres contraintes pour lier entre les blocs de début et de fin admissibles selon les règles ponctuelles de transfert (composition en départements).

4.3.2 Variables de décision

En plus des variables de sous-couverture et de surcouverture, on se propose de définir deux types de variables de décision : les variables de blocs de début X et les variables de blocs de fin Y . Un lien approprié entre ces types de variables est assuré par un ensemble restreint de contraintes qui garantira que des quarts admissibles peuvent être obtenus à partir des blocs sélectionnés.

Énumération des segments

Pour chaque employé $e \in E$, pour chaque jour travaillé $j \in J_e$ de cet employé et pour chaque type de quart $t \in T$ de l'organisation, nous définissons un problème de transport noté $G_{e,j,t} = (K_{e,j,t}^1, K_{e,j,t}^2, A_{e,j,t})$ avec $K_{e,j,t}^1$ l'ensemble des segments de début (l'offre), $K_{e,j,t}^2$ l'ensemble des segments de fin (la demande) et $A_{e,j,t}$ l'ensemble des liens admissibles entre segments. Afin de simplifier les notations, on va enlever le triplet d'indices (e, j, t) dans le reste des explications puis on va le réincorporer lors de l'écriture du modèle.

L'énumération des segments de début et de fin doit se faire en respectant les règles ponctuelles de l'organisation. Premièrement, chaque segment non fictif k doit respecter la restriction sur la durée minimale d'un bloc ($l_k \geq B_{min}$). Deuxièmement, on peut avoir au maximum deux blocs par jour travaillé pour un employé donné. Si le quart est multidépartement, un nœuds de début et de fin servent à le modéliser tel que le nœud de fin correspond à un segment non fictif. Si le quart est monodépartement, le nœud de fin servant à le modéliser correspond nécessairement à un segment de fin fictif. Troisièmement, les segments de début commencent à toutes les heures de début admissibles selon le type de quart t correspondant au graphe G , possèdent une durée comprise entre B_{min} et l_t^{max} et peuvent être complétés par au moins un segment de fin admissible. Les segments de fin commencent à toutes les heures de fin des segments de début admissibles énumérés, possèdent une durée permettant de ramener la durée cumulée (de début et de fin) à une durée admissible entre l_t^{min} et l_t^{max} et ne chevauchant pas des jours de repos.

Exemple A. Prenons un exemple pour illustrer la procédure d'énumération des segments. Pour simplifier la représentation, un segment k sera modélisé par la paire (s_k, e_k) avec s_k son heure de début et e_k son heure de fin. La période de discrétisation est égale à 1 heure et la durée minimale d'un bloc est $B_{min} = 2h$. Soit un type de quart t dont les débuts admissibles sont $S_t = \{8h00, 9h00\}$ et les durées admissibles sont $L_t = \{5h, 6h\}$ (c.-à-d. $l_{min}^t = 5h$ et $l_{max}^t = 6h$). Les segments de début sont ceux qui commencent à 8h00 ou 9h00 et durent entre 2h et 6h. Prenons le segment de début (8h00,10h00) comme exemple. Les segments de fin qui peuvent compléter ce segment de début sont ceux qui débutent à 10h et durent entre 3h et

4h, c'est à dire les segments (10h,13h) et (10h,14h). Le quart vide qui débute à 8h00 et dure 5h, par exemple, peut être obtenu en liant le segment de début (8h00,10h00) au segment de fin (10h,13h). L'énumération de tous les segments admissibles donne les résultats suivants.

- Les segments de début sont : (8h00, 10h00), (8h00, 11h00), (8h00, 12h00), (8h00, 13h00), (8h00, 14h00), (9h00, 11h00), (9h00, 12h00), (9h00, 13h00), (9h00, 14h00) et (9h00, 15h00).
- Les segments de fin sont : (10h00, 13h00), (10h00, 14h00), (11h00, 13h00), (11h00, 14h00), (11h00, 15h00), (12h00, 14h00), (12h00, 15h00), (13h00, 13h00), (13h00, 15h00), (14h00, 14h00) et (15h00, 15h00).

Nous verrons plus tard le graphe associé à cet exemple.

Énumération des blocs

Chaque segment de début $b \in K^1$ peut être exécuté au complet dans l'un des départements admissibles (d'origine ou d'accueil). L'ensemble des blocs de début est obtenu en énumérant toutes les possibilités d'exécuter chaque segment de début dans chaque département admissible. On note $\tilde{D}_e = D_e \cup \{d_e\}$ l'ensemble des départements admissibles. L'ensemble des blocs de début est donné par $K^1 \times \tilde{D}_e$.

Pour les segments de fin, on distingue entre les segments ayant une durée strictement positive et les segments ayant une durée nulle. Dans le premier cas, chaque segment de fin $f \in K^2$ vérifiant $l_f > 0$ peut être exécuté au complet dans l'un des départements admissibles (d'origine ou d'accueil). Dans le deuxième cas, pour chaque segment de fin $f \in K^2$ vérifiant $l_f = 0$, on peut énumérer un seul bloc fictif exécuté dans un département fictif d_0 . On note D_e^f l'ensemble des départements admissibles du segment de fin f . L'ensemble des blocs de fin est donné par $\{(f, d) | f \in K^2 \text{ et } d \in D_e^f\}$ avec :

$$D_e^f = \begin{cases} \tilde{D}_e & \text{si } l_f > 0 \\ \{d_0\} & \text{si } l_f = 0 \end{cases}$$

On pose $K^{2,0}$ l'ensemble des segments de fin ayant une durée nulle et $K^{2,1}$ l'ensemble des segments de fin ayant une durée strictement positive. Ainsi, $K^2 = K^{2,0} \cup K^{2,1}$.

Exemple A. Reprenons l'exemple précédent. Supposons de plus que l'employé associé au graphe considéré est rattaché au département d_1 et qu'il est qualifié pour travailler dans son département d'accueil d_2 . Pour simplifier la représentation, un bloc (k, d) sera modélisé par la triplet (s_k, e_k, d) avec s_k l'heure de début du segment k , e_k l'heure de fin du segment k et d le département dans lequel le bloc sera exécuté. L'énumération de tous les blocs admissibles donne les résultats suivants :

- Les blocs de début sont : (8h00, 10h00, d_1), (8h00, 10h00, d_2), (8h00, 11h00, d_1), (8h00, 11h00, d_2), (8h00, 12h00, d_1), (8h00, 12h00, d_2), (8h00, 13h00, d_1), (8h00, 13h00, d_2), (8h00, 14h00, d_1), (8h00, 14h00, d_2), (9h00, 11h00, d_1), (9h00, 11h00, d_2), (9h00, 12h00, d_1), (9h00,

12h00, d_2), (9h00, 13h00, d_1), (9h00, 13h00, d_2), (9h00, 14h00, d_1), (9h00, 14h00, d_2), (9h00, 15h00, d_1) et (9h00, 15h00, d_2).

- Les blocs de fin sont : (10h00, 13h00, d_1), (10h00, 13h00, d_2), (10h00, 14h00, d_1), (10h00, 14h00, d_2), (11h00, 13h00, d_1), (11h00, 13h00, d_2), (11h00, 14h00, d_1), (11h00, 14h00, d_2), (11h00, 15h00, d_1), (11h00, 15h00, d_2), (12h00, 14h00, d_1), (12h00, 14h00, d_2), (12h00, 15h00, d_1), (12h00, 15h00, d_2), (13h00, 13h00, d_0), (13h00, 15h00, d_1), (13h00, 15h00, d_2), (14h00, 14h00, d_0) et (15h00, 15h00, d_0).

Les variables de décision de blocs de début et de fin sont définies comme suit :

- $X_{b,d}$: Pour chaque bloc (paire segment-département) de début $(b, d) \in K^1 \times \tilde{D}_e$, on définit une variable de décision qui vaut 1 si ce bloc est sélectionné et 0 sinon.
- $Y_{f,d}$: Pour chaque bloc de fin $(f, d) \in (K^{2,0} \times \{d_0\}) \cup (K^{2,1} \times \tilde{D}_e)$, on définit une variable de décision qui vaut 1 si ce bloc est sélectionné et 0 sinon.

4.3.3 Contraintes de liaison entre blocs

Dans ce paragraphe, on montrera comment on peut lier entre les blocs de début et de fin en utilisant des problèmes de transport ayant chacun la structure particulière identifiée par [Rekik et collab. \(2004\)](#). La liaison entre les blocs de début et de fin se fait implicitement à l'aide de contraintes spécifiques permettant le respect des règles ponctuelles de travail et de transfert. Rappelons que le bloc est une paire segment-département. Il existe deux types de contraintes de liaison :

- les contraintes de «*liaison entre segments*» pour lier les blocs de début et de fin de point de vue des heures de début et des durées, et donc pour assurer les règles ponctuelles de travail.
- les contraintes de «*liaison entre départements*» pour lier les blocs de début et de fin de point de vue de la composition en départements, et donc pour assurer les règles ponctuelles de transfert.

Les règles ponctuelles de travail sont les heures de début et les durées admissibles des types de quart de l'organisation indépendamment des départements affectés. L'énumération des blocs selon un type de quart t permet d'obtenir un ensemble de blocs de début qui commencent à des heures de début admissibles relativement au type de quart t . Il reste à modéliser la règle sur la durée des quarts. Cette règle est respectée à l'aide d'un ensemble de contraintes de liaison entre segments. Les contraintes de liaison entre segments sont données par les contraintes de type forward et backward en plus d'un ensemble d'équations qu'on détaillera. Ces contraintes permettent la reconstitution des quarts admissibles en termes d'heures de début et de durées.

Les règles ponctuelles de transfert sont la durée minimale d'un bloc, un quart est formé par au plus deux blocs et s'il s'agit d'un quart à deux blocs, au moins l'un des deux doit

être interne. La restriction sur la durée minimale d'un bloc est vérifiée explicitement lors de l'énumération des blocs. La règle de deux blocs au maximum est aussi vérifiée explicitement par la considération d'un graphe biparti. Il reste donc à respecter la règle sur les couples de départements de début et de fin admissibles. Ceci est assuré par l'ajout de contraintes de «liaison entre départements». Elles sont données par un ensemble d'inégalités pour permettre la reconstitution de quarts admissibles en termes de composition en départements.

Dans le reste de cette section, on va s'intéresser aux contraintes de liaison. Celles-ci doivent permettre de reconstituer des quarts personnalisés admissibles une fois le problème résolu.

On considère le problème de transport défini pour chaque employé e , pour chaque jour travaillé j et pour chaque type de quart t . Ce problème est représenté par le graphe biparti $G = (K^1, K^2, A)$. Un nœud d'offre dans le problème de transport est défini pour chaque segment de début $b \in K^1$. L'offre de ce nœud est égale au nombre total de blocs de début assignés à l'employé e le jour j tel que le segment de début $b \in K^1$ soit exécuté ($\sum_{d \in \tilde{D}_e} X_{b,d}$). De même, un nœud de demande est défini pour chaque segment de fin $f \in K^2$. La demande de ce nœud est égale au nombre total de blocs de fin assignés à l'employé e le jour j tel que le segment de fin f soit exécuté ($\sum_{d \in D_e^f} Y_{f,d}$).

Contraintes ponctuelles de travail

Chaque type de quart t est défini par un ensemble de débuts admissibles S_t , une durée de travail minimale permise l_t^{min} et une durée de travail maximale permise l_t^{max} . Un arc $(b, f) \in A$ qui lie le segment de début b au segment de fin f est défini si et seulement si :

- l'heure de début du segment de fin f est égale à l'heure de début du segment de début b augmentée de sa durée ($s_f = s_b + l_b$),
- et la somme des durées des deux segments est admissible, c.-à-d. respecte les restrictions sur les durées de travail minimale et maximale permises ($l_t^{min} \leq l_b + l_f \leq l_t^{max}$).

De plus, on définit la relation d'ordre totale $<_2$ sur les segments de fin. Un segment de fin f est inférieur à un segment f' ($f <_2 f'$) s'il commence plus tôt ou si les deux segments possèdent la même heure de début et que la durée de f soit plus petite que celle de f' . De même, on définit une relation d'ordre totale $<_1$ sur l'ensemble des segments de début. Un segment de début b est inférieur à un segment b' ($b <_1 b'$) s'il finit plus tôt ou si les deux segments possèdent la même heure de fin et la durée de b est plus petite que celle de b' .

Proposition 1. *Le graphe de transport G respecte la structure identifiée par Rekik et collab. (2004) de contiguïté des intervalles et d'absence d'extrachevauchement.*

Démonstration. On commence par vérifier que la propriété de contiguïté est respectée. Soit b un segment de début. On pose $K^2(b)$ l'ensemble des segments de fin reliés à b dans le graphe

$G. K^2(b)$ est formé par des segments admissibles f ayant le même début ($s_f = s_b + l_b$) et dont la durée

- $l_f \in [\max(l_{min}^t - l_b, B_{min}), l_{max}^t - l_b]$ si $B_{min} \leq l_b < l_{min}^t$ (si le segment de début b n'est pas assez long pour former un quart alors la durée de f doit permettre de respecter les restrictions sur les durées minimale et maximale permises),
- $l_f \in \{0\} \cup [B_{min}, l_{max}^t - l_b]$ si $l_{min}^t \leq l_b \leq l_{max}^t - B_{min}$ (si 1) le segment de début b est assez long pour former un quart et 2) lorsqu'il est incrémenté du segment de fin le plus court ne dépasse pas la durée maximale permise du quart alors la durée de f peut être nulle ou comprise entre B_{min} et $l_{max}^t - l_b$),
- et $l_f = 0$ si $l_{max}^t - B_{min} < l_b$ (si 1) le segment de début b est assez long pour former un quart et 2) lorsqu'il est incrémenté du segment de fin le plus court dépasse la durée maximale permise du quart alors la durée de f ne peut être que nulle).

En utilisant la relation d'ordre sur les segments de fin $<_2$, la propriété de contiguïté des intervalles est satisfaite dans les trois cas. Il est à noter que le deuxième cas n'induit pas de discontinuité étant donné qu'aucun segment de fin ne peut être de durée strictement positive et strictement inférieure à B_{min} . Dorénavant, on désigne les ensembles $K^2(b)$ par des intervalles pour souligner qu'ils sont constitués d'éléments consécutifs triés selon un ordre croissant selon la relation d'ordre $<_2$.

On passe à vérifier que la propriété de l'absence d'extrachevauchement est respectée. Soient deux segments de début distincts b et b' qui sont liés respectivement aux intervalles de segments de fin $K^2(b)$ et $K^2(b')$. Selon la relation d'ordre sur les segments de début et sans perte de généralité, on suppose que le segment de début b est inférieur à b' . Afin de montrer qu'il est impossible d'avoir une inclusion stricte entre ces intervalles, on distingue deux cas. Le premier cas est obtenu lorsque les deux segments de début possèdent deux heures de fin différentes. Dans ce cas, les deux intervalles de segments de fin qui leur correspondent sont disjoints ($K^2(b) \cap K^2(b') = \emptyset$). Ceci découle directement du fait qu'un intervalle donné $K^2(b)$ est formé par l'ensemble des segments de fin qui commencent juste après l'heure de fin du segment de début b . Le deuxième cas correspond à la situation pour laquelle les deux segments de début b et b' se terminent à la même heure. Pour assurer le respect des restrictions relatives à la durée minimale et maximale d'un type de quart t , on doit vérifier les deux affirmations suivantes :

- le plus petit segment de fin lié au segment de début b' étant donné la relation d'ordre $<_2$ possède une durée inférieure ou égale à la durée du plus petit segment de fin lié au segment de début b , ainsi on peut affirmer que $\min K^2(b') <_2 \min K^2(b)$, et
- le plus grand segment de fin lié au segment de début b' possède une durée au maximum égale à la durée du plus grand segment de fin lié au segment de début b , ainsi on peut affirmer que $\max K^2(b') <_2 \max K^2(b)$.

On peut conclure alors qu'il est impossible que l'un des deux intervalles soit inclus strictement dans l'autre ($K^2(b) \not\subset K^2(b')$ et $K^2(b') \not\subset K^2(b)$). D'où, la propriété d'absence d'ex-trachevauchement est vérifiée. \square

Dans ce qui suit, on passe à expliquer l'écriture des contraintes forward et backward. On note $K^2(b) = [f_b^S, f_b^E]$ l'intervalle des segments de fin liés au segment de début b . On pose $f^S = \min(K^2)$ et $f^E = \max(K^2)$. On définit les ensembles suivants pour chaque employé e , pour chaque jour travaillé j et pour chaque type de quart t :

$$\begin{aligned} K^{2S} &= \cup_{b \in K^1} \{f_b^S\} \\ K^{2E} &= \cup_{b \in K^1} \{f_b^E\} \\ K^{2B}(f) &= \{f' \in K^2 \mid f' \geq f\} = [f, f^E], \forall f \in K^{2S} \\ K^{2F}(f) &= \{f' \in K^2 \mid f' \leq f\} = [f^S, f], \forall f \in K^{2E} \\ K^{1B}(f) &= \{b \in K^1 \mid K^2(b) \subseteq K^{2B}(f)\} = \{b \in K^1 \mid f \leq f_b^S\}, \forall f \in K^{2S} \\ K^{1F}(f) &= \{b \in K^1 \mid K^2(b) \subseteq K^{2F}(f)\} = \{b \in K^1 \mid f_b^E \leq f\}, \forall f \in K^{2E} \end{aligned}$$

Les contraintes forward et backward pour lier entre les segments admissibles (indépendamment des départements formant les blocs) sont données respectivement par les expressions suivantes :

$$\sum_{f' \in K^{2F}(f)} \left(\sum_{d \in D_e^{f'}} Y_{f',d} \right) - \sum_{b \in K^{1F}(f)} \left(\sum_{d \in \tilde{D}_e} X_{b,d} \right) \geq 0, \quad \forall f \in K^{2E} \setminus \{f^E\} \quad (4.14)$$

$$\sum_{f' \in K^{2B}(f)} \left(\sum_{d \in D_e^{f'}} Y_{f',d} \right) - \sum_{b \in K^{1B}(f)} \left(\sum_{d \in \tilde{D}_e} X_{b,d} \right) \geq 0, \quad \forall f \in K^{2S} \setminus \{f^S\} \quad (4.15)$$

La réalisabilité du problème de transport associé à un employé e durant un jour travaillé j relativement à un type de quart t est alors assurée par les contraintes forward (4.14) et backward (4.15), en plus de la contrainte d'égalité (4.16) dont l'objectif est d'égaliser entre le nombre total de blocs de début assignés à l'employé e le jour j relativement au type de quart t (l'offre totale) et le nombre total de blocs de fin assignés à l'employé e le jour j relativement au type de quart t (la demande totale).

$$\sum_{f \in K^2} \sum_{d \in D_e^f} Y_{f,d} - \sum_{b \in K^1} \sum_{d \in \tilde{D}_e} X_{b,d} = 0 \quad (4.16)$$

Exemple A. La Figure 4.1 illustre le graphe de transport correspondant à l'exemple A.

Les segments de début sont $b_1 = (8h00,10h00)$, $b_2 = (9h00,11h00)$, \dots , $b_{10} = (9h00,15h00)$. De même, les segments de fin sont $f_1 = (10h00,13h00)$, $f_2 = (10h00,14h00)$, \dots , $f_{11} = (15h00,15h00)$.

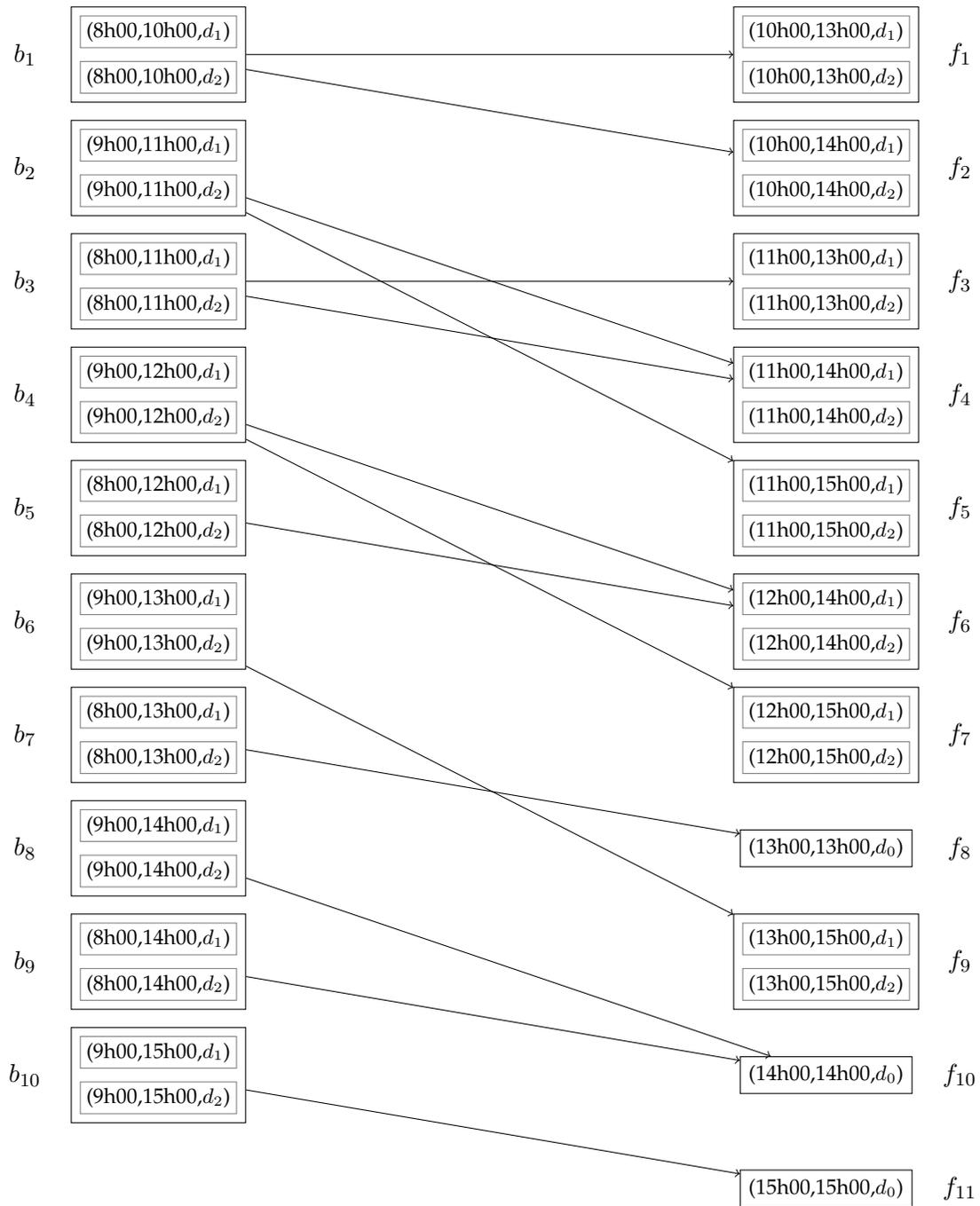


Figure 4.1 – Graphe de transport associé à l'exemple A

De plus, l'employé peut exécuter chaque segment de début dans son segment d'origine d_1 ou d'accueil d_2 . Par exemple, le segment b_1 peut être exécuté dans d_1 ou d_2 obtenant ainsi les deux blocs $(b_1, d_1) = (8h00, 10h00, d_1)$ et $(b_1, d_2) = (8h00, 10h00, d_2)$. Il en est de même pour tout segment de fin non fictif. Un segment fictif est associé à un seul bloc fictif. Par exemple, le segment fictif f_{11} est associé au bloc $(f_{11}, d_0) = (15h00, 15h00, d_0)$.

Prenons comme exemple le segment f_9 appartenant à l'ensemble K^{2S} . On a $K^{2B}(f_9) = \{f_9, f_{10}, f_{11}\}$ et $K^{1B}(f_9) = \{b_6, b_8, b_9, b_{10}\}$. La contrainte backward correspondante à f_9 s'écrit comme suit :

$$(Y_{(f_9, d_1)} + Y_{(f_9, d_2)}) + Y_{(f_{10}, d_0)} + Y_{(f_{11}, d_0)} - (X_{(b_6, d_1)} + X_{(b_6, d_2)}) - (X_{(b_8, d_1)} + X_{(b_8, d_2)}) - (X_{(b_9, d_1)} + X_{(b_9, d_2)}) - (X_{(b_{10}, d_1)} + X_{(b_{10}, d_2)}) \geq 0$$

Prenons comme exemple les deux segments f_2 et f_4 appartenant à l'ensemble K^{2E} . Pour le segment f_2 , on a $K^{2F}(f_2) = \{f_1, f_2\}$ et $K^{1F}(f_2) = \{b_1\}$. Pour le segment f_4 , on a $K^{2F}(f_4) = \{f_1, f_2, f_3, f_4\}$ et $K^{1F}(f_4) = \{b_1, b_3\}$. Les contraintes forward associées aux segments f_2 et f_4 sont données respectivement par les inégalités suivantes :

$$(Y_{(f_1, d_1)} + Y_{(f_1, d_2)}) + (Y_{(f_2, d_1)} + Y_{(f_2, d_2)}) - (X_{(b_1, d_1)} + X_{(b_1, d_2)}) \geq 0 \quad (4.17)$$

$$(Y_{(f_1, d_1)} + Y_{(f_1, d_2)}) + (Y_{(f_2, d_1)} + Y_{(f_2, d_2)}) + (Y_{(f_3, d_1)} + Y_{(f_3, d_2)}) + (Y_{(f_4, d_1)} + Y_{(f_4, d_2)}) - (X_{(b_1, d_1)} + X_{(b_1, d_2)}) - (X_{(b_3, d_1)} + X_{(b_3, d_2)}) \geq 0 \quad (4.18)$$

Plus récemment, [Rekik et collab. \(2010\)](#) ont proposé une reformulation des contraintes forward et backward qui réduit considérablement la densité de la matrice de contraintes sans augmenter le nombre total de contraintes. Les auteurs ont démontré que la nouvelle formulation résulte en une amélioration considérable de la taille du modèle et des temps de résolution. On se propose d'appliquer la même technique au modèle de construction de blocs de travail. Dans ce qui suit, on expliquera la technique introduite par [Rekik et collab. \(2010\)](#).

Soient M et R les cardinalités respectives des deux ensembles K^{2E} et K^{2S} . On commence par trier les éléments de l'ensemble K^{2E} selon un ordre croissant de la relation d'ordre $<_2$. On obtient ainsi $K^{2E} = \{f_1^E, f_2^E, \dots, f_M^E\}$ avec $f_1^E < f_2^E < \dots < f_M^E$. De plus, on trie les éléments de l'ensemble K^{2S} selon un ordre décroissant de la relation d'ordre $<_2$. Ainsi, on obtient $K^{2S} = \{f_1^S, \dots, f_{R-1}^S, f_R^S\}$ avec $f_R^S < f_{R-1}^S < \dots < f_1^S$. Soient f_m^E et f_{m+1}^E deux éléments consécutifs de l'ensemble K^{2E} . Soient de plus Z_m^E et Z_{m+1}^E les variables artificielles (variables d'écart) qui correspondent respectivement aux contraintes forward associées aux éléments f_m^E et f_{m+1}^E . On peut ainsi remplacer les inégalités forward au niveau des nœuds f_m^E et f_{m+1}^E par les égalités suivantes :

$$\sum_{f' \in K^{2F}(f_m^E)} \sum_{d \in D_e^{f'}} Y_{f',d} - \sum_{b \in K^{1F}(f_m^E)} \sum_{d \in \bar{D}_e} X_{b,d} = Z_m^E \quad (4.19)$$

$$\sum_{f' \in K^{2F}(f_{m+1}^E)} \sum_{d \in D_e^{f'}} Y_{f',d} - \sum_{b \in K^{1F}(f_{m+1}^E)} \sum_{d \in \bar{D}_e} X_{b,d} = Z_{m+1}^E \quad (4.20)$$

avec $Z_m^E \geq 0$ et $Z_{m+1}^E \geq 0$. Or, étant donné la structure particulière du problème de transport identifiée par Rekik et collab. (2004), on a $K^{2F}(f_m^E) \subseteq K^{2F}(f_{m+1}^E)$ et $K^{1F}(f_m^E) \subseteq K^{1F}(f_{m+1}^E)$. On peut remplacer alors la contrainte (4.20) par la nouvelle contrainte forward suivante :

$$\sum_{f' \in K^{2F}(f_{m+1}^E) \setminus K^{2F}(f_m^E)} \sum_{d \in D_e^{f'}} Y_{f',d} - \sum_{b \in K^{1F}(f_{m+1}^E) \setminus K^{1F}(f_m^E)} \sum_{d \in \bar{D}_e} X_{b,d} + Z_m^E = Z_{m+1}^E \quad (4.21)$$

Le même raisonnement peut être appliqué aux contraintes backward pour l'obtention de nouvelles contraintes backward moins denses. L'ensemble des nouvelles contraintes forward et backward comporte beaucoup moins de coefficients non nuls, ce qui explique la réduction significative de la densité de la matrice de contraintes.

Afin d'écrire les nouvelles contraintes, on définit un ensemble de variables d'écart correspondant aux contraintes forward notées $(Z_m^E)_{m \in \{1,2,\dots,M-1\}}$ et un ensemble de variables d'écart correspondantes aux contraintes backward notées $(Z_r^S)_{r \in \{1,2,\dots,R-1\}}$. De plus, afin de simplifier l'écriture du modèle, on définit trois variables Z_0^E, Z_0^S et Z_M^E tel que $Z_0^E = Z_0^S = Z_M^E = 0$. On considère également deux segments additionnels $f_0^E = -\infty$ et $f_0^S = +\infty$ tel que $K^{1F}(f_0^E) = K^{2F}(f_0^E) = K^{1B}(f_0^S) = K^{2B}(f_0^S) = \emptyset$.

Soient les ensembles suivants :

$$\begin{aligned} K_m^{2F} &= K^{2F}(f_{m+1}^E) \setminus K^{2F}(f_m^E) =]f_m^E, f_{m+1}^E], \forall m \in \{0, 1, \dots, M-1\} \\ K_m^{1F} &= K^{1F}(f_{m+1}^E) \setminus K^{1F}(f_m^E) = \{b \in K^1 | f_b^E \in]f_m^E, f_{m+1}^E]\}, \forall m \in \{0, 1, \dots, M-1\} \\ K_r^{2B} &= K^{2B}(f_{r+1}^S) \setminus K^{2B}(f_r^S) = [f_{r+1}^S, f_r^S[, \forall r \in \{0, 1, \dots, R-1\} \\ K_r^{1B} &= K^{1B}(f_{r+1}^S) \setminus K^{1B}(f_r^S) = \{b \in K^1 | f_b^S \in [f_{r+1}^S, f_r^S[], \forall r \in \{0, 1, \dots, R-1\} \end{aligned}$$

Pour chaque graphe G , on obtient de nouvelles contraintes forward (4.22) (l'équation d'initialisation avec $m = 0$ et la relation récursive avec $m \in \{1, 2, \dots, M-2\}$), de nouvelles contraintes backward (4.23) (l'équation d'initialisation $r = 0$ et la relation récursive $r \in \{1, 2, \dots, R-2\}$) et une nouvelle contrainte d'égalité (l'équation (4.22) avec $m = M-1$).

$$Z_{m+1}^E = Z_m^E + \sum_{f' \in K_m^{2F}} \sum_{d \in D_e^{f'}} Y_{f',d} - \sum_{b \in K_m^{1F}} \sum_{d \in \tilde{D}_e} X_{b,d}, \forall m \in \{0, 1, \dots, M-1\} \quad (4.22)$$

$$Z_{r+1}^S = Z_r^S + \sum_{f' \in K_r^{2B}} \sum_{d \in D_e^{f'}} Y_{f',d} - \sum_{b \in K_r^{1B}} \sum_{d \in \tilde{D}_e} X_{b,d}, \forall r \in \{0, 1, \dots, R-2\} \quad (4.23)$$

Exemple A. La reformulation des contraintes forward (4.17) et (4.18) donne les égalités suivantes :

$$Z_1^E = (Y_{(f_1,d_1)} + Y_{(f_1,d_2)}) + (Y_{(f_2,d_1)} + Y_{(f_2,d_2)}) - (X_{(b_1,d_1)} + X_{(b_1,d_2)}) \quad (4.24)$$

$$Z_2^E = Z_1^E + (Y_{(f_3,d_1)} + Y_{(f_3,d_2)}) + (Y_{(f_4,d_1)} + Y_{(f_4,d_2)}) - (X_{(b_3,d_1)} + X_{(b_3,d_2)}) \quad (4.25)$$

Dans le reste, on va réincorporer le triplet d'indices (e, j, t) dans les notations.

Contraintes ponctuelles de transfert

Les contraintes de liaison entre départements servent à lier les blocs en s'appuyant sur leurs compositions en termes de départements. Ces contraintes doivent assurer que lorsqu'un employé reçoit durant un jour de travail un quart exécuté dans deux départements différents, il doit nécessairement travailler dans son département d'origine durant l'un de ces blocs. Autrement dit, si un employé reçoit un bloc de début externe dans un département d'accueil d , aucun bloc de fin externe dans un département d'accueil $d' \neq d$ ne pourra être sélectionné.

$$\sum_{t \in T} \left(\sum_{b \in K_{e,j,t}^1} X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} \sum_{d' \in D_e \setminus \{d\}} Y_{f,d'} \right) \leq 1, \forall e \in E, j \in J_e, d \in D_e \quad (4.26)$$

Notons qu'un quart monodépartement de durée l_q qui est exécuté dans le département d peut être modélisé, non seulement par un bloc de début de durée l_q dans d suivi d'un bloc de fin fictif dans d_0 , mais aussi par toutes les possibilités de blocs de début b et de fin f admissibles exécutés dans le département d et de durée cumulée $l_b + l_f = l_q$. Pour éviter de représenter le même quart monodépartement de deux manières, on se propose d'ajouter des contraintes qui interdisent qu'un employé reçoive un bloc de début et de fin durant un jour donné dans le même département. On considère ainsi la contrainte qui impose qu'aucun bloc de fin dans d ne pourra être sélectionné si un bloc de début dans d est sélectionné.

$$\sum_{t \in T} \left(\sum_{b \in K_{e,j,t}^1} X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} Y_{f,d} \right) \leq 1, \forall e \in E, j \in J_e, d \in D_e \cup \{d_e\} \quad (4.27)$$

Dans ce qui suit, pour chaque département externe $d \in D_e$, on va fusionner la contrainte d'admissibilité des séquences de départements (4.26) et la contrainte de non redondance

(4.27). On obtient ainsi la contrainte $\sum_{t \in T} (\sum_{b \in K_{e,j,t}^1} X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} \sum_{d' \in D_e} Y_{f,d'}) \leq 1, \forall e \in E, j \in J_e, d \in D_e$. De plus, on va garder la contrainte de non redondance (4.27) pour le département d'origine d_e .

4.3.4 Formulation du modèle

Le coût d'un bloc interne (k, d_e) est égal au coût de travail $c_{(k,d_e)}^l = c_e^l l_k$. Le coût d'un bloc externe (k, d) comporte deux termes : le coût de travail $c_{(k,d)}^l = c_e^l l_k$ et le coût de transfert $c_{(k,d)}^{tr} = c_{e,d}^{tr} l_k$. Le coût d'un bloc fictif est nul. On pose $\delta_{k,d,i}$ le paramètre qui vaut 1 si à la période i l'employé recevant le segment k est affecté au département d , et 0 sinon.

Modèle (MCB1)

$$\begin{aligned} \min & \gamma^- \sum_{d \in D} \sum_{i \in I} c_d^- U_{d,i} + \gamma^+ \sum_{d \in D} \sum_{i \in I} c_d^+ O_{d,i} \\ & + \gamma^{tr} \left(\sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \sum_{d \in D_e} \left(\sum_{b \in K_{e,j,t}^1} c_{(b,d)}^{tr} X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} c_{(f,d)}^{tr} Y_{f,d} \right) \right) \\ & + \gamma^l \left(\sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \sum_{d \in \tilde{D}_e} \left(\sum_{b \in K_{e,j,t}^1} c_{(b,d)}^l X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} c_{(f,d)}^l Y_{f,d} \right) \right) \end{aligned} \quad (4.28)$$

s.c.

$$\sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \left(\sum_{b \in K_{e,j,t}^1} \delta_{b,d,i} X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} \delta_{f,d,i} Y_{f,d} \right) + U_{d,i} - O_{d,i} = r_{d,i}, \forall d \in D, i \in I \quad (4.29)$$

$$\begin{aligned} Z_{e,j,t,m+1}^E &= Z_{e,j,t,m}^E + \sum_{f \in K_{e,j,t,m}^{2F}} \sum_{d \in D_e^f} Y_{f,d} - \sum_{b \in K_{e,j,t,m}^{1F}} \sum_{d \in \tilde{D}_e} X_{b,d}, \\ & \forall e \in E, j \in J_e, t \in T, m \in \{0, 1, \dots, M_{e,j,t} - 1\} \end{aligned} \quad (4.30)$$

$$\begin{aligned} Z_{e,j,t,r+1}^S &= Z_{e,j,t,r}^S + \sum_{f \in K_{e,j,t,r}^{2B}} \sum_{d \in D_e^f} Y_{f,d} - \sum_{b \in K_{e,j,t,r}^{1B}} \sum_{d \in \tilde{D}_e} X_{b,d}, \\ & \forall e \in E, j \in J_e, t \in T, r \in \{0, 1, \dots, R_{e,j,t} - 2\} \end{aligned} \quad (4.31)$$

$$\sum_{t \in T} \left(\sum_{b \in K_{e,j,t}^1} X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} \sum_{d' \in D_e} Y_{f,d'} \right) \leq 1, \forall e \in E, j \in J_e, d \in D_e \quad (4.32)$$

$$\sum_{t \in T} \left(\sum_{b \in K_{e,j,t}^1} X_{b,d_e} + \sum_{f \in K_{e,j,t}^{2,1}} Y_{f,d_e} \right) \leq 1, \forall e \in E, j \in J_e \quad (4.33)$$

$$\sum_{t \in T} \sum_{b \in K_{e,j,t}^1} \sum_{d \in \tilde{D}_e} X_{b,d} = 1, \forall e \in E, j \in J_e \quad (4.34)$$

$$\sum_{t \in T} \left(\sum_{b \in K_{e,j+1,t}^1} \sum_{d \in \tilde{D}_e} s_b X_{b,d} - \sum_{f \in K_{e,j,t}^2} \sum_{d \in D_e^f} (s_f + l_f) Y_{f,d} \right) \geq S_{min}, \forall e \in E, (j, j+1) \in J_e^2 \quad (4.35)$$

$$\sum_{j \in J_e} \sum_{t \in T} \left(\sum_{b \in K_{e,j,t}^1} \sum_{d \in \tilde{D}_e} l_b X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} \sum_{d \in \tilde{D}_e} l_f Y_{f,d} \right) \leq L_{max}, \forall e \in E \quad (4.36)$$

$$\sum_{j \in J_e} \sum_{t \in T} \left(\sum_{b \in K_{e,j,t}^1} l_b X_{b,d_e} + \sum_{f \in K_{e,j,t}^{2,1}} l_f Y_{f,d_e} \right) \geq T_{min} \quad (4.37)$$

$$\sum_{j \in J_e} \sum_{t \in T} \sum_{d \in \tilde{D}_e} \left(\sum_{b \in K_{e,j,t}^1} l_b X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} l_f Y_{f,d} \right), \forall e \in E$$

$$Z_{e,j,t,0}^E = 0, Z_{e,j,t,0}^S = 0, Z_{e,j,t,M}^E = 0, \forall e \in E, j \in J_e, t \in T \quad (4.38)$$

$$Z_{e,j,t,m}^E \geq 0, \forall e \in E, j \in J_e, t \in T, m \in \{1, 2, \dots, M_{e,j,t} - 1\} \quad (4.39)$$

$$Z_{e,j,t,r}^S \geq 0, \forall e \in E, j \in J_e, t \in T, r \in \{1, 2, \dots, R_{e,j,t} - 1\} \quad (4.40)$$

$$X_{b,d} \in \{0, 1\}, \forall e \in E, j \in J_e, t \in T, (b, d) \in K_{e,j,t}^1 \times \tilde{D}_e \quad (4.41)$$

$$Y_{f,d} \in \{0, 1\}, \forall e \in E, j \in J_e, t \in T, (f, d) \in (K_{e,j,t}^{2,0} \times \{d_0\}) \cup (K_{e,j,t}^{2,1} \times \{\tilde{D}_e\}) \quad (4.42)$$

$$U_{d,i}, O_{d,i} \geq 0, \forall d \in D, i \in I \quad (4.43)$$

La fonction objectif (4.28) consiste à minimiser le coût total de sous-couverture et de surcouverture en plus du coût total de transfert et de travail des blocs sélectionnés. Les contraintes de demande sont données par les équations (4.29). Les équations (4.30), (4.31) et (4.38) sont les contraintes forward, backward et d'égalité associées aux règles ponctuelles de travail. Les inégalités (4.32) et (4.33) représentent les contraintes ponctuelles de transfert. Les contraintes d'affectation (4.34) imposent le respect du cycle de travail de chaque employé. De plus, on introduit le même ensemble de contraintes considéré précédemment pour le modèle MCQ pour la liaison implicite entre les quarts d'un même horaire. Ces contraintes sont : les contraintes de séparation minimale entre quarts travaillés durant deux jours consécutifs pour chaque employé (4.35), les contraintes de travail maximal par employé (4.36) et les contraintes de pourcentage minimal de non transfert par employé (4.37).

On trouve finalement les contraintes d'intégralité des variables de blocs (4.41) et (4.42) et les contraintes de non négativité des variables de sous-couverture et de surcouverture (4.43) et des variables d'écart correspondantes aux contraintes forward (4.39) et backward (4.40).

4.4 Variante du modèle de construction de blocs

Une variante du modèle de construction de blocs est de supprimer l'ensemble des contraintes (4.32) et (4.33) et de modéliser la liaison entre départements dans la définition des contraintes forward et backward. Dans ce qui suit, on présente uniquement les changements qui doivent être appliqués. Les contraintes (4.30) et (4.31) sont remplacées par deux types de contraintes : les contraintes qui lient les blocs de début internes aux blocs de fin externes et fictifs et les contraintes qui lient les blocs de début externes aux blocs de fin internes et fictifs.

Les contraintes de liaison entre blocs de début internes et blocs de fin externes et fictifs sont :

$$V_{e,j,t,m+1}^E = V_{e,j,t,m}^E + \sum_{f' \in K_{e,j,t,m}^{2F}} \sum_{d \in D_e^{f'} \setminus \{d_e\}} Y_{f',d} - \sum_{b \in K_{e,j,t,m}^{1F}} X_{b,d_e}, \quad (4.44)$$

$$\forall e \in E, j \in J_e, t \in T, m \in \{0, 1, \dots, M_{e,j,t} - 1\}$$

$$V_{e,j,t,r+1}^S = V_{e,j,t,r}^S + \sum_{f' \in K_{e,j,t,r}^{2B}} \sum_{d \in D_e^{f'} \setminus \{d_e\}} Y_{f',d} - \sum_{b \in K_{e,j,t,r}^{1B}} X_{b,d_e}, \quad (4.45)$$

$$\forall e \in E, j \in J_e, t \in T, r \in \{0, 1, \dots, R_{e,j,t} - 2\}$$

avec V^E (respectivement V^S) des variables d'écart associées aux contraintes forward (respectivement backward) pour la modélisation des quarts internes et des quarts externes débutant au département d'origine.

De même, les contraintes de liaison entre les blocs de début externes et les blocs de fin internes et fictifs sont données comme suit :

$$W_{e,j,t,m+1}^E = W_{e,j,t,m}^E + \sum_{f' \in K_{e,j,t,m}^{2F}} \sum_{d \in D_e^{f'} \setminus D_e} Y_{f',d} - \sum_{b \in K_{e,j,t,m}^{1F}} \sum_{d \in D_e} X_{b,d}, \quad (4.46)$$

$$\forall e \in E, j \in J_e, t \in T, m \in \{0, 1, \dots, M_{e,j,t} - 1\}$$

$$W_{e,j,t,r+1}^S = W_{e,j,t,r}^S + \sum_{f' \in K_{e,j,t,r}^{2B}} \sum_{d \in D_e^{f'} \setminus D_e} Y_{f',d} - \sum_{b \in K_{e,j,t,r}^{1B}} \sum_{d \in D_e} X_{b,d}, \quad (4.47)$$

$$\forall e \in E, j \in J_e, t \in T, r \in \{0, 1, \dots, R_{e,j,t} - 2\}$$

avec W^E (respectivement W^S) des variables d'écart associées aux contraintes forward (respectivement backward) pour la modélisation des quarts totalement externes et des quarts externes finissant au département d'origine.

4.5 Modèle réduit de construction de blocs

Le modèle réduit de construction de blocs est obtenu en modifiant légèrement la définition du graphe de transport. L'idée est de modéliser le transfert entre départements en plaçant un bloc de fin à l'intérieur d'un quart monodépartement. La liaison entre quarts monodépartements et blocs de fin admissibles se fait implicitement à l'aide d'un ensemble de contraintes similaire à celui utilisé précédemment. Tel que décrit ultérieurement, ce modèle est très proche du modèle (MCB1) sauf que l'ensemble des blocs de début sera limité uniquement aux blocs ayant une durée assez longue pour obtenir un quart admissible lorsqu'ils sont liés à des blocs de fin fictifs.

Ce modèle permet ainsi de diminuer le nombre de variables de décision par rapport au modèle de blocs (MCB1). Ceci dit, il est appliqué uniquement lorsqu'on restreint l'ensemble des quarts externes admissibles. Cette hypothèse consiste à supposer que l'on sache à priori vers

quel département d'accueil l'employé pourrait être transféré durant chaque jour de travail. En d'autres termes, pour chaque employé e possédant un ensemble de départements d'accueil D_e , supposons que l'employé peut être transféré durant son jour travaillé j uniquement vers un département d'accueil spécifique $d_{e,j} \in D_e$ connu à l'avance. Ainsi, l'ensemble des quarts considérés par le modèle (MCB2) est formé par tous les quarts internes durant ce jour et les quarts externes transférant l'employé vers le département $d_{e,j}$ durant une partie ou la totalité du jour. On verra dans les chapitres qui suivent que cette hypothèse, quoiqu'elle semble restrictive, sera exploitée lors de la résolution du problème lorsqu'on a une idée sur les transferts potentiels déduits d'une étape précédente.

Principe de base

Les quarts monodépartements et les blocs de fin admissibles sont énumérés comme suit. Pour chaque employé e , pour chaque jour travaillé j et pour chaque type de quart t , on commence par énumérer explicitement les débuts et les durées admissibles pour obtenir l'ensemble des quarts vides. L'idée est de placer à l'intérieur de chaque quart vide un segment de fin qui se termine à la même heure du quart. On énumère toutes les possibilités de placer un segment de fin à l'intérieur du quart en respectant que les heures de fin soient les mêmes et que la restriction sur la durée minimale des blocs de travail (non fictifs) soit satisfaite. On obtient ainsi l'ensemble des quarts vides et des segments de fin admissibles de l'employé e le jour j relativement au type de quart t . Ensuite, chaque quart (respectivement segment) vide peut être rempli par un département admissible de l'employé donnant ainsi un quart rempli monodépartement (respectivement une paire segment-département ou bloc). Les quarts monodépartements et blocs obtenus forment l'ensemble des quarts monodépartements et blocs de fin personnalisés de l'employé e le jour j admissibles relativement au type de quart t . Un quart monodépartement est entièrement défini en plaçant le bloc de fin fictif ayant la même heure de fin à l'intérieur de ce quart. Un quart multidépartement est entièrement défini par un quart monodépartement et un bloc de fin non fictif qui lui est admissible. Cette admissibilité est imposée selon les règles ponctuelles de travail et de transfert.

Soit un quart q modélisé par le quintuplet $(s_q, l_b, l_f, d_b, d_f)$. Ce quart commence à s_q , dure $l_q = l_b + l_f$ et se termine à $e_q = s_q + l_q - 1$. Dans le modèle, q sera obtenu à l'aide du quart qui se termine à e_q , dure l_q et se déroule au département d_b et à l'aide du segment de fin qui se termine à e_q , dure l_f et se déroule au département d_f .

Formulation

La nouvelle formulation du modèle de construction de blocs est obtenue en changeant légèrement la définition du graphe de transport $G_{e,j,t}$ en un nouveau graphe $\bar{G}_{e,j,t}$. Ceci se fait en réduisant l'ensemble des segments de début à l'ensemble des quarts vides. La définition des liens admissibles entre quarts vides et segments de fin ainsi que la relation d'ordre sur

l'ensemble des segments de fin seront définies en se basant sur les heures de fin.

Le problème de transport $G_{e,j,t} = (K_{e,j,t}^1, K_{e,j,t}^2, A_{e,j,t})$ est modifié au niveau de la définition des nœuds de l'offre. Le nouveau graphe $\bar{G}_{e,j,t} = (\bar{K}_{e,j,t}^1, \bar{K}_{e,j,t}^2, \bar{A}_{e,j,t})$ vérifie $\bar{K}_{e,j,t}^1 = K_{e,j,t}$ avec $K_{e,j,t}$ l'ensemble des quarts vides ou encore l'ensemble des segments de début de durée entre l_t^{min} et l_t^{max} et $\bar{K}_{e,j,t}^2 = K_{e,j,t}^2$ l'ensemble des segments de fin. Un nœud de début est énuméré dans le graphe $\bar{G}_{e,j,t}$ pour chaque quart vide b qui :

- commence à une heure admissible du type de quart t le jour travaillé j ($s_b \in S_{t,j}$),
- dure entre l_t^{min} et l_t^{max} ($l_t^{min} \leq l_b \leq l_t^{max}$),
- et qu'il ne chevauche pas un jour de repos de l'employé selon sa séquence ($s_b + l_b - 1 < s_{j+1}$ si $j + 1 \in J \setminus J_e$).

Un nœud de fin est énuméré dans le graphe $\bar{G}_{e,j,t}$ pour chaque segment de fin f vérifiant qu'il existe un quart vide b tel que :

- l'heure de fin de f coïncide avec l'heure de fin de b ($l_f + s_f = l_b + s_b$),
- et de durée appartenant à l'ensemble $\{0\} \cup [B_{min}, l_b - B_{min}]$.

Un arc $(b, f) \in \bar{A}_{e,j,t}$ existe si et seulement si b et f finissent en même temps et $l_f \in \{0\} \cup [B_{min}, l_b - B_{min}]$.

On modifie la relation d'ordre $<_2$ définie précédemment sur les segments de fin comme suit. Un segment de fin f est inférieur à un segment de fin f' s'il finit plus tôt ou si les deux segments possèdent la même heure de fin et que la durée de f_1 soit plus petite que celle de f' . On garde la même définition de la relation d'ordre $<_1$ sur l'ensemble des quarts vides.

Proposition 2. *Le graphe de transport modifié $\bar{G}_{e,j,t}$ vérifie la structure particulière de contiguïté des intervalles et d'absence d'extrachevauchement.*

Chaque segment de début (ou encore quart vide) $b \in K_{e,j,t}$ peut être exécuté au complet dans le département d'origine ou le département d'accueil correspondant à ce jour. L'ensemble des blocs de début est obtenu en énumérant toutes les possibilités d'exécuter chaque segment de début dans chacun de ces deux départements. Dans ce cas, $\tilde{D}_{e,j} = \{d_e, d_{e,j}\}$. L'ensemble des blocs de début est donné par $K_{e,j,t} \times \tilde{D}_{e,j}$.

Chaque segment de fin $f \in K_{e,j,t}^2$ vérifiant $l_f > 0$ peut être exécuté au complet dans l'un des deux départements d'origine ou d'accueil. Dans le deuxième cas, pour chaque segment de fin $f \in K_{e,j,t}^2$ vérifiant $l_f = 0$, on peut énumérer un seul bloc fictif exécuté dans un département fictif d_0 . L'ensemble des blocs de fin est donné par $\{(f, d) | f \in K_{e,j,t}^2 \text{ et } d \in D_{e,j}^f\}$ avec :

$$D_{e,j}^f = \begin{cases} \{d_e, d_{e,j}\} & \text{si } l_f > 0 \\ \{d_0\} & \text{si } l_f = 0 \end{cases}$$

Les variables de décision sont les variables de sous-couverture et de surcouverture, les variables de quarts monodépartements X , les variables de blocs de fin Y et les variables correspondant aux contraintes forward et backward \bar{Z} des nouveaux graphes $\bar{G}_{e,j,t}$.

Dans la fonction objectif, le coût total de travail est déduit directement à partir des variables de quarts monodépartements sélectionnés ($= \sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \sum_{b \in K_{e,j,t}} \sum_{d \in \bar{D}_{e,j}} c_e^l l_b X_{b,d}$). Le coût de transfert est plus compliqué à déterminer. Il est formé de deux termes selon que les quarts débutent ou se terminent dans un département externe :

- $\sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} c_{e,d_{e,j}}^{tr} (\sum_{b \in K_{e,j,t}} l_b X_{b,d_{e,j}} - \sum_{f \in K_{e,j,t}^{2,1}} l_f Y_{f,d_e})$ pour les quarts débutant au département externe $d_{e,j}$ et se terminant au département interne d_e ,
- $\sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \sum_{f \in K_{e,j,t}^{2,1}} c_{e,d_{e,j}}^{tr} l_f Y_{f,d_{e,j}}$ pour le cas contraire.

Le nouveau modèle s'écrit comme suit :

Modèle (MCB2)

$$\begin{aligned} \min & \gamma^- \sum_{d \in D} \sum_{i \in I} c_d^- U_{d,i} + \gamma^+ \sum_{d \in D} \sum_{i \in I} c_d^+ O_{d,i} \\ & + \gamma^{tr} \left(\sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} c_{e,d_{e,j}}^{tr} \left(\sum_{b \in K_{e,j,t}} l_b X_{b,d_{e,j}} - \sum_{f \in K_{e,j,t}^{2,1}} l_f Y_{f,d_e} + \sum_{f \in K_{e,j,t}^{2,1}} l_f Y_{f,d_{e,j}} \right) \right) \\ & + \gamma^l \sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \sum_{b \in K_{e,j,t}} \sum_{d \in \bar{D}_{e,j}} c_e^l l_b X_{b,d} \end{aligned} \quad (4.48)$$

$$\begin{aligned} \text{s.c.} & \sum_{e \in E} \sum_{j \in J_e} \sum_{t \in T} \left(\sum_{b \in K_{e,j,t}} \delta_{b,d,i} X_{b,d} - \sum_{f \in K_{e,j,t}^{2,1}} \delta_{f,d',i} Y_{f,d'} + \sum_{f \in K_{e,j,t}^{2,1}} \delta_{f,d,i} Y_{f,d} \right) \\ & + U_{d,i} - O_{d,i} = r_{d,i}, \forall d \in D, i \in I \end{aligned} \quad (4.49)$$

$$\begin{aligned} \bar{Z}_{e,j,t,m+1}^E &= \bar{Z}_{e,j,t,m}^E + \sum_{f \in \bar{K}_{e,j,t,m}^{2F}} \sum_{d \in D_{e,j}^f} Y_{f,d} - \sum_{b \in \bar{K}_{e,j,t,m}^{1F}} \sum_{d \in \bar{D}_{e,j}} X_{b,d}, \\ & \forall e \in E, j \in J_e, t \in T, m \in \{0, 1, \dots, \bar{M}_{e,j,t} - 1\} \end{aligned} \quad (4.50)$$

$$\begin{aligned} \bar{Z}_{e,j,t,r+1}^S &= \bar{Z}_{e,j,t,r}^S + \sum_{f \in \bar{K}_{e,j,t,r}^{2B}} \sum_{d \in D_{e,j}^f} Y_{f,d} - \sum_{b \in \bar{K}_{e,j,t,r}^{1B}} \sum_{d \in \bar{D}_{e,j}} X_{b,d}, \\ & \forall e \in E, j \in J_e, t \in T, r \in \{0, 1, \dots, \bar{R}_{e,j,t} - 2\} \end{aligned} \quad (4.51)$$

$$\sum_{t \in T} \left(\sum_{b \in K_{e,j,t}} X_{b,d} + \sum_{f \in K_{e,j,t}^{2,1}} Y_{f,d} \right) \leq 1, \forall e \in E, j \in J_e, d \in \{d_e, d_{e,j}\} \quad (4.52)$$

$$\sum_{t \in T} \sum_{b \in K_{e,j,t}} \sum_{d \in \bar{D}_{e,j}} X_{b,d} = 1, \forall e \in E, j \in J_e \quad (4.53)$$

$$\sum_{t \in T} \left(\sum_{b \in K_{e,j+1,t}} \sum_{d \in \bar{D}_{e,j}} s_b X_{b,d} - \sum_{b \in K_{e,j,t}} \sum_{d \in \bar{D}_{e,j}} (s_b + l_b) X_{b,d} \right) \geq S_{min}, \forall e \in E, (j, j+1) \in J_e^2 \quad (4.54)$$

$$\sum_{j \in J_e} \sum_{t \in T} \sum_{b \in K_{e,j,t}} \sum_{d \in \bar{D}_{e,j}} l_b X_{b,d} \leq L_{max}, \forall e \in E \quad (4.55)$$

$$\begin{aligned} \sum_{j \in J_e} \sum_{t \in T} \left(\sum_{b \in K_{e,j,t}} l_b X_{b,d_e} + \sum_{f \in K_{e,j,t}^{2,1}} l_f Y_{f,d_e} - \sum_{f \in K_{e,j,t}^{2,1}} l_f Y_{f,d_{e,j}} \right) \\ \geq T_{min} \sum_{j \in J_e} \sum_{t \in T} \sum_{b \in K_{e,j,t}} \sum_{d \in \tilde{D}_{e,j}} l_b X_{b,d}, \forall e \in E \end{aligned} \quad (4.56)$$

$$\bar{Z}_{e,j,t,0}^E = 0, \bar{Z}_{e,j,t,0}^S = 0, \bar{Z}_{e,j,t,M}^E = 0, \forall e \in E, j \in J_e, t \in T \quad (4.57)$$

$$\bar{Z}_{e,j,t,m}^E \geq 0, \forall e \in E, j \in J_e, t \in T, m \in \{1, 2, \dots, \bar{M}_{e,j,t} - 1\} \quad (4.58)$$

$$\bar{Z}_{e,j,t,r}^S \geq 0, \forall e \in E, j \in J_e, t \in T, r \in \{1, 2, \dots, \bar{R}_{e,j,t} - 1\} \quad (4.59)$$

$$X_{b,d} \in \{0, 1\}, \forall e \in E, j \in J_e, (b, d) \in K_{e,j,t} \times \tilde{D}_{e,j} \quad (4.60)$$

$$Y_{f,d} \in \{0, 1\}, \forall e \in E, j \in J_e, (f, d) \in (K_{e,j,t}^{2,0} \times \{d_0\}) \cup (K_{e,j,t}^{2,1} \times \tilde{D}_{e,j}) \quad (4.61)$$

$$U_{d,i}, O_{d,i} \geq 0, \forall d \in D, i \in I \quad (4.62)$$

4.6 Analyse comparative des formulations proposées

L'objectif de cette section est de choisir les formulations les plus adéquates pour modéliser les problèmes qui seront résolus dans les prochains chapitres.

4.6.1 Jeux de données

Le jeu de données JEU1 décrit à la Section 2.3 (voir Figure 2.4) est trop gros. Considérons deux jeux de données JEU2 et JEU3 qui sont plus petits et qui seront utilisés dans les approches de décomposition. Ces deux nouveaux jeux de données sont générés à partir du JEU1 (voir Figure 4.2) :

- JEU1 : Rappelons que pour le JEU1, la période de discrétisation est de 15 minutes. Un seul type de quart qu'on notera T1 est considéré dont les heures de début sont celles entre 0h00 et 4h00, 8h00 et 12h00 et finalement 16h00 et 20h00 et dont les durées admissibles sont celles entre 7h et 9h.
- JEU2 : Il s'agit d'instances conçues à partir du JEU1. La période de discrétisation est de 2 heures. Un seul type de quart qu'on notera T2 est considéré dont les heures de début sont celles entre 0h00 et 4h00, 8h00 et 12h00 et finalement 16h00 et 20h00 et dont les durées admissibles sont celles entre 6h et 10h. Chaque instance du JEU2 est obtenue à partir d'une instance du JEU1 comme suit : on considère exactement la même organisation (même ensemble de départements et d'employés avec les mêmes qualifications internes et externes), on remplace le type de quart T1 par T2 et on construit des courbes de demande de deux heures. Il est à noter que le reste des règles de travail et de transfert demeurent les mêmes à l'exception de la durée minimale d'un bloc qui devient égale à 2 heures au lieu d'une heure.
- JEU3 : Il s'agit d'instances conçues à partir du JEU1 par une simple restriction de l'ensemble des quarts admissibles. Pour les instances du JEU3, la période de discrétisation

est de 15 minutes. Un seul type de quart qu'on notera T3 est considéré dont les heures de début sont celles entre 1h00 et 3h00, 9h00 et 11h00 et finalement 17h00 et 19h00 et dont les durées admissibles sont celles entre 7h et demie et 8h et demie. Chaque instance du JEU3 est obtenue à partir d'une instance du JEU1 comme suit : on considère exactement la même organisation, on remplace le type de quart T1 par T3 et on garde les mêmes courbes de demande originales. Toutes les règles de travail et de transfert sont les mêmes. La seule différence est que l'on affecte à chaque employé un seul département d'accueil parmi ceux pour lesquels il est réellement qualifié. Autrement dit, la résolution d'une instance du JEU3 fournit une borne supérieure sur la valeur de l'objectif de l'instance qui lui correspond appartenant au JEU1.

Jeu de données

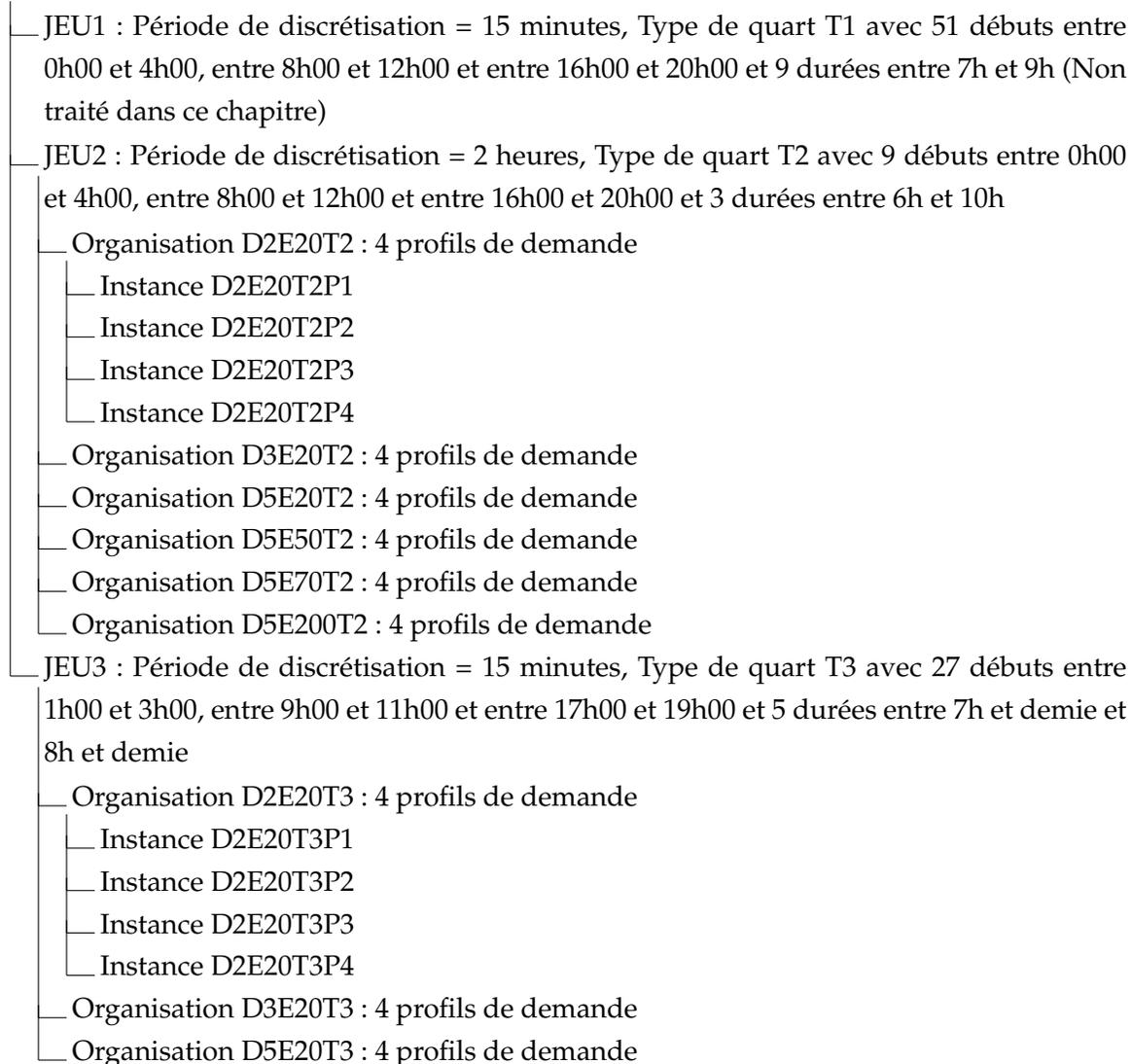


Figure 4.2 – Arborescence des jeux de données

4.6.2 Plan d'expérimentation

Le plan d'expérimentation peut être divisé en deux parties indépendantes : une partie dédiée au JEU2 et une autre dédiée au JEU3. Pour le JEU2, nous comparons entre le modèle de quarts MCQ et le modèle de blocs MCB1. En effet, pour cet ensemble d'instances, le modèle réduit de blocs MCB2 ne peut pas être appliqué étant donné qu'on ignore le département d'accueil vers lequel un employé peut être redirigé pendant un jour de travail. Pour le JEU3, cette hypothèse est vérifiée et nous allons comparer entre le modèle de quarts MCQ et le modèle réduit de blocs MCB2.

Notons que dans toutes les formulations, nous considérons des variables de quarts pour les employés qui ne possèdent aucun département d'accueil. Les formulations diffèrent ainsi uniquement de point de vue des employés qui peuvent être transférés. Pour chacun des modèles implicites MCB1 et MCB2, nous considérons les quatre versions suivantes :

1. avec blocs fictifs et avec contraintes de liaison entre départements,
2. sans blocs fictifs et avec contraintes de liaison entre départements,
3. avec blocs fictifs et sans contraintes de liaison entre départements,
4. sans blocs fictifs et sans contraintes de liaison entre départements.

D'une part, on peut modéliser les quarts monodépartements en utilisant des blocs de début (de même durée que les quarts à modéliser) liés à des blocs de fin fictifs (de durée nulle) comme il est présenté précédemment dans les Sections (4.3) et (4.5). Il s'agit dans ce cas des versions «avec blocs fictifs». Une autre variante peut être obtenue lorsqu'on définit des variables de quarts pour modéliser les quarts monodépartements. Dans ce cas, les graphes de transport sont utilisés uniquement pour modéliser les quarts multidépartements et ne contiennent pas des blocs de fin fictifs. Ainsi, on obtient des modèles hybrides qui combinent l'utilisation des variables de quarts et de blocs ensemble pour modéliser les horaires d'un employé donné. Il s'agit dans ce cas des versions «sans blocs fictifs».

D'autre part, on peut modéliser la liaison entre les nœuds de début et de fin des graphes de transport de deux manières différentes. La première manière consiste à lier les nœuds de début et de fin en se basant uniquement sur les règles de travail ponctuelles (heures de début et de fin). Ensuite, on ajoute des contraintes de liaison entre départements afin d'interdire la sélection de deux blocs travaillés à l'externe dans deux départements différents et d'éliminer aussi la redondance dans la définition des quarts monodépartements. Il s'agit dans ce cas des versions «avec contraintes de liaison entre départements» correspondant aux Sections (4.3) et (4.5). La deuxième manière consiste à lier directement entre les nœuds de début et de fin en se basant simultanément sur les règles de travail et de transfert ponctuelles. Ceci se fait en considérant deux graphes distincts. Le premier graphe sert à modéliser les quarts débutant à l'interne et le deuxième graphe sert à modéliser les quarts débutant à l'externe. Il s'agit dans

ce cas des versions «*sans contraintes de liaison entre départements*» correspondant à la section (4.4).

Ainsi, les neuf formulations étudiées sont les suivantes :

1. modèle de construction de quarts MCQ,
2. 4 versions du modèle de construction de blocs MCB11, MCB12, MCB13 et MCB14,
3. et 4 versions du modèle réduit de construction de blocs MCB21, MCB22, MCB23 et MCB24.

L'objectif de ce chapitre est de questionner les performances de ces formulations en résolvant les instances considérées directement à l'aide d'un solveur de programmation linéaire CPLEX 12.6. Dans toute la thèse, nous utilisons une machine ayant les spécifications suivantes :

- Processeur : Intel(R) Xeon(R) CPU X5650 @ 2.67 GHz 2.66 GHz (2 processeurs),
- Mémoire installée (RAM) : 72.0 GB,
- Type de système : Système d'exploitation 64-bit.

Les pondérations des quatre termes de la fonction objectif sont choisies comme suit : $\gamma^- = \gamma^+ = 47\%$, $\gamma^{tr} = 5\%$ et $\gamma^l = 1\%$.

Dans cette partie, on se limite à un ensemble de problèmes qui pourraient être résolus en utilisant la méthode de séparation et d'évaluation progressive «*Branch and Bound*» de CPLEX. La tolérance du gap d'optimalité «*Relative MIP gap tolerance*» est fixée à 1% et le temps limite de résolution est fixé à 1 heure.

Le plan d'expérimentation est décrit comme suit. On commence par résoudre les instances du JEU2 en utilisant les cinq formulations MCQ, MCB11, MCB12, MCB13 et MCB14. On identifie la meilleure formulation implicite parmi les quatre testées (MCB1*) puis on la compare à son tour à la formulation basée sur les quarts MCQ. On identifie ainsi la formulation à retenir pour résoudre les problèmes similaires à ceux traités dans le cadre du JEU2. On passe par la suite à résoudre les instances du JEU3 en utilisant les six formulations MCQ, MCB21, MCB22, MCB23, MCB24 et MCB1*. Comme précédemment, on identifie la meilleure formulation implicite parmi les quatre versions testées du modèle réduit de blocs (MCB2*) puis on la compare à son tour aux formulations MCB1* et MCQ. On identifie ainsi la formulation à retenir pour résoudre les problèmes similaires à ceux traités dans le cadre du JEU3.

4.6.3 Tests expérimentaux du JEU2

Commençons par comparer les quatre versions de la formulation implicite basée sur les blocs MCB1. La Figure 4.3 montre le nombre de variables, de contraintes et de coefficients non nuls lorsque les formulations MCB11, MCB12, MCB13 et MCB14 sont utilisées pour modéliser le problème correspondant aux instances issues des cinq premières classes du

JEU2 (D2E20T2, D3E20T2, D5E20T2, D5E50T2 et D5E70T2). Nous nous limitons à ces classes car CPLEX atteint la limite de temps et renvoie une solution réalisable avec un $\text{gap} > 1\%$ à partir des instances de la classe D5E70T2 pour toutes les formulations. Notons que la taille des modèles est donnée par classe d'instances étant donné que chaque classe contient quatre instances qui ne diffèrent que par les profils de demande.

Rappelons que les formulations MCB11 et MCB13 utilisent toutes les deux des variables de blocs fictifs et diffèrent uniquement du point de vue de l'utilisation des contraintes de liaison entre départements. De même, les formulations MCB12 et MCB14 n'utilisent pas des variables de blocs fictifs et diffèrent uniquement du point de vue de l'utilisation des contraintes de liaison entre départements. L'utilisation des contraintes de liaison entre départements permet d'obtenir des modèles avec moins de variables de décision, moins de contraintes mais plus de coefficients non nuls. En effet, l'utilisation de deux graphes de transport au lieu d'un seul double le nombre des contraintes forward et backward ainsi que le nombre de variables d'écart correspondant à ces contraintes. Par contre, les contraintes de liaison entre départements disparaissent. Ces contraintes devraient occasionner donc un nombre de coefficients non nuls légèrement élevé ce qui explique la diminution du nombre de coefficients non nuls lorsqu'elles sont supprimées du modèle.

Nous analysons maintenant l'effet de l'utilisation des blocs fictifs sur la taille du modèle implicite. Rappelons que les formulations MCB11 et MCB12 utilisent toutes les deux des contraintes de liaison entre départements et diffèrent uniquement du point de vue de l'utilisation des blocs fictifs. De même pour les formulations MCB13 et MCB14. L'utilisation des blocs fictifs augmente le nombre de contraintes mais diminue le nombre de coefficients non nuls. Le nombre de variables de décision quant à lui reste sensiblement le même. En effet, l'utilisation des blocs fictifs résulte en des graphes de transport de taille légèrement plus élevée augmentant ainsi le nombre de contraintes forward et backward. L'effet de l'usage des blocs fictifs sur le nombre de variables de décision et le nombre de coefficients non nuls dépend étroitement des règles d'admissibilité des quarts et diffère d'un jeu de données à un autre selon le type de quart adopté.

Pour le cas du JEU2, l'ajout des blocs fictifs permet la diminution des coefficients non nuls. La Figure 4.4 illustre un sous-graphe de transport extrait de celui du type de quart T2. Le couple (s,e) désigne l'heure de début s et de fin e de chaque segment ou chaque quart vide. Les heures de débuts admissibles de travail sont 0h00 et 2h00 et les durées sont 6h, 8h et 10h. Pour simplifier la représentation, on se limite aux segments de début de durée entre 6h et 10h. Comme on peut le voir, l'option « avec blocs fictifs » permet d'utiliser le même bloc pour modéliser un bloc de début et un quart simultanément. Ceci a la vocation de diminuer le nombre de coefficients non nuls dans les contraintes de demande et les contraintes horizontales de liaison entre quarts.

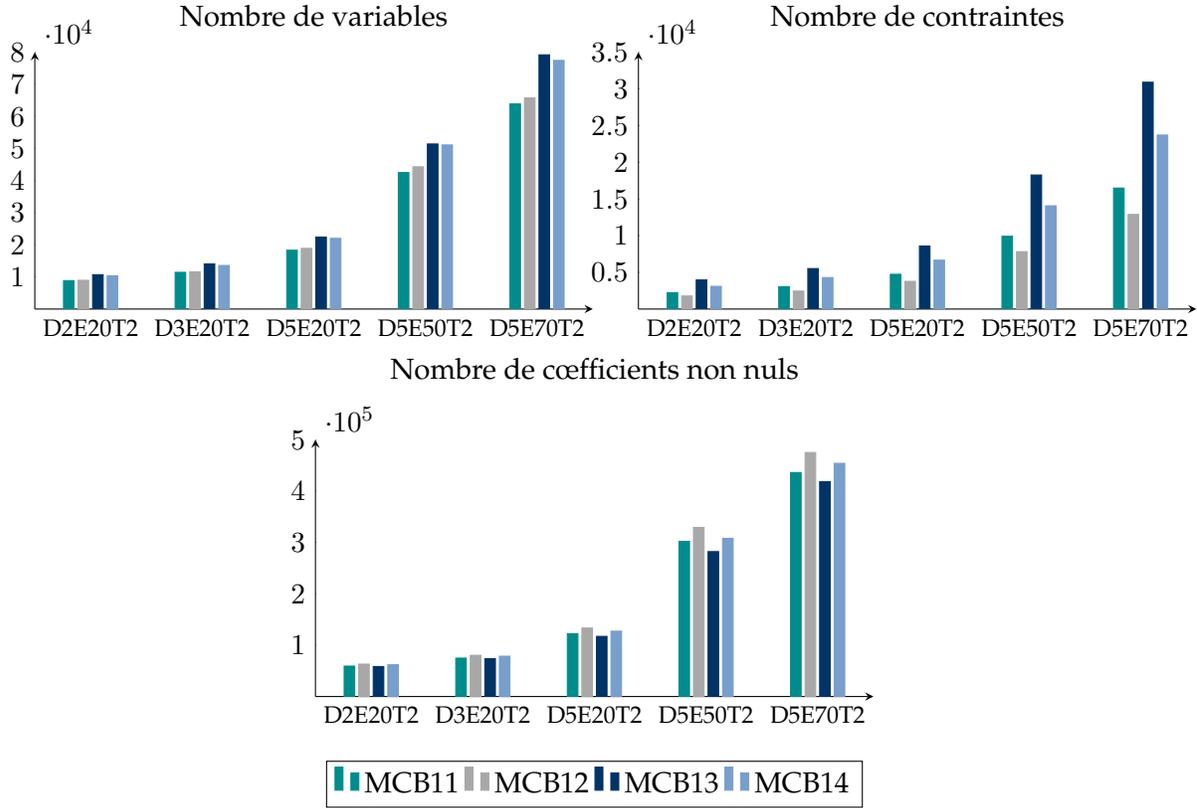


Figure 4.3 – Taille des modèles implicites de construction de blocs pour les instances du JEU2

Passons maintenant à comparer les quatre versions du modèle implicite de construction de blocs en se basant sur le temps de calcul et la qualité des solutions trouvées. D’abord, on a résolu les 20 instances des classes D2E20T2, D3E20T2, D5E20T2, D5E50T2 et D5E70T2 du JEU2 en utilisant CPLEX avec un gap limite égal à 1% et un temps limite de résolution égal à 1 heure. Le gap d’optimalité est donné par l’expression suivante :

$$Gap = \frac{f_{IP}^* - f_{LP}}{f_{IP}^*}$$

avec f_{IP}^* et f_{LP} correspondant respectivement à la meilleure borne supérieure et à la meilleure borne inférieure dans l’arbre de branchement. f_{IP}^* est la meilleure solution entière trouvée du problème en nombres entiers et f_{LP} est la meilleure solution de la relaxation linéaire du problème au niveau des nœuds évalués et non explorés. Dans le reste de ce chapitre, une instance est considérée comme résolue à l’optimalité par CPLEX lorsque la meilleure solution trouvée est prouvée être à 1% ou moins de l’optimum (c.-à-d. $Gap \leq 1\%$). Dans le cas contraire, la meilleure solution trouvée est dite réalisable avec un $Gap > 1\%$.

Le Tableau 4.1 est une représentation matricielle permettant de comparer chaque paire de modèles. Une ligne i correspond à un modèle implicite MCB1i et Une colonne j corres-

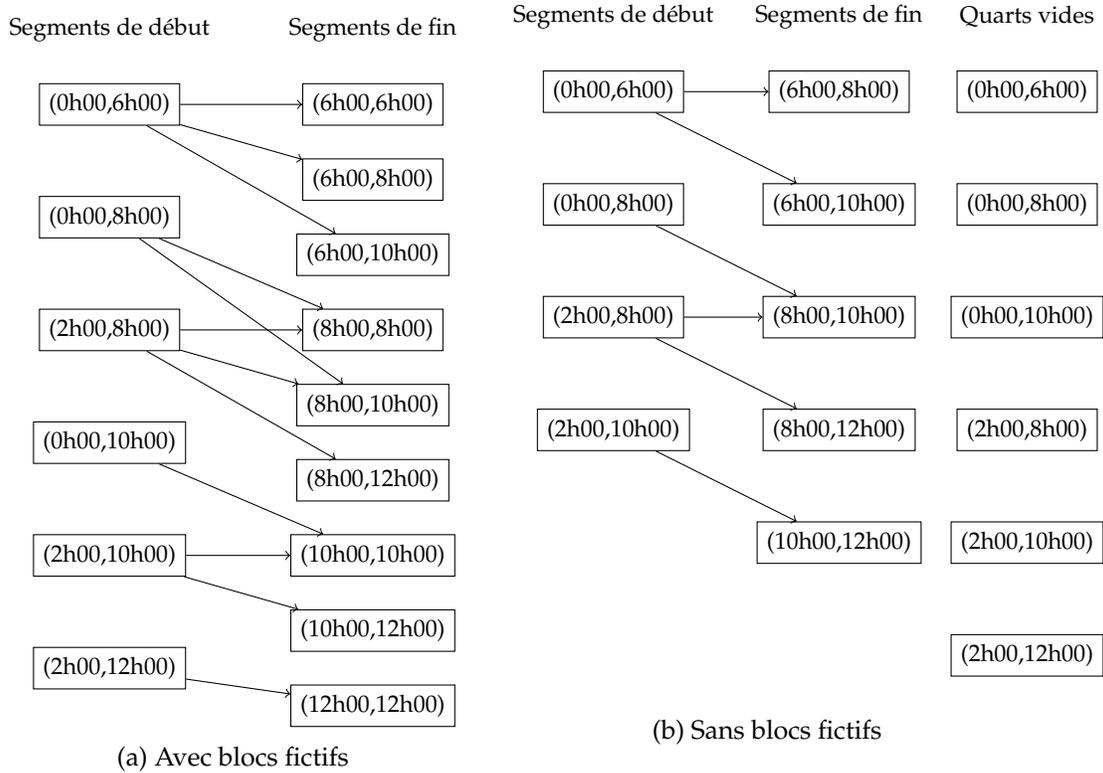


Figure 4.4 – Effet de l’utilisation des blocs fictifs sur le modèle MCB1

pond à un modèle implicite MCB1j. Pour chaque ligne i et chaque colonne j , on donne le nombre d’instances qui ont été résolues à l’optimalité par les deux modèles implicites MCB1i et MCB1j tel que le temps de résolution en utilisant le modèle MCB1i est inférieur à celui obtenu en utilisant le modèle MCB1j. De plus, le gain moyen en temps de calcul offert par le modèle le plus performant des deux est donné. Prenons l’exemple des modèles MCB11 et MCB12. Il y a au total 16 instances parmi les 20 testées qui ont été résolues à l’optimalité par ces deux modèles. Le modèle MCB11 était plus rapide que MCB12 à résoudre 10 instances alors que le modèle MCB12 était plus rapide que MCB11 à résoudre les six instances restantes. Lorsque le modèle MCB11 était plus performant que MCB12, il a permis un gain moyen en temps de calcul égal à 340.99 secondes sur les 10 instances identifiées alors que le modèle MCB12 a permis un gain moyen beaucoup moins important égal à 15.9 secondes sur les six instances pour lesquelles il était meilleur que MCB11.

Le Tableau 4.2 donne pour chaque ligne i et chaque colonne j le nombre d’instances qui ont été résolues à l’optimalité par le modèle MCB1i et non résolues à l’optimalité par le modèle MCB1j. Le temps de calcul moyen du modèle MCB1i et le gap d’optimalité moyen des solutions réalisables trouvées lors de la résolution du modèle MCB1j sont donnés. Prenons l’exemple des modèles MCB11 et MCB12. Rappelons que 16 instances parmi les 20 testées ont été résolues à l’optimalité par les deux modèles. Pour les quatre restantes, il existe deux

Tableau 4.1 – Instances du JEU2 résolues à l’optimalité par les deux modèles implicites MCB1i et MCB1j : # d’instances et écarts des temps de calcul en secondes

		MCB1j							
		MCB11		MCB12		MCB13		MCB14	
MCB1i	MCB11	–		10	340.99	11	278.66	10	321.52
	MCB12	6	15.9	–		7	20.53	6	6.75
	MCB13	6	149.39	9	433.07	–		8	165.72
	MCB14	7	68.84	10	296.39	9	84.61	–	

instances résolues uniquement par le modèle MCB11 à l’optimalité. De plus, le temps de calcul moyen de la résolution du modèle MCB11 est de 1,202 secondes et le gap d’optimalité moyen des solutions réalisables trouvées en utilisant le modèle MCB12 pour résoudre ces deux instances est égal à 5.57%.

Tableau 4.2 – Instances du JEU2 résolues à l’optimalité par le modèle MCB1i uniquement : # d’instances, temps de calcul du modèle MCB1i en secondes et gaps d’optimalité du modèle MCB1j en %

		MCB1j											
		MCB11		MCB12		MCB13		MCB14					
MCB1i	MCB11	–		2	1,202	5.57%	1	1,972	3.92%	1	1,972	8.02%	
	MCB12	0		–		0		0		0		0	
	MCB13	1	3,600	4.91%	2	3,321	28.25%	–		1	3,600	32.24%	
	MCB14	0		1	2,775	7.8%	0		–		–		

Le Tableau 4.3 donne pour chaque ligne i et chaque colonne j le nombre d’instances qui n’ont été résolues à l’optimalité ni par le modèle MCB1i ni par le modèle MCB1j tel que la meilleure solution trouvée par le modèle MCB1i soit de qualité meilleure que celle trouvée par le modèle MCB1j. On donne également le gap d’optimalité moyen des solutions réalisables trouvées par le modèle MCB1i puis par le modèle MCB1j. Prenons l’exemple des modèles MCB11 et MCB12. Rappelons que 16 instances parmi les 20 testées ont été résolues à l’optimalité par les deux modèles et deux instances ont été résolues à l’optimalité uniquement par le modèle MCB11. Concernant les deux instances restantes, aucun des deux modèles n’a réussi à trouver une solution optimale. De plus, c’est le modèle MCB11 qui a réussi à identifier les meilleures solutions réalisables ayant un gap d’optimalité moyen égal à 10.88% alors que cette valeur est de 53.28% pour le modèle MCB12.

À partir des Tableaux 4.1, 4.2 et 4.3, il est clair que le modèle MCB11 domine sur l’ensemble des 20 instances testées le reste des modèles implicites de construction de blocs. Lorsque des solutions optimales ont été trouvées par MCB11 et tout autre modèle MCB1j, le modèle MCB11 était le plus rapide dans la plupart des cas avec un écart en temps de calcul plus important que celui obtenu lorsque l’autre modèle MCB1j le surpassait. Pour une seule ins-

Tableau 4.3 – Instances du JEU2 non résolues à l’optimalité par les deux modèles MCB1i et MCB1j : # d’instances et gaps d’optimalité du modèle MCB1i puis MCB1j en %

		MCB1j									
		MCB11		MCB12		MCB13		MCB14			
MCB1i	MCB11	–	2	10.88%	53.28%	1	16.86%	30.68%	2	10.88%	31.00%
	MCB12	0		–		1	3.34%	3.92%	1	3.34%	8.02%
	MCB13	0	1	30.68%	57.87%		–		1	3.92%	8.02%
	MCB14	0	2	31.00%	53.28%	1	29.76%	30.68%		–	

tance, le modèle MCB11 a identifié une solution réalisable avec un gap d’optimalité de 4.91% lorsque le modèle MCB13 a identifié une solution optimale. Finalement, lorsqu’aucune solution optimale n’a été identifiée ni par MCB11 ni par tout autre modèle MCB1j, la solution réalisable retrouvée par le modèle MCB11 était tout le temps de qualité meilleure.

Passons maintenant à comparer les deux modèles de construction de quarts MCQ et de construction de blocs MCB11 en se référant aux 24 instances du JEU2. Nous ajoutons ici les quatre instances de la plus grande classe D5E200T2 du JEU2 pour comparer le comportement des deux formulations même lorsque CPLEX ne réussit pas à identifier des solutions optimales. Le Tableau 4.4 donne le nombre de variables de décision, de contraintes et de coefficients non nuls par classe d’instances et par modèle. On a identifié en gras le minimum obtenu par chacun des deux modèles. On remarque que le nombre de variables de décision du modèle de construction de quarts est dans la plupart des cas plus petit que celui du modèle de construction de blocs. Le modèle implicite de construction de blocs induit un nombre plus élevé de contraintes étant donné qu’on rajoute les contraintes forward et backward ainsi que les contraintes de liaison entre départements. Par contre, on observe que le nombre de coefficients non nuls du modèle MCQ excède toujours celui du modèle MCB11. Ceci est dû notamment aux coefficients dans les contraintes de demande (4.29) et de liaison entre quarts ((4.34) - (4.37)). Par exemple, pour la contrainte de respect de la séquence des jours de travail et de repos pour chaque employé, tous les quarts qui débutent à 0h00 au département d durant 4h puis se terminent dans n’importe quel autre département d' durant 2h, 4h ou 6h ont pour coefficient 1 dans la contrainte (4.8) alors que seulement le bloc de début qui commence à 0h00 au département d durant 4h a pour coefficient 1 dans la contrainte (4.34) indépendamment du bloc de fin qui lui correspond.

Le tableau 4.5 donne le temps de calcul, la valeur de la fonction objectif et le gap d’optimalité pour les 24 instances du JEU2 et pour les deux modèles MCB11 et MCQ. Nous avons identifié en gras, d’une part, le meilleur coût trouvé et, d’autre part, le meilleur temps de calcul inférieur à la limite de temps. Pour analyser les résultats, nous séparons les 12 instances issues des classes D2E20T2, D3E20T2 et D5E20T2 et les 12 instances issues des classes D5E50T2, D5E70T2 et D5E200T2. Pour les 12 premières instances, les deux modèles MCB11

Tableau 4.4 – Taille des modèles MCB11 et MCQ pour les classes d’instances du JEU2

Classe	# Variables		# Contraintes		# Coefficients non nuls	
	MCB11	MCQ	MCB11	MCQ	MCB11	MCQ
D2E20T2	8,851	7,436	2,218	373	59,213	59,579
D3E20T2	11,436	9,448	3,056	457	74,744	75,387
D5E20T2	18,313	17,022	4,736	625	122,407	136,439
D5E50T2	42,498	42,596	9,911	932	301,983	350,775
D5E70T2	63,873	57,353	16,482	1,138	435,948	476,426
D5E200T2	176,417	159,481	44,750	2,472	1,214,608	1,331,968

et MCQ réussissent à identifier des solutions optimales ($\text{Gap} \leq 1\%$). Le modèle MCQ est plus performant que le modèle MCB11 pour la totalité de ces 12 instances avec un écart moyen de temps de calcul égal à 33.4 secondes. Pour les 12 instances restantes, elles peuvent être classées en quatre catégories :

- Instances marquées par (1) : il s’agit de cinq instances résolues à l’optimalité par les deux modèles MCB11 et MCQ. D’une part, le modèle de construction de quarts MCQ était le plus rapide à résoudre trois instances parmi les cinq avec un écart moyen en temps de résolution de 142.4 secondes. D’autre part, le modèle de construction de blocs MCB11 était le plus rapide à résoudre les deux instances restantes avec un écart moyen en temps de résolution beaucoup plus important de 1,382.6 secondes.
- Instances marquées par (2) : il s’agit de deux instances résolues à l’optimalité par le modèle MCB11 mais pas par le modèle MCQ. Le gap d’optimalité moyen des solutions réalisables trouvées par le modèle de construction de quarts est égal à 6.73% alors que le temps de résolution moyen du modèle de construction de blocs MCB11 est égal 971.25 secondes.
- Instances marquées par (3) : il s’agit de deux instances qui ne sont résolues à l’optimalité ni par le modèle MCB11 ni par le modèle MCQ tel que la meilleure solution réalisable trouvée par MCB11 soit meilleure que celle trouvée par MCQ. Le gap d’optimalité moyen de ces deux solutions réalisables est égal à 10.88% pour le modèle MCB11 et de 26.81% pour le modèle MCQ.
- Instances marquées par (4) : il s’agit de trois instances qui ne sont résolues à l’optimalité ni par le modèle MCB11 ni par le modèle MCQ tel que la meilleure solution réalisable trouvée par MCQ soit meilleure que celle trouvée par MCB11. Le gap d’optimalité moyen de ces trois solutions réalisables est égal à 11.90% pour le modèle MCQ et de 46.72% pour le modèle MCB11.

Lorsqu’on essaie de comprendre la différence entre les 12 instances issues des petites classes et les 12 instances issues des grandes classes, on remarque que le nombre de qualifications par employé augmente avec la taille de l’instance (voir Figure 4.5). Ce paramètre structurel définissant les organisations permet au modèle de blocs de réunir dans un segment de dé-

Tableau 4.5 – Résultats obtenus par les modèles MCB11 et MCQ pour les instances du JEU2

Instance	Temps de résolution		Fonction objectif		Gap d'optimalité	
	(secondes)		(\$)		(%)	
	MCB11	MCQ	MCB11	MCQ	MCB11	MCQ
D2E20T2P1	3.29	1.99	251.50	251.50	0.00	0.00
D2E20T2P2	13.00	10.91	355.90	355.80	0.18	0.00
D2E20T2P3	32.04	29.35	318.50	318.50	0.00	0.00
D2E20T2P4	154.61	12.49	319.00	319.00	0.00	0.00
D3E20T2P1	20.69	6.40	262.10	262.20	0.04	0.06
D3E20T2P2	8.53	2.83	504.90	505.00	0.00	0.02
D3E20T2P3	19.15	9.32	402.40	402.40	0.80	0.00
D3E20T2P4	18.62	7.52	513.90	514.00	0.00	0.06
D5E20T2P1	13.31	7.30	345.40	345.50	0.00	0.03
D5E20T2P2	195.18	57.76	458.70	457.90	0.39	0.00
D5E20T2P3	80.31	21.16	450.50	450.50	0.39	0.76
D5E20T2P4	18.25	9.13	770.10	770.10	0.00	0.00
D5E50T2P1 (1)	353.23	215.20	538.40	540.60	0.00	0.41
D5E50T2P2 (1)	446.29	158.19	754.40	753.70	0.22	0.14
D5E50T2P3 (2)	1,972.18	3,600.20	527.70	572.50	0.07	7.90
D5E50T2P4 (1)	423.10	2,229.87	808.20	809.60	0.00	0.20
D5E70T2P1 (1)	432.33	431.24	660.50	659.90	0.11	0.02
D5E70T2P2 (1)	392.03	1,350.55	996.40	996.70	0.00	0.03
D5E70T2P3 (3)	3,600.28	3,600.36	889.10	1,019.10	16.86	27.47
D5E70T2P4 (3)	3,600.36	3,600.29	756.30	969.30	4.91	26.16
D5E200T2P1 (2)	3,547.08	3,601.00	1,414.80	1,497.40	0.05	5.57
D5E200T2P2 (4)	3,600.40	3,600.69	3,730.70	1,919.70	53.04	8.80
D5E200T2P3 (4)	3,600.59	3,600.59	3,503.20	2,365.70	41.11	12.80
D5E200T2P4 (4)	3,600.49	3,600.54	3,461.90	2,175.60	46.02	14.11

but ou de fin un nombre plus important de blocs comprenant tous les blocs ayant la même heure de début et la même durée que le segment en question et se réalisant dans différents départements.

Nous concluons que le modèle MCQ est plus performant que le modèle MCB11 pour les petites instances et que les deux modèles de construction de quarts MCQ et de blocs MCB11 deviennent compétitifs pour les instances de taille plus importante, notamment lorsque les employés possèdent un nombre plus important de départements d'accueil.

4.6.4 Tests expérimentaux du JEU3

Comme dans la partie 4.6.3, commençons par comparer les quatre versions de la formulation implicite basée sur les blocs MCB2. La Figure 4.6 montre le nombre de variables, de

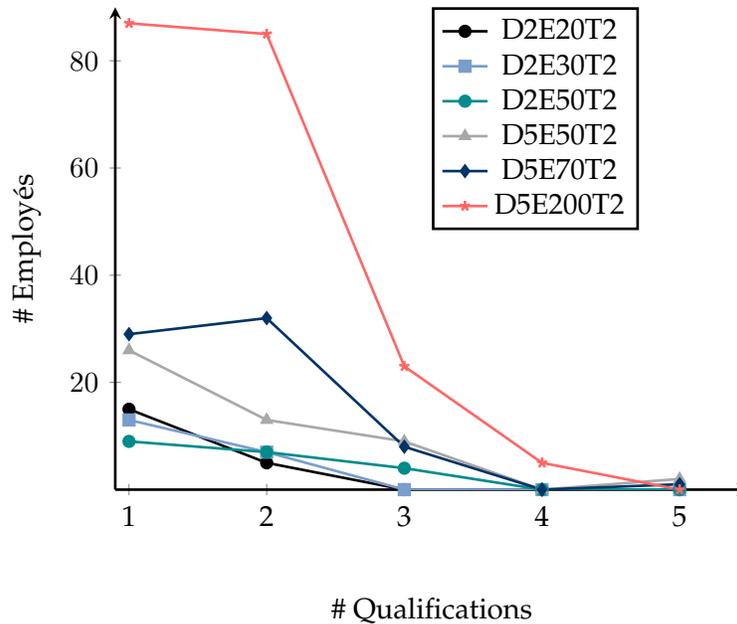


Figure 4.5 – Qualifications internes et externes des employés

contraintes et de coefficients non nuls lorsque les formulations MCB21, MCB22, MCB23 et MCB24 sont utilisées pour modéliser le problème correspondant aux instances issues des trois classes du JEU3 (rappelons que le jeu de données JEU3 renferme trois classes).

D'une part, nous analysons l'effet de l'utilisation des contraintes de liaison entre départements sur la taille du modèle réduit de construction de blocs MCB2. Rappelons que les formulations MCB21 et MCB23 utilisent toutes les deux des variables de blocs fictifs et diffèrent uniquement du point de vue de l'utilisation des contraintes de liaison entre départements. De même pour les formulations MCB22 et MCB24. Comme précédemment et pour les mêmes raisons, l'utilisation des contraintes de liaison entre départements permet d'obtenir des modèles avec moins de variables de décision, moins de contraintes mais plus de coefficients non nuls.

D'autre part, nous analysons l'effet de l'utilisation des blocs fictifs sur la taille du modèle réduit de construction de blocs MCB2. Rappelons que les formulations MCB21 et MCB22 utilisent toutes les deux des contraintes de liaison entre départements et diffèrent uniquement du point de vue de l'utilisation des blocs fictifs. De même pour les formulations MCB23 et MCB24. L'utilisation des blocs fictifs diminue le nombre de variables et le nombre de coefficients non nuls. Le nombre de contraintes quant à lui reste le même. Ceci est dû aux règles d'admissibilité des quarts et à la définition des graphes de transport. La Figure 4.7 illustre un sous-graphe correspondant aux quarts vides débutant à 1h et à 1h15 et de durée entre 7h 30min et 7h 45min. Étant donné que le nombre de nœuds au niveau desquels les contraintes forward et backward s'écrivent ne change pas, le nombre de contraintes ne change pas à

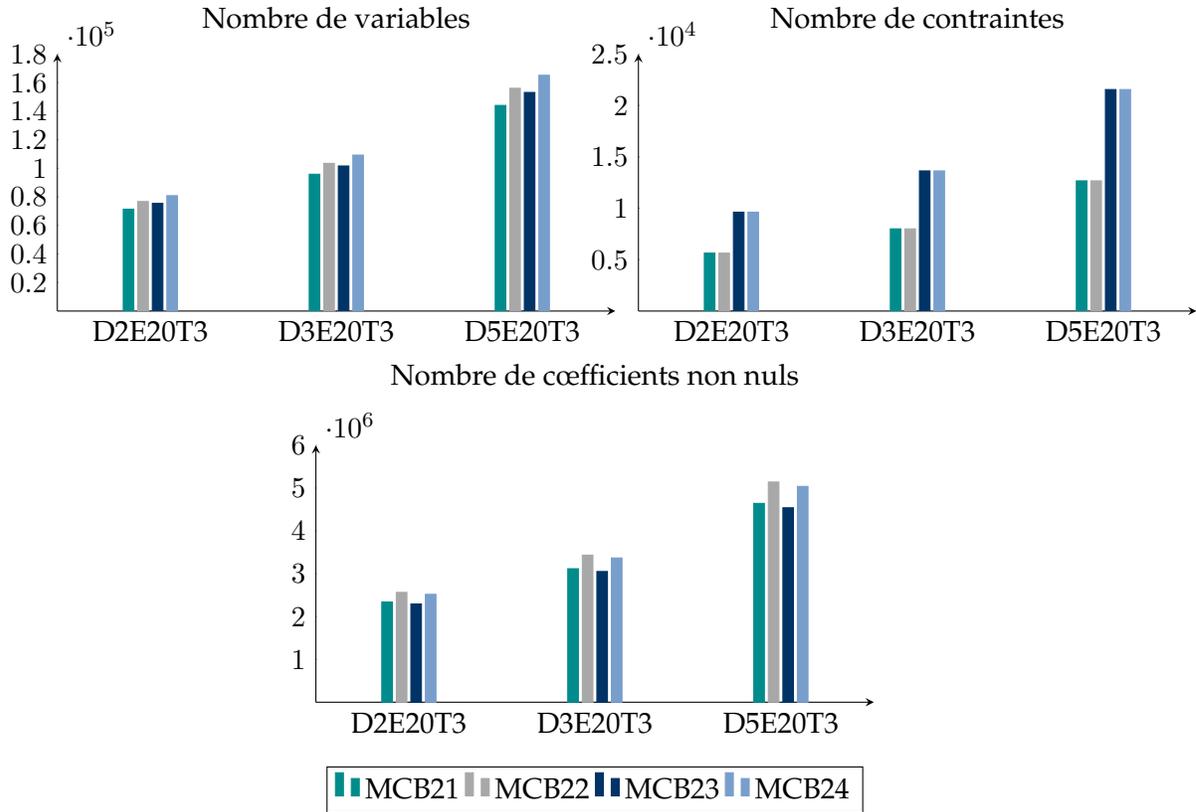


Figure 4.6 – Taille des modèles implicites réduits de construction de blocs pour les instances du JEU3

son tour. Il est facile de vérifier que cette affirmation reste valable pour les graphes de transport au complet. En conclusion, l'utilisation des blocs fictifs diminue la taille des modèles résultants.

Les Tableaux 4.6, 4.7 et 4.8 sont similaires aux Tableaux 4.1, 4.2 et 4.3 et illustrent les résultats obtenus pour les quatre versions du modèle réduit de construction de blocs MCB2 lorsqu'elles sont testées sur les 12 instances du JEU3 selon la même démarche.

Le Tableau 4.6 montre que lorsque des solutions optimales sont trouvées par MCB24 et par chacun des autres modèles MCB2j, le modèle MCB24 est tout le temps le plus rapide avec des écarts moyens significatifs en temps de calcul. Par exemple, le modèle MCB24 et le modèle MCB21 réussissent à identifier des solutions optimales pour cinq instances et le modèle MCB24 est tout le temps le plus rapide des deux avec un écart moyen de temps de calcul de 1,027.63 secondes.

Le Tableau 4.7 montre que lorsque des solutions optimales sont identifiées par MCB24 et des solutions réalisables sont trouvées par chacun des autres modèles, le temps de calcul moyen du modèle MCB24 est relativement petit. Par exemple, il existe trois instances pour

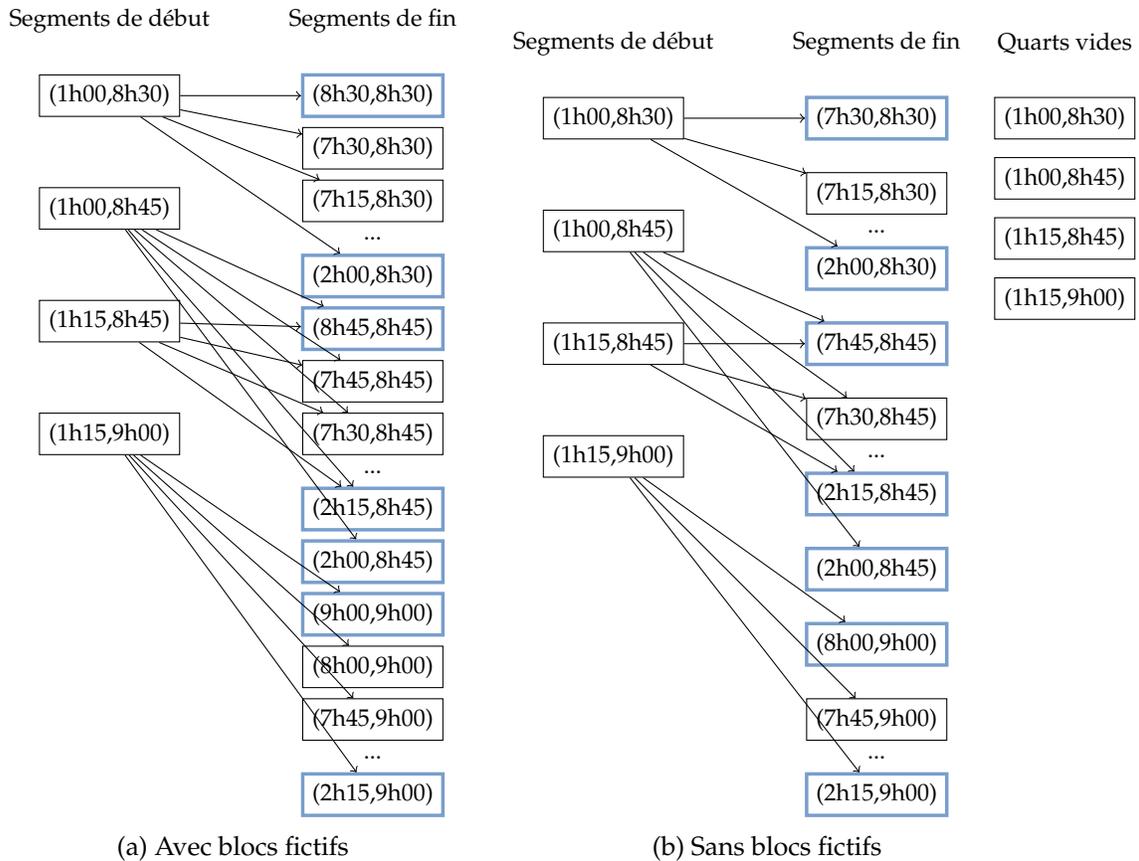


Figure 4.7 – Effet de l’utilisation des blocs fictifs sur le modèle MCB2

Tableau 4.6 – Instances du JEU3 résolues à l’optimalité par les deux modèles implicites MCB2i et MCB2j : # d’instances et écarts des temps de calcul en secondes

		MCB2j					
		MCB21	MCB22	MCB23	MCB24		
MCB2i	MCB21	–	2	1,774.56	0	0	
	MCB22	1	1,101.80	–	0	0	
	MCB23	5	719.87	3	1,504.26	–	0
	MCB24	5	1,027.63	3	1,570.35	7	467.4

lesquelles le modèle MCB24 réussit à trouver des solutions optimales. Le temps de calcul moyen de ces trois instances est égal à 600.53 secondes alors que le gap d’optimalité moyen des solutions réalisables trouvées par MCB21 est égal à 2.83%. Au contraire, il y a une seule instance que le modèle MCB24 n’a pas réussi à la résoudre à l’optimalité alors que le modèle MCB21 a identifié une solution optimale au bout de 3,306.48 secondes. Le gap d’optimalité de la solution réalisable trouvée par le modèle MCB24 est égal à 1.58%.

Le Tableau 4.8 montre que lorsque seulement des solutions réalisables ont été identifiées par MCB24 et par chacun des autres modèles, il existe une seule instance D5E20T3P4 pour

Tableau 4.7 – Instances du JEU3 résolues à l’optimalité par le modèle MCB2i uniquement : # d’instances, temps de calcul du modèle MCB2i en secondes et gaps d’optimalité du modèle MCB2j en %

		MCB2j										
		MCB21		MCB22		MCB23		MCB24				
MCB2i	MCB21	–		3	2,154.18	3.26%	1	3,306.48	1.06%	1	3,306.48	1.58%
	MCB22	0			–		0			0		
	MCB23	2	1,522.87	2.47%	4	1,166.98	3.57%		–		0	
	MCB24	3	600.53	2.83%	5	416.64	6.85%	1	488.86	1.69%		–

laquelle le modèle MCB24 a identifié la solution ayant la valeur de l’objectif la plus élevée. Ceci dit, le gap d’optimalité est acceptable (égal à 4.21%).

Tableau 4.8 – Instances du JEU3 non résolues à l’optimalité par les deux modèles MCB2i et MCB2j : # d’instances et gaps d’optimalité du modèle MCB2i puis MCB2j en %

		MCB2j										
		MCB21		MCB22		MCB23		MCB24				
MCB2i	MCB21	–		1	2.61%	5.61%	1	2.61%	2.97%	1	2.61%	4.21%
	MCB22	0			–		0			1	5.61%	4.21%
	MCB23	0		1	2.97%	5.61%		–		1	2.97%	4.21%
	MCB24	0		0			0				–	

À partir de ces trois Tableaux, il est clair que le modèle MCB24 domine sur l’ensemble des 12 instances testées le reste des modèles réduits de construction de blocs. Donc, il s’avère judicieux d’introduire deux graphes de transport distincts par employé, par jour et par type de quart afin de lier les blocs de début externes aux blocs de fin internes et vice-versa au lieu d’ajouter des contraintes de liaison entre départements. De plus, l’ajout des blocs fictifs n’est pas adéquat. Dans ce qui suit, nous allons comparer le modèle de construction de quarts MCQ, le modèle de construction de blocs MCB11 et le modèle réduit de construction de blocs MCB24 en se basant sur les instances du JEU 3.

Le Tableau 4.9 donne le nombre de variables de décision, le nombre de contraintes et le nombre de coefficients non nuls pour les classes d’instances D2E20T3, D3E20T3 et D5E20T3 et pour les modèles MCQ, MCB11 et MCB24. Il était prévisible que le modèle réduit de construction de blocs MCB24 fasse intervenir le minimum de variables de décision étant donné que l’on énumère des quarts vides et des segments de fin que nous remplissons par les départements admissibles correspondants. Bien évidemment, le modèle de quarts est celui qui requiert le nombre de contraintes le moins élevé étant donné que la liaison entre blocs est assurée explicitement lors de l’énumération des quarts admissibles. De plus, il est important de souligner que le modèle MCB24 nécessite moins de contraintes que le modèle MCB11. Concernant le nombre de coefficients non nuls, c’est le modèle MCB11 qui est le

meilleur entre les trois modèles puis vient le modèle MCB24 dont le nombre se rapproche beaucoup de celui du modèle MCB11 et finalement vient le modèle de construction de quarts MCQ qui implique un nombre très élevé de coefficients non nuls.

Tableau 4.9 – Taille des modèles MCB11, MCB24 et MCQ pour les classes d’instances du JEU3

	Modèle	Classe d’instances		
		D2E20T3	D3E20T3	D5E20T3
# Variables	MCB11	133,514	183,762	282,572
	MCB24	80,829	109,250	165,159
	MCQ	176,119	244,018	377,521
# Contraintes	MCB11	35,034	49,578	78,188
	MCB24	9,619	13,635	21,551
	MCQ	1,549	2,221	3,565
# Coefficients $\neq 0$	MCB11	2,458,826	3,287,324	4,914,762
	MCB24	2,518,400	3,368,839	5,039,556
	MCQ	6,232,488	8,623,197	13,319,520

Passons maintenant à comparer les trois modèles MCQ, MCB11 et MCB24 en se référant aux temps de résolution et à la qualité des solutions obtenues. Le tableau 4.10 illustre le temps de résolution, la valeur de la fonction objectif et le gap d’optimalité pour les 12 instances du JEU3. Premièrement, on remarque que le modèle MCB11 est dominé par les modèles MCQ et MCB24 de point de vue temps de résolution et qualité des solutions. Ceci est prévisible du moment que ce modèle trouve sa pertinence du fait qu’un employé possède plusieurs qualifications à la fois ce qui n’est pas le cas pour les instances du JEU3 qui stipulent que chaque employé possède au plus une qualification externe (en plus de sa qualification interne). Nous allons exclure ainsi le modèle MCB11 de l’analyse et nous allons limiter la comparaison au niveau des modèles MCQ et MCB24.

L’ensemble des 12 instances testées peuvent être classées en quatre catégories :

- Instances marquées par (1) : il s’agit de six instances résolues à l’optimalité par les deux modèles MCB24 et MCQ. Le modèle de construction de blocs MCB24 est le plus rapide à résoudre toute ces instances avec un écart moyen en temps de résolution significatif égal à 913.15 secondes.
- Instances marquées par (2) : il s’agit de deux instances résolues à l’optimalité par le modèle MCB24 mais pas par le modèle MCQ. Le gap d’optimalité moyen des solutions réalisables trouvées par le modèle de construction de quarts est égal à 2.35% alors que le temps de résolution moyen du modèle de construction de blocs MCB24 est égal à 793.63 secondes.
- Instances marquées par (3) : il s’agit de deux instances résolues à l’optimalité par le modèle MCQ mais pas par le modèle MCB24. Le gap d’optimalité moyen des solutions

Tableau 4.10 – Résultats obtenus par les modèles MCB11, MCB24 et MCQ pour les instances du JEU3

Instance	Temps de résolution (secondes)			Fonction objectif (\$)			Gap d'optimalité (%)		
	MCB11	MCB24	MCQ	MCB11	MCB24	MCQ	MCB11	MCB24	MCQ
D2E20T3P1 (1)	3,600.7	136.0	535.1	2,098.6	2,095.3	2,093.7	16.03	0.99	0.93
D2E20T3P2 (1)	3,601.1	108.4	134.3	1,667.7	1,623.8	1,656.9	11.28	0.43	0.76
D2E20T3P3 (2)	3,600.4	1,098.4	3,601.7	1,477.5	1,399.6	1,424.9	19.95	0.64	2.05
D2E20T3P4 (3)	3,600.5	3,600.4	3,297.2	1,344.0	1,265.0	1,269.3	19.41	1.58	0.86
D3E20T3P1 (1)	3,600.8	241.6	830.4	2,200.7	2,187.8	2,182.3	15.53	0.91	0.74
D3E20T3P2 (1)	3,600.9	119.6	930.8	1,904.7	1,885.6	1,891.7	10.37	0.98	0.88
D3E20T3P3 (1)	3,600.8	162.0	2,541.7	1,759.9	1,710.9	1,735.1	13.73	0.99	0.89
D3E20T3P4 (1)	3,601.0	214.3	1,488.4	1,439.2	1,415.6	1,441.9	8.83	0.98	1.00
D5E20T3P1 (2)	3,604.0	488.9	3,601.8	2,090.7	2,043.4	2,084.3	11.99	0.40	2.65
D5E20T3P2 (4)	3,600.9	3,600.8	3,602.1	1,361.6	1,353.7	1,365.7	4.88	4.11	4.71
D5E20T3P3 (3)	3,601.2	3,601.8	1,600.6	1,303.4	1,278.6	1,272.7	10.54	1.42	0.97
D5E20T3P4 (4)	3,600.7	3,600.9	3,603.2	1,435.5	1,442.9	1,424.5	6.34	4.21	2.70

réalisables trouvées par le modèle réduit de construction de blocs est égal à 1.50% alors que le temps de résolution moyen du modèle de construction de quarts MCQ est égal à 2,448.92 secondes. Il est à noter que pour l'une de ces deux instances (D2E20T3P4), le modèle MCB24 a trouvé une solution de meilleure qualité avec une valeur de l'objectif égale à 1,264.96 même si elle a un gap d'optimalité supérieur à celui de la solution trouvée par MCQ. En d'autres termes, les solutions trouvées par le modèle MCB24 sont de bonne qualité.

- Instances marquées par (4) : il s'agit de deux instances qui ne sont résolues à l'optimalité ni par le modèle MCB24 ni par le modèle MCQ. Dans l'une des deux, la meilleure solution réalisable trouvée par MCB24 est meilleure que celle trouvée par MCQ et dans l'autre, on observe le contraire.

D'après la synthèse des résultats, le modèle réduit de construction de blocs MCB24 s'avère meilleur que le modèle de construction de quarts MCQ étant donné qu'il est le plus rapide à résoudre par CLPEX et donne des solutions de meilleure qualité dans la plupart des cas. Ce modèle correspond donc à une contribution de la thèse et il sera utilisé ultérieurement pour résoudre des problèmes similaires à ceux du JEU3.

En conclusion, nous avons mené une analyse comparative de la complexité de plusieurs formulations proposées. La performance de ces formulations est étroitement liée à la définition des règles de travail et de transfert ainsi que les paramètres structurels régissant les qualifications externes des employés pour les départements de l'organisation. Deux contributions sont à souligner :

- À travers cette partie expérimentale, nous avons pu voir que les modèles de construction de blocs représentent une alternative à la formulation basée sur les quarts lorsque le nombre de qualifications externes des employés est important. Dans le cas contraire, c'est le modèle MCQ qui est le plus performant. L'une des perspectives futures est de questionner la compétitivité entre le modèle implicite de construction de blocs et le modèle de construction de quarts en se basant sur une procédure autre que CPLEX permettant de résoudre des instances de plus grande taille. Dans le cadre de la thèse, on se contente d'introduire le modèle implicite de construction de blocs et de mettre en évidence son cadre d'application. Nous avons décidé de garder le modèle MCQ ayant prouvé son efficacité.
- Nous avons aussi introduit un modèle réduit de construction de blocs qui est plus performant que le modèle de construction de quarts et qui s'applique à un ensemble spécifique de problèmes stipulant qu'on connaît à priori le département d'accueil vers lequel un employé donné pourrait être transféré durant un jour de travail donné.

Chapitre 5

Approche de résolution basée sur la génération de colonnes

Ce chapitre est une description de la procédure basée sur la génération de colonnes que nous utilisons pour la résolution de certains problèmes spécifiques définis dans nos heuristiques de décomposition. La section 5.1 rappelle certaines notions théoriques portant sur la résolution de problèmes linéaires en nombres entiers. La section 5.2 décrit notre procédure basée sur la génération de colonnes. Finalement, la section 5.4 est une étude expérimentale de la performance de cette procédure.

5.1 Notions préliminaires

Dans cette section, nous rappelons un ensemble de méthodes de résolution des problèmes linéaires en nombres entiers. Notons que nous considérons le cas d'un problème de minimisation.

5.1.1 Méthode de séparation et d'évaluation progressive

En optimisation combinatoire, la méthode de séparation et d'évaluation progressive ou encore le « *Branch and Bound* » est une méthode d'énumération implicite permettant la résolution des problèmes linéaires en nombres entiers.

La séparation est le processus permettant le partitionnement du domaine réalisable pour diviser itérativement le problème en plusieurs sous-problèmes de taille plus petite. Ce mécanisme d'énumération se fait en utilisant un arbre de recherche dont les nœuds correspondent aux sous-problèmes définis lors du branchement. La séparation est le résultat de la Proposition 3 (Lawler et Wood, 1966).

Proposition 3.

Soit le problème linéaire en nombres entiers :

$$(PE) : z^* = \min\{c^T x \mid x \in \mathbb{X}\}, \mathbb{X} = \{x \mid Ax \leq b, x \in \mathbb{Z}_+^n\}, c \in \mathbb{R}^n, b \in \mathbb{R}^m, A \in \mathbb{R}^{m \times n}$$

Soit $\{\mathbb{X}_1, \mathbb{X}_2, \dots, \mathbb{X}_l\}$ une partition de l'ensemble \mathbb{X} .

Considérons les sous-problèmes suivants :

$$(PE_k) : z_k^* = \min\{cx \mid x \in \mathbb{X}_k\}$$

$$\text{Alors, } z^* = \min_{k \in \{1, 2, \dots, l\}} z_k^*$$

L'évaluation permet d'éviter l'énumération exhaustive de toutes les solutions. On parle plutôt d'une énumération implicite qui se base sur certaines règles permettant ainsi l'exploration efficace de l'arbre de recherche. À chaque nœud de l'arbre, une borne inférieure est calculée par la résolution de la relaxation linéaire d'un sous-problème spécifique. Ce sous-problème correspond au problème initial augmenté des décisions de branchement imposées au niveau de la branche. L'exploration de cette branche est interrompue lorsque le nœud n'est pas prometteur, et plus spécifiquement lorsque l'une de ces trois conditions est vérifiée :

- la solution du sous-problème relaxé est entière (il s'agit donc de la meilleure solution entière que l'on peut trouver sur la branche),
- le sous-problème relaxé est non réalisable (il n'existe donc aucune solution entière sur la branche),
- la borne inférieure, solution du sous-problème relaxé, est supérieure à la valeur de la meilleure solution entière trouvée (la meilleure solution entière que l'on puisse trouver sur la branche est dominée par la borne supérieure).

L'algorithme général du Branch and Bound peut être décrit par un processus itératif qui repose sur la définition d'une liste de nœuds prometteurs. Cette liste contient initialement le nœud racine de l'arbre. Le sous-problème associé à ce nœud correspond au problème initial à résoudre. À chaque itération, on commence par retirer un nœud de la liste. La sélection de ce nœud se fait selon la stratégie de branchement adoptée. On passe à l'évaluation du nœud sélectionné en résolvant la relaxation linéaire du sous-problème associé. Si le sous-problème relaxé est non réalisable alors on passe à l'itération suivante. Sinon si la solution du sous-problème relaxé est entière alors on met à jour la meilleure solution entière trouvée puis on passe à l'itération suivante. Sinon si la borne inférieure solution du sous-problème relaxé est dominée par la borne supérieure alors on passe à l'itération suivante. Sinon on branche au niveau du nœud courant en définissant une partition du domaine réalisable du sous-problème associé et en rajoutant les nœuds résultant de la séparation à la liste des nœuds prometteurs. On itère jusqu'à ce que la liste soit vide. Pour plus de détails, on réfère le lecteur au livre de Neapolitan et Naimipour (2010).

5.1.2 Génération de colonnes

La génération de colonnes est une méthode qui permet la résolution efficace des problèmes linéaires de grande taille. En effet, bon nombre de modèles linéaires repose sur la définition explicite d'un nombre élevé de variables de décision ou encore de colonnes. À l'optimum, la plupart de ces variables sont nulles. En conséquence, on peut se contenter d'un petit sous-ensemble de variables lors de la résolution du problème. Dans ce qui suit, nous décrivons brièvement les éléments clés de la génération de colonnes et nous référons à Desrosiers et Lübbecke (2005) pour plus de détails.

La génération de colonnes est une technique d'optimisation distribuée impliquant un problème principal appelé « *problème maître* » et un sous-problème appelé problème de « *pricing* ». Considérons le problème linéaire désigné par le problème maître :

$$(PM) \ z_{PM}^* = \min \sum_{j \in J} c_j \lambda_j \quad (5.1)$$

$$\text{s.c.} \ \sum_{j \in J} a_j \lambda_j \geq b \quad (5.2)$$

$$\lambda_j \geq 0, \forall j \in J \quad (5.3)$$

À chaque itération de l'algorithme du simplexe, on effectue une opération de « *pricing* ». Cette opération consiste à calculer les coûts réduits de toutes les variables hors base. Rappelons que l'on considère le cas d'un problème de minimisation. Ainsi, si le coût réduit minimal est strictement négatif alors on identifie la variable entrante, on effectue des opérations de pivot et on itère. L'algorithme s'arrête lorsqu'aucune variable hors base ne possède un coût réduit strictement négatif.

Contrairement à l'algorithme du simplexe qui effectue un pricing explicite, la génération de colonnes permet d'effectuer un pricing implicite. Un problème maître restreint à un ensemble de variables $(\lambda_j)_{j \in J'}$ avec $J' \subseteq J$ est considéré. Soient λ et Π les solutions optimales primale et duale de ce problème restreint. L'évaluation des coûts réduits se fait d'une manière implicite en résolvant le sous-problème suivant :

$$\bar{c}^* = \min\{c(a) - \Pi^t a \mid a \in \mathcal{A}\}, \quad (5.4)$$

avec \mathcal{A} l'ensemble des colonnes $\{a_j, j \in J\}$ et c une fonction qui permet de calculer le coût c_j d'une colonne donnée a_j . Si $\bar{c}^* \geq 0$ alors la solution optimale du problème maître restreint λ est optimale pour le problème global (PM). Sinon, on ajoute la colonne générée par le sous-problème (5.4) au problème maître restreint et on itère jusqu'à ce que $\bar{c}^* \geq 0$. Tout comme l'algorithme du simplexe, l'initialisation de la génération de colonnes se fait en introduisant une solution artificielle ou une solution heuristique. Il est à noter que la convergence de la génération de colonnes est assurée du moment que toute colonne entrante $a \in \mathcal{A}$ peut

être générée au plus une fois par le sous-problème. En effet, aucune variable existante dans le problème maître restreint ne peut avoir un coût réduit strictement négatif à l'optimum. Ainsi, le processus itératif converge après un nombre fini d'itérations lorsque l'ensemble \mathcal{A} est fini.

Un problème peut être modélisé de plusieurs façons équivalentes. Il est important de savoir que la modélisation est une étape déterminante dans le choix de la stratégie de résolution. Pour envisager d'utiliser la génération de colonnes, il faut que la formulation fasse intervenir un nombre élevé de variables de décision et un nombre petit de contraintes liantes sans lesquelles le problème peut être divisé en un ou plusieurs petits problèmes indépendants. La décomposition de Dantzig et Wolfe permet de faire apparaître une formulation ayant cette structure particulière se prêtant ainsi à une résolution par génération de colonnes.

Considérons le problème en nombres entiers à résoudre :

$$(PE) z_{PE}^* = \min \sum_{j \in J} c^j x \quad (5.5)$$

$$\text{s.c. } Ax \geq b \quad (5.6)$$

$$x \in \mathbb{X}, \quad (5.7)$$

avec $\mathbb{X} = \{x \in \mathbb{Z}_+^n \mid Dx \geq d\} \neq \emptyset$. D'après le théorème de Minkowski et de Weyl, chaque élément $x \in \mathbb{X}$ peut s'écrire comme la somme d'une combinaison convexe des points extrêmes $\{x_p\}_{p \in P}$ et une combinaison conique des rayons extrêmes $\{x_r\}_{r \in R}$ de l'enveloppe convexe de \mathbb{X} notée $\text{conv}(\mathbb{X})$.

$$x = \sum_{p \in P} x_p \lambda_p + \sum_{r \in R} x_r \lambda_r, \quad \sum_{p \in P} \lambda_p = 1, \lambda \in \mathbb{R}_+^{|P|+|R|} \quad (5.8)$$

En remplaçant x dans (5.5) et (5.6) par son expression (5.8), on obtient la formulation équivalente suivante :

$$z_{PE}^* = \min \sum_{p \in P} c_p \lambda_p + \sum_{r \in R} c_r \lambda_r \quad (5.9)$$

$$\text{s.c. } \sum_{p \in P} a_p \lambda_p + \sum_{r \in R} a_r \lambda_r \geq b \quad (5.10)$$

$$\sum_{p \in P} \lambda_p = 1 \quad (5.11)$$

$$\lambda \geq 0 \quad (5.12)$$

$$x = \sum_{p \in P} x_p \lambda_p + \sum_{r \in R} x_r \lambda_r \quad (5.13)$$

$$x \in \mathbb{Z}_+^n \quad (5.14)$$

Pour résoudre ce problème, on commence par relaxer les contraintes d'intégralité (5.14). Par conséquent, les contraintes (5.13) liant x et λ peuvent être supprimées. Le problème relaxé

obtenu constitue le problème maître. Ses colonnes sont définies par les points extrêmes et les rayons extrêmes de l'enveloppe convexe de \mathbb{X} . Initialement, un sous-ensemble de colonnes est considéré. Une fois le problème maître restreint résolu, on déduit la solution duale optimale. Soient Π le vecteur des multiplicateurs duaux optimaux associées aux contraintes (5.10) et Π_0 le multiplicateur dual optimal associé à la contrainte (5.11). Le sous-problème de pricing s'écrit donc :

$$\bar{c}^* = \min\{c_j - \Pi^t a_j - \Pi_0, c_{j'} - \Pi^t a_{j'} | j \in P, j' \in R\} \quad (5.15)$$

En utilisant la transformation linéaire (5.8), le sous-problème (5.15) peut être reformulé en un problème linéaire en nombres entiers énoncé comme suit :

$$\bar{c}^* = \min\{(c^t - \Pi^t A)x - \Pi_0 | x \in \mathbb{X}\} \quad (5.16)$$

Si $\bar{c}^* \geq 0$ alors il n'existe aucune colonne entrante et l'algorithme converge. Dans le cas contraire (c.-à-d. $\bar{c}^* < 0$), deux cas se présentent. Si \bar{c}^* est fini alors la colonne $[c^t x_p, (Ax_p)^1, 1]^t$ correspondante à un point extrême x_p de $\text{conv}(\mathbb{X})$ est introduite dans le problème maître restreint. Si par contre le sous-problème (5.16) est non borné inférieurement ($\bar{c}^* = -\infty$) alors la colonne $[c^t x_r, (Ax_r)^1, 0]^t$ correspondant à un rayon extrême x_r de $\text{conv}(\mathbb{X})$ est introduite dans le problème maître restreint. On itère jusqu'à ce que l'algorithme converge vers une solution optimale de la relaxation linéaire du problème (PE).

Pour résoudre le problème en nombres entiers (PE), la méthode de génération de colonnes est imbriquée dans une méthode de séparation et d'évaluation progressive. Des contraintes additionnelles sont introduites dans le problème maître restreint ou dans le sous-problème afin d'exprimer les décisions de branchement au niveau de l'arbre de recherche.

5.2 Génération de colonnes suivie d'un Branch and Bound

Pour le cas des problèmes linéaires en nombres entiers, la génération de colonnes est imbriquée dans une procédure de séparation et d'évaluation progressive. Cette méthode d'optimisation combinatoire est connue sous le nom de « *Branch and Price* ». À chaque nœud de l'arbre, la génération de colonnes permet de calculer la borne inférieure du sous-problème associé à ce nœud.

Une alternative souvent utilisée consiste à utiliser séquentiellement la génération de colonnes et le Branch and Bound. D'abord, la génération de colonnes est utilisée pour résoudre la relaxation linéaire du problème. Ensuite, le problème en nombres entiers restreint à l'ensemble des colonnes considérées par la génération de colonnes est résolu par le « *Branch and Bound* ». Contrairement au Branch and Price qui est une méthode exacte, cette procédure est une méthode heuristique. Ceci dit, les solutions sous-optimales obtenues demeurent de bonne qualité dans bon nombre d'applications.

Nous utilisons la génération de colonnes suivie d'un Branch and Bound au niveau du nœud racine. Dans ce qui suit, nous présentons les éléments clés de cette procédure en considérant notre problème de construction de quarts multi-jours dans un contexte multidépartement. Nous reprenons également certains résultats théoriques présentés dans la Section 5.1.2 afin d'illustrer le fonctionnement de la génération de colonnes.

Initialement, le problème maître restreint considère un sous-ensemble spécifique de variables de quart. Une fois le problème maître restreint résolu, nous passons à résoudre un sous-ensemble spécifique de sous-problèmes de pricing en cherchant des quarts ayant des coûts réduits strictement négatifs. Ces quarts sont des nouvelles colonnes à introduire dans le problème maître restreint. On itère jusqu'à ce que l'optimum de tous les sous-problèmes (de tous les employés et durant tous les jours travaillés) est positif. Nous obtenons ainsi la solution optimale de la relaxation linéaire du problème de construction de quarts.

Nous rétablissons l'intégralité des variables de quart considérées par la génération de colonnes. Nous obtenons un problème de construction de quarts restreint à l'ensemble des quarts introduits à chaque itération après le pricing. Ce problème restreint en nombres entiers est résolu par la procédure de séparation et d'évaluation progressive.

Plusieurs tests sont effectués afin de bien configurer notre procédure. Dans ce qui suit, nous listons les paramètres considérés. L'ensemble des configurations testées seront expliquées plus tard dans la partie expérimentale à la Section 6.3.

Solution initiale : Plusieurs techniques sont possibles pour initialiser la génération de colonnes. Lorsqu'il est difficile de connaître une solution de base réalisable pour le problème, l'introduction des variables artificielles permet d'outrepasser cette difficulté. Dans le cas contraire, une solution initiale peut être obtenue d'une manière heuristique en considérant par exemple un algorithme de construction glouton. Il est préférable de commencer le processus de la génération de colonnes à partir d'une solution de base réalisable. Il s'agit d'une stratégie d'accélération permettant d'éviter des valeurs duales élevées lors des premières itérations ce qui accélère la convergence de la méthode. Dans les deux cas que nous testons, les colonnes initiales contiennent une solution de base réalisable formée exclusivement par des quarts internes (c.-à-d. sans transfert).

- Premier cas désigné par QI : Les colonnes initiales sont formées par toutes les variables de quart internes admissibles $(\cup_{e \in E} \cup_{j \in J_e} \cup_{t \in \bar{T}} \bar{Q}_{e,j,t}^{int})$.
- Deuxième cas désigné par SI : Nous résolvons le problème maître restreint aux quarts internes. Une fois ce problème résolu, uniquement les colonnes correspondant à des variables de quart sélectionnées (c'est à dire ayant des valeurs non nulles) sont retenues comme colonnes initiales.

Sélection des sous-problèmes : Pour chaque employé $e \in E$ et durant chaque jour travaillé $j \in J_e$ de cet employé, nous définissons le sous-problème de pricing suivant :

$$\begin{aligned}
\bar{c}^* = \{ & \min_{t \in \bar{T}, q \in \bar{Q}_{e,j,t}} (\gamma^l c_q^l + \gamma^{tr} c_q^{tr} - \sum_{i \in I} \Pi_{d,i}^{dem} \delta_{q,d,i} \\
& - s_q \delta_{e,j-1,j} \Pi_{e,j-1,j}^{sep} + (s_q + l_q) \delta_{e,j,j+1} \Pi_{e,j,j+1}^{sep} - l_q \Pi_e^l \\
& - ((l_q - \sum_{d \in D_e} l_{q,d}^{tr}) - T_{min} l_q) \Pi_e^t) X_q - \Pi_{e,j}^{jour}, X_q \in \{0, 1\} \},
\end{aligned} \tag{5.17}$$

avec $\Pi_{d,i}^{dem}$, $\Pi_{e,j}^{jour}$, $\Pi_{e,j,j+1}^{sep}$, Π_e^l et Π_e^t les multiplicateurs duaux optimaux associés aux contraintes (4.7) - (4.11) respectivement. Le paramètre $\delta_{e,j,j+1}$ est égal à 1 si les jours consécutifs j et $j+1$ sont des jours travaillés de l'employé e et 0 sinon.

Il existe plusieurs manières d'approcher les sous-problèmes de pricing. Une pratique courante consiste à définir des problèmes de plus court chemin avec contraintes de ressources et de les résoudre par programmation dynamique. Le défi serait dans ce cas de gérer la complexité de ces problèmes qui augmente avec le nombre et le type de ressources (Bouchard, 2005). Dans cette thèse, les sous-problèmes de pricing considèrent explicitement tous les quarts qui ne sont pas encore considérés par le problème maître restreint. La résolution de chaque sous-problème (5.17) est lancée sur un « *thread* » et tous les threads sont lancés en parallèle. Chaque thread considère toutes les variables de quarts hors base d'un employé durant un jour travaillé. Selon la stratégie de parcours des variables hors base expliquée ultérieurement, le thread va parcourir un ensemble ou toutes les variables hors base et va calculer les coûts réduits associés à ces variables. Les expérimentations montrent que le temps de parcours des sous-problèmes reste petit même pour les grandes instances.

La définition des sous-problèmes que nous considérons confère à notre procédure une grande flexibilité quant aux contraintes d'admissibilité des quarts. D'abord, les sous-problèmes sont triés selon un ordre croissant de leur jour d'occurrence et placés dans une file d'attente. Les sous-problèmes qui correspondent au même jour et diffèrent uniquement par l'identité des employés, sont triés selon un ordre arbitraire. À chaque itération de la génération de colonnes, nous parcourons la file en cherchant au plus un nombre fixe de sous-problèmes tels que $\bar{c}^* < 0$ (c.-à-d. ayant au moins un quart dont le coût réduit est strictement négatif). À chaque fois, le sous-problème en tête de la file est retiré, testé puis ajouté en fin de la file. Le nombre de sous-problèmes à résoudre est calculé comme un pourcentage du nombre total de tous les sous-problèmes. Nous allons tester deux valeurs de ce paramètre : $\%SP \in \{25\%, 100\%\}$.

Nombre de colonnes : Pour chaque sous-problème parcouru, on cherche au maximum un nombre fixe de quarts ayant un coût réduit strictement négatif. Nous allons tester deux valeurs de ce paramètre : $NMAX \in \{1, 10\}$.

Sélection des colonnes : Pour chaque sous-problème sélectionné, on cherche les quarts à introduire au problème maître selon deux critères différents :

- FT : On se contente des NMAX premières colonnes rencontrées ayant un coût réduit strictement négatif.
- BT : On cherche les NMAX meilleures colonnes non considérées par le problème maître restreint (c.-à-d. les NMAX quarts ayant les coûts réduits strictement négatifs les moins élevés).

Dans le premier cas, le temps de parcours d'un sous-problème est moins élevé étant donné que l'on n'est pas obligé de faire un pricing total de l'ensemble des quarts non considérés dans le problème maître restreint. Dans le deuxième cas, on retient à chaque fois les NMAX variables ayant les coûts réduits les moins élevés et donc entraînant potentiellement l'amélioration la plus importante de la valeur de l'objectif pour une itération donnée. Ceci dit, ce critère peut augmenter le temps de parcours d'un sous-problème étant donné que l'on doit calculer le coût réduit de tous les quarts non considérés dans le problème maître restreint. Il s'agit donc d'identifier le critère offrant le meilleur compromis entre le temps de parcours d'un sous-problème et la convergence globale de la méthode.

5.3 Analyse de performance de la procédure de génération de colonnes

Cette partie est une analyse de la procédure basée sur la génération de colonnes. Cette procédure sera utilisée ultérieurement pour la résolution de certains problèmes en nombres entiers définis dans les heuristiques de décomposition. En conséquence, nous commençons par la configuration de cette procédure afin de pouvoir l'utiliser plus tard.

5.4 Procédure basée sur la génération de colonnes

La procédure basée sur la génération de colonnes implique un certain nombre de paramètres. Par conséquent, une configuration de cette procédure est requise avant de l'utiliser dans l'heuristique de prétransfert. Une fois la procédure configurée, elle est comparée à la procédure de séparation et d'évaluation progressive de CPLEX.

Analyse des paramètres de la procédure basée sur la génération de colonnes

Nous avons testé cinq combinaisons décrites dans le Tableau 5.1. Chaque combinaison est caractérisée par quatre paramètres listés précédemment dans la Section 5.2 qui sont : les colonnes initiales, %SP, la méthode de sélection des nouvelles colonnes et NMAX. La première combinaison initialise la génération de colonnes avec une solution de base réalisable formée exclusivement par des quarts internes. Un sous-problème est considéré pour chaque employé et durant chaque jour de travail de cet employé. Les sous-problèmes sont triés par jour. Ils sont donc parcourus dans cet ordre. À chaque itération, nous identifions au maximum

25% de sous-problèmes ayant au moins une colonne avec un coût réduit strictement négatif. Chaque sous-problème est traité comme suit. Nous parcourons l'ensemble des quarts internes et externes hors base qui ne sont pas encore introduits dans le problème maître restreint. Ce parcours arrête au premier quart ayant un coût réduit strictement négatif. Dans la deuxième combinaison, nous changeons uniquement la méthode d'initialisation de la génération de colonnes. Dans ce cas, tous les quarts internes sont initialement dans le problème maître restreint. Par conséquent, l'ajout des nouvelles colonnes porte uniquement sur les quarts externes. La troisième combinaison diffère de la deuxième uniquement par le nombre maximal des nouvelles colonnes à rajouter par sous-problème. Au lieu de se contenter d'une seule colonne, nous identifions les 10 premiers quarts externes ayant un coût strictement négatif. La combinaison 4 change la méthode d'ajout des nouveaux quarts externes à introduire dans le problème maître restreint. Au lieu de prendre au maximum les 10 premières colonnes rencontrées ayant un coût strictement négatif, nous identifions un maximum de 10 colonnes ayant les coûts réduits strictement négatifs les moins élevés. Finalement, la combinaison 5 ne considère pas un pourcentage maximal de sous-problèmes ayant au moins un quart de coût réduit strictement négatif comme le font les autres combinaisons. Au contraire, elle parcourt tous les sous-problèmes à chaque itération de la génération de colonnes. Dans cette partie, chaque combinaison sera désignée par la valeur de ses quatre paramètres. Par exemple, la combinaison 1 s'écrit (SI,25,FT,1).

Tableau 5.1 – Combinaisons de la procédure basée sur la génération de colonnes

Combinaison	Colonnes initiales	Sélection des sous-problèmes %SP (%)	Sélection des colonnes	# Colonnes NMAX
1	SI	25	FT	1
2	QI	25	FT	1
3	QI	25	FT	10
4	QI	25	BT	10
5	QI	100	BT	10

Pour les cinq combinaisons, nous avons utilisé CPLEX pour implémenter la procédure de génération de colonnes suivie d'un Branch and Bound. Le solveur sert à résoudre deux problèmes : 1) le problème maître restreint et 2) le problème en nombres entiers obtenu au niveau du nœud racine restreint aux quarts sélectionnés par la génération de colonnes. Le temps total maximal de la procédure est fixé à 1 heure. De plus, nous avons fixé l'écart d'optimalité de CPLEX à 1%. Le jeu de données correspond aux 24 instances issues du JEU2 (voir 4.2).

Soit Z la valeur de l'objectif correspondant à la meilleure solution trouvée pour chaque instance et chaque combinaison. Soit $\min Z$ la meilleure valeur de l'objectif trouvée par l'ensemble des combinaisons pour une instance donnée. Pour comparer les cinq combinaisons selon la qualité des solutions trouvées, notons par « *Déviatio*n » l'écart en pourcentage entre la valeur de l'objectif Z de chaque combinaison et la meilleure valeur de l'objectif $\min Z$

Tableau 5.2 – Qualité relative des solutions obtenues par la procédure de génération de colonnes suivie d'un Branch and Bound

Instance	(SI,25,FT,1)	(QI,25,FT,1)	(QI,25,FT,10)	(QI,25,BT,10)	(QI,100,BT,10)
D2E20T2P1	344.5	251.5	251.5	251.5	251.5
D2E20T2P2	429.5	355.8	355.8	355.8	355.8
D2E20T2P3	412.1	318.5	318.5	328.2	328.2
D2E20T2P4	392.6	337.0	319.0	319.0	319.0
D3E20T2P1	317.7	262.1	262.1	262.2	262.2
D3E20T2P2	663.7	514.6	505.0	504.9	505.0
D3E20T2P3	458.2	402.4	402.4	411.2	411.2
D3E20T2P4	683.8	513.9	513.9	513.9	513.9
D5E20T2P1	395.2	345.4	345.4	345.4	345.4
D5E20T2P2	523.1	466.8	466.4	466.5	466.5
D5E20T2P3	479.7	450.5	450.5	450.5	450.7
D5E20T2P4	836.3	770.1	770.1	770.1	770.1
D5E50T2P1	670.3	538.4	538.4	538.5	538.4
D5E50T2P2	768.4	763.0	763.6	763.2	753.4
D5E50T2P3	564.1	536.4	536.5	535.2	536.9
D5E50T2P4	830.3	808.8	808.6	808.7	808.2
D5E70T2P1	663.2	659.9	659.9	659.9	660.0
D5E70T2P2	1,101.1	1,005.6	1,006.6	996.4	1,006.3
D5E70T2P3	947.0	739.5	767.7	739.7	903.9
D5E70T2P4	816.5	719.9	719.7	721.5	719.3
D5E200T2P1	1,495.0	1,414.4	1,414.3	1,414.3	1,414.3
D5E200T2P2	2,613.7	1,788.2	1,771.0	1,769.7	1,769.8
D5E200T2P3	2,320.7	2,355.7	2,131.3	2,113.4	2,215.7
D5E200T2P4	2,387.5	1,914.4	2,008.5	2,026.3	1,888.9
%Ins. : Déviation \leq 1%	4.17	75.00	83.33	83.33	83.33

(Déviation = $\frac{Z - \min Z}{\min Z}$). Le Tableau 5.2 donne la valeur de l'objectif pour toutes les instances et toutes les combinaisons. Chaque colonne du tableau correspond à l'une des cinq combinaisons. Les deux dernières lignes du tableau correspondent au pourcentage des instances telles que la Déviation soit \leq à 1%. La valeur de l'objectif Z est affichée en gras lorsque la Déviation est \leq à 1%.

Les tests montrent que la première combinaison (SI,25,FT,1) affiche la pire performance avec une déviation supérieure à 1% dans 95.83% des cas. Cette performance s'explique par les colonnes initiales considérées par le problème maître restreint. Alors que la première combinaison considère uniquement une solution de base réalisable formée par des quarts internes, le reste des quatre combinaisons initialisent la génération de colonnes par tous les quarts internes. Notons que le nombre de variables de décision moyen du problème en nombres

entiers est égal à 4,671 lorsque l’option SI est utilisée alors qu’il varie entre 10,325 et 12,165 lorsque l’option QI est choisie. La solution est de qualité moindre à cause du nombre limité de quarts présents dans le problème maître restreint à la fin de la génération de colonnes. Étant donné que l’on ne procède pas par un Branch and Price et que l’on se limite aux colonnes rajoutées au nœud racine, l’ajout de tous les quarts internes donne lieu à des solutions de qualité meilleure. Remarquons que la qualité de la solution est relativement la même pour les 4 combinaisons utilisant l’option QI. Nous passons donc à étudier la performance en se basant sur le critère du temps de résolution.

Le Tableau 5.3 montre les temps de calcul en secondes de la procédure de génération de colonnes suivie par un Branch and Bound pour les 24 instances résolues par les cinq combinaisons. Soit CPU le temps total requis pour résoudre une instance en utilisant une combinaison. De plus, soit min CPU le meilleur temps requis pour résoudre une instance donnée avec une déviation \leq à 1%. Les cellules en gris correspondent aux instances résolues avec une déviation \leq à 1%. Les temps de calcul sont en gras pour la ou les combinaisons ayant trouvé une solution avec une déviation \leq 1 % dans un temps minimal égal à min CPU. La deuxième partie du tableau donne pour chaque combinaison le pourcentage d’instances résolues avec une déviation \leq à 1% et dont l’écart de temps $\Delta\text{CPU} = \text{CPU} - \text{min CPU}$ appartient à un intervalle de temps donné. Clairement, les deux combinaisons qui affichent les meilleures performances sont (QI,25,FT,10) et (QI,100,BT,10).

Notons que l’effet des paramètres de configuration est plus visible lorsque nous analysons, d’une part les temps de calcul requis pour résoudre la relaxation linéaire du problème, et d’autre part la taille des modèles du problème en nombres entiers obtenu au niveau du nœud racine. Pour chacun des quatre graphiques illustrés par la Figure 5.1, l’axe des abscisses correspond aux 24 instances du JEU2. L’axe des ordonnées représente respectivement le nombre de variables, le nombre de coefficients non nuls, le temps requis pour résoudre la relaxation linéaire du problème en utilisant la génération de colonnes et le temps requis pour résoudre le problème en nombres entiers obtenu au niveau du nœud racine en utilisant le Branch and Bound.

L’initialisation de la génération de colonnes par tous les quarts internes (QI) induit un problème en nombres entiers dont le nombre de variables de décision et le nombre de coefficients non nuls est nettement supérieur à celui obtenu en initialisant par une solution de base réalisable (SI). De même, lorsque nous ajoutons à chaque itération de la génération de colonnes plusieurs nouveaux quarts par sous-problème à la fois (NMAX= 10) au lieu de nous contenter d’une seule (NMAX= 1), la taille du problème en nombres entiers augmente. Concentrons nous sur les combinaisons (QI,25,FT,10) et (QI,25,BT,10). Rappelons que ces deux combinaisons diffèrent par la méthode de parcours des colonnes hors base non considérées par le problème maître restreint. La taille du problème en nombres entiers diminue lorsque la génération de colonnes recherche pour chaque sous-problème les 10 meilleurs

Tableau 5.3 – Temps de calcul en secondes de la procédure de génération de colonnes suivie d'un Branch and Bound

Instance	(SI,25,FT,1) (QI,25,FT,1) (QI,25,FT,10) (QI,25,BT,10) (QI,100,BT,10)				
	CPU				
D2E20T2P1	5	3	3	2	2
D2E20T2P2	5	7	7	19	27
D2E20T2P3	29	121	11	191	191
D2E20T2P4	8	18	22	20	20
D3E20T2P1	11	8	2	4	2
D3E20T2P2	4	5	3	3	2
D3E20T2P3	18	14	14	9	4
D3E20T2P4	3	6	10	7	7
D5E20T2P1	14	4	3	2	2
D5E20T2P2	6	16	8	6	6
D5E20T2P3	22	6	5	6	19
D5E20T2P4	7	5	4	12	8
D5E50T2P1	3,600	51	49	15	21
D5E50T2P2	393	71	27	78	74
D5E50T2P3	3,600	2,650	526	2,302	655
D5E50T2P4	1,258	210	175	136	100
D5E70T2P1	3,252	72	140	88	65
D5E70T2P2	867	181	175	570	196
D5E70T2P3	3,600	1,251	3,600	427	3,600
D5E70T2P4	3,600	627	296	3,181	466
D5E200T2P1	3,600	391	1,720	1,024	1,368
D5E200T2P2	3,600	502	630	706	471
D5E200T2P3	3,600	3,600	3,600	3,600	3,600
D5E200T2P4	3,600	3,600	3,600	3,600	1,903
Intervalle	%Ins. : Déviation $\leq 1\%$ et Δ CPU \in Intervalle				
[0, 0]	0.00	20.83	37.50	29.17	45.83
]0,300]	0.00	41.67	41.67	37.50	33.33
]300,600]	0.00	4.17	0.00	4.17	0.00
]600,1800]	0.00	4.17	4.17	8.33	4.17
]1800,3600]	4.17	4.17	0.00	4.17	0.00

quarts à introduire dans le problème maître restreint (BT) au lieu de se contenter des 10 premiers rencontrés. Finalement, lorsque nous passons de la combinaison (QI,25,BT,10) à la combinaison (QI,100,BT,10), nous augmentons légèrement la taille du problème. Rajouter les colonnes par blocs de sous-problèmes (%SP= 25%) permet donc d'obtenir un problème de taille plus petite que celui obtenu par un ajout de colonnes pour tous les sous-problèmes à la fois (%SP= 100%).

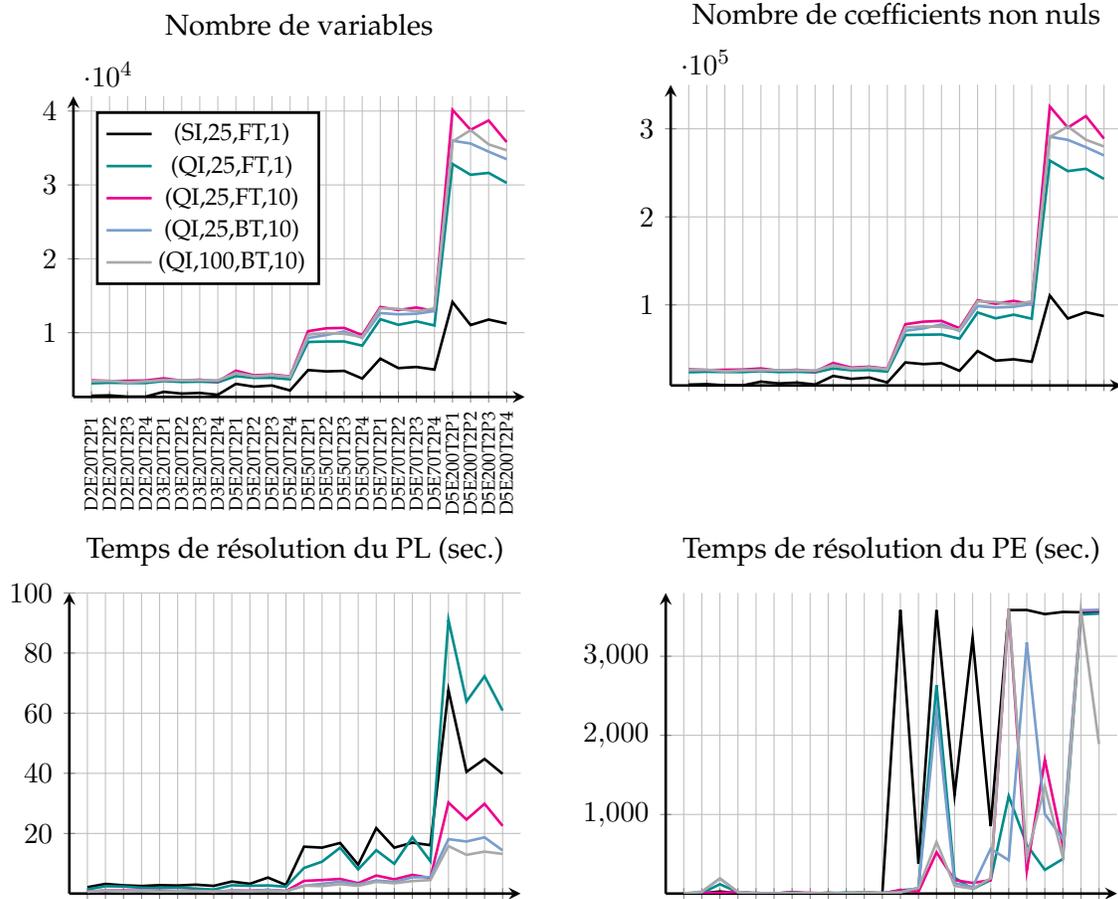


Figure 5.1 – Complexité des calculs de la procédure basée sur la génération de colonnes

Passons maintenant au temps requis pour résoudre la relaxation linéaire du problème en utilisant la génération de colonnes. Les temps de calcul varient entre 1 seconde et 91 secondes. Les combinaisons (QI,25,FT,1) et (SI,25,FT,1) induisent les temps les plus élevés. L'ajout de plusieurs colonnes à la fois par itération et par sous-problème permet donc d'accélérer la génération de colonnes. Les trois combinaisons restantes peuvent être triées de la plus lente à la plus rapide comme suit : (QI,25,FT,10), (QI,25,BT,10) et (QI,100,BT,10). Deux remarques se dégagent : la recherche des colonnes ayant les coûts réduits strictement négatifs les moins élevés et le parcours de tous les sous-problèmes à chaque itération de la génération de colonnes permettent d'accélérer la convergence.

Le temps requis pour résoudre le problème en nombres entiers en utilisant le Branch and Bound de CPLEX constitue la proportion la plus élevée du temps total de la procédure basée sur la génération de colonnes. Il varie entre 1 seconde et 3596 secondes. Singulièrement, la combinaison (SI,25,FT,1) induit des problèmes en nombres entiers de taille moindre mais

affiche les temps de résolution les plus élevés. Pour le reste des combinaisons, il n'y a pas de dominance nette d'aucune d'entre elles.

En se basant sur le Tableau 5.3, il est clair que la combinaison (QI,100,BT,10) affiche les meilleures performances. Cette combinaison sera retenue à chaque fois que la procédure basée sur la génération de colonnes sera utilisée.

Comparaison entre la procédure basée sur la génération de colonnes et le Branch and Bound de CPLEX

Dans cette partie, notre procédure de génération de colonnes suivie par un Branch and Bound est notée GenColBB alors que la procédure de Branch and Bound de CPLEX est désignée par BB. Reprenons les résultats montrés dans la Section 4.6.3 de la formulation MCQ pour les instances du JEU2. Rappelons que nous avons utilisé CPLEX pour résoudre cette formulation avec un Gap d'optimalité limite égal à 1% et un temps limite égal à 1 heure. Notre procédure GenColBB considère également la formulation MCQ du problème. Elle commence par résoudre la relaxation linéaire du problème en utilisant la génération de colonnes. Ensuite, elle rétablit l'intégralité de toutes les variables de quarts du problème maître restreint obtenu à la dernière itération de la génération de colonnes. Elle résout après ce problème en nombres entiers en utilisant CPLEX. Rappelons que notre procédure GenColBB est utilisée avec un temps limite égal à 1 heure et avec un gap d'optimalité limite de CPLEX égal à 1%.

Tableau 5.4 – Taille des problèmes en nombres entiers résolus par GenColBB et par BB

Classes	# Variables			# Contraintes		# Coefficients $\neq 0$		
	BB	GenColBB	Écart (%)	BB	GenColBB	BB	GenColBB	Écart (%)
D2E20T2	7,436	3,398	54.31	373	373	59,579	25,603	57.03
D3E20T2	9,448	3,545	62.48	457	457	75,387	25,553	66.10
D5E20T2	17,022	4,206	75.29	625	625	136,439	28,653	79.00
D5E50T2	42,596	9,737	77.14	932	932	350,775	74,211	78.84
D5E70T2	57,353	13,168	77.04	1,138	1,138	476,426	102,986	78.38
D5E200T2	159,481	35,872	77.51	2,472	2,472	1,331,968	290,216	78.21

Le Tableau 5.4 montre la taille du problème en nombres entiers résolu par les deux procédures BB et GenColBB, c'est à dire sans et avec la génération de colonnes. Les lignes correspondent aux classes du JEU2. Chaque classe est formée par quatre instances qui diffèrent uniquement par la variabilité de la demande en effectifs. La première partie du tableau donne le nombre de variables lorsque la génération de colonnes n'est pas utilisée, le nombre moyen de variables lorsque la génération de colonnes est utilisée et l'écart moyen en % entre le nombre de variables sans et avec la génération de colonnes (Écart = $(\#Variables\ BB - \#Variables\ GenColBB) / \#Variables\ BB$). La deuxième partie du tableau donne le nombre de

contraintes qui reste le même pour les deux procédures. De même pour la troisième partie qui montre le nombre de coefficients non nuls sans génération de colonnes, le nombre moyen avec génération de colonnes et l'écart moyen. Les cellules en gras correspondent au nombre le moins élevé de variables et de coefficients non nuls pour chaque instance. Clairement, l'utilisation de la génération de colonnes comme un prétraitement de sélection des quarts permet de diminuer significativement le nombre de variables de décision avec un écart moyen allant jusqu'à 77.51% et le nombre de coefficients non nuls avec un écart moyen allant jusqu'à 79%.

Tableau 5.5 – Performances de la procédure basée sur la génération de colonnes et la procédure de séparation et d'évaluation progressive

Instances	CPU (secondes)		Z			Gap d'optimalité(%)	
	BB	GenColBB	BB	GenColBB	Écart (%)	BB	GenColBB
D2E20T2P1	2.0	1.7	251.5	251.5	0.00	0.00	0.00
D2E20T2P2	10.9	26.8	355.8	355.8	0.00	0.00	0.96
D2E20T2P3	29.3	191.4	318.5	328.2	3.05	0.00	0.00
D2E20T2P4	12.5	20.0	319.0	319.0	0.00	0.00	0.00
D3E20T2P1	6.4	2.2	262.2	262.2	0.00	0.06	0.06
D3E20T2P2	2.8	1.7	505.0	505.0	0.00	0.02	0.02
D3E20T2P3	9.3	3.8	402.4	411.2	2.19	0.00	0.00
D3E20T2P4	7.5	6.9	514.0	513.9	-0.02	0.06	0.02
D5E20T2P1	7.3	2.0	345.5	345.4	-0.03	0.03	0.00
D5E20T2P2	57.8	6.2	457.9	466.5	1.88	0.00	0.99
D5E20T2P3	21.2	19.3	450.5	450.7	0.04	0.76	0.04
D5E20T2P4	9.1	7.6	770.1	770.1	0.00	0.00	0.00
D5E50T2P1	215.2	20.8	540.6	538.4	-0.41	0.41	0.00
D5E50T2P2	158.2	74.3	753.7	753.4	-0.04	0.14	0.04
D5E50T2P3	3,600.2	655.4	572.5	536.9	-6.22	7.90	0.98
D5E50T2P4	2,229.9	100.0	809.6	808.2	-0.17	0.20	0.00
D5E70T2P1	431.2	64.8	659.9	660.0	0.02	0.02	0.03
D5E70T2P2	1,350.6	196.2	996.7	1,006.3	0.96	0.03	0.04
D5E70T2P3	3,600.4	3,600.1	1,019.1	903.9	-11.30	27.47	18.20
D5E70T2P4	3,600.3	465.6	969.3	719.3	-25.79	26.16	0.00
D5E200T2P1	3,601.0	1,367.8	1,497.4	1,414.3	-5.55	5.57	0.01
D5E200T2P2	3,600.7	470.7	1,919.7	1,769.8	-7.81	8.80	0.51
D5E200T2P3	3,600.6	3,600.3	2,365.7	2,215.7	-6.34	12.80	6.88
D5E200T2P4	3,600.5	1,902.8	2,175.6	1,888.9	-13.18	14.11	0.97

Le Tableau 5.5 est une synthèse de la performance de notre procédure basée sur la génération de colonnes par rapport à celle de la procédure de séparation et d'évaluation progressive de CPLEX. Les lignes du tableau correspondent aux 24 instances du JEU2. La première partie du tableau représente le temps total de résolution du problème en utilisant la procédure BB d'une part et notre procédure GenColBB d'autre part. La deuxième partie du tableau repré-

sente la valeur de l'objectif de la meilleure solution trouvée par la procédure BB et par notre procédure GenColBB ainsi que le pourcentage d'écart entre les deux. La troisième partie montre le gap d'optimalité obtenu par la résolution du problème en nombres entiers sans et avec la génération de colonnes. Pour chaque instance, le temps de résolution, la valeur de l'objectif et le gap d'optimalité les moins élevés sont en gras. De plus, le pourcentage d'écart qui montre la qualité relative de la solution trouvée par la GenColBB par rapport à celle du BB est en gras lorsqu'elle est $> 1\%$ en valeur absolue. Cet écart relatif est calculé comme suit : $\frac{Z_{GenColBB} - Z_{BB}}{Z_{BB}}$ avec $Z_{GenColBB}$ et Z_{BB} la valeur de l'objectif des meilleures solutions obtenues par la GenColBB et le BB respectivement.

Commençons par comparer les deux procédures en se basant sur la qualité relative des solutions obtenues. Cette qualité est mesurée par l'écart relatif. Dans 58% des cas, la valeur absolue de l'écart relatif reste $\leq 1\%$ et donc la qualité des solutions obtenues par la GenColBB et le BB est relativement proche. Uniquement pour trois instances (13% des cas), le BB donne une solution de qualité meilleure avec un écart relatif moyen égal à 2.37%. Au contraire, la GenColBB trouve une solution de meilleure qualité dans 29% des cas. De plus, cette qualité est nettement meilleure avec un écart relatif moyen en valeur absolue égal à 10.88% variant entre 5.55% et 25.79%. Cet écart est plus élevé pour les plus grandes instances.

Pour ce qui est de la qualité absolue des solutions trouvées, le BB trouve dans 71% des cas (17 instances) une solution qui est prouvée à être à 1% ou moins de l'optimum par CPLEX. Pour 13 instances parmi ces 17, la valeur absolue de l'écart relatif reste $\leq 1\%$ ce qui signifie que la solution trouvée par la GenColBB est très proche de l'optimum. Pour les 3 restantes, la valeur absolue de l'écart relatif atteint au maximum 3.05%. Ainsi, la génération de colonnes réussit dans la plupart des cas à identifier les bons quarts externes qu'il faut introduire au problème maître restreint. Par conséquent, la restriction n'altère pas la qualité des solutions trouvées par notre procédure heuristique.

Lorsque la GenColBB et le BB trouvent des solutions de qualité proche (c.-à-d. que l'écart relatif est $\leq 1\%$), notre procédure basée sur la génération de colonnes est la plus rapide dans la majorité des cas. Rappelons que l'écart relatif est $\leq 1\%$ pour 14 instances parmi les 24 formant le JEU2. Pour 12 instances parmi ces 14, le temps requis par la GenColBB est inférieur à celui du BB en moyenne de 328.7 secondes. Cet écart est significatif et atteint les 2,129.9 secondes pour l'instance D5E70T2P1. Au contraire, l'écart est nettement moins important lorsque le BB devient plus rapide que la GenColBB. Il est égal à 7.5 et 15.9 secondes pour les deux instances restantes parmi les 14. Rappelons également que la qualité de la solution trouvée par le BB est 3% meilleure que celle de la GenColBB uniquement pour trois instances. Dans ce cas, l'écart moyen de temps entre les deux procédures est égal à 28.6 secondes pour les deux instances telles que le BB est le plus rapide alors qu'il est égal à 162.1 secondes pour l'instance restante lorsque la GenColBB est la plus rapide. Finalement,

la solution trouvée par notre procédure est de meilleure qualité que celle du BB dans le cas des sept instances restantes du JEU2. Pour deux instances parmi ces sept, l'écart de temps est négligeable. Pour les cinq autres instances dont la taille est la plus élevée de tout le JEU2, notre procédure est de loin plus rapide avec un écart de temps moyen égal à 2,628.1 secondes. Cet écart est significatif et il varie entre 1,697.7 et 3,134.7 secondes. Ainsi, l'utilisation de la génération de colonnes permet de réduire significativement les temps de calcul.

En conclusion, notre procédure basée sur la génération de colonnes montre de bonnes performances par rapport à la procédure de séparation et d'évaluation progressive. En effet, elle permet de calculer des solutions de meilleure qualité en un temps plus court pour la plupart des cas, notamment lorsque la taille du problème augmente.

Chapitre 6

Approche de résolution prétransfert

Dans ce chapitre, nous proposons une heuristique de résolution du problème de construction de quarts multi-jours dans un contexte multidépartement. La particularité de notre heuristique est de combiner la construction des quarts et l'attribution des transferts en commençant d'abord par l'identification d'un ensemble restreint de transferts avant de passer à construire les quarts personnalisés. Le problème est scindé en deux étapes résolues séquentiellement :

- Dans la première étape, nous définissons un problème agrégé qui est une approximation du problème original. Cette approximation permet de faire ressortir un problème plus facile à résoudre. Le problème agrégé sera formulé à l'aide du modèle de construction de quarts MCQ et résolu en utilisant une heuristique basée sur une procédure de génération de colonnes. L'extrait de cette étape correspond aux transferts agrégés identifiés lors de la résolution du modèle agrégé.
- Dans la deuxième étape, nous définissons un problème désagrégé qui est une restriction du problème original. Cette restriction permet d'intensifier la recherche dans le domaine réalisable en exploitant l'information provenant des transferts agrégés identifiés précédemment. Différentes procédures de résolution du problème restreint sont proposées et évaluées.

6.1 Problème agrégé

Dans cette section, nous décrivons l'étape d'agrégation. Cette étape vise à réduire la taille du problème étudié. Au lieu de considérer des données temporelles précises aux 15 minutes, nous considérons des données temporelles approximatives aux deux heures. Par conséquent, la question qui se pose dans cette étape est de définir la méthode permettant d'effectuer la conversion des données temporelles du problème. Ces intrants permettent donc de définir le problème agrégé perçu comme une approximation du problème original. Premièrement, nous proposons plusieurs méthodes d'agrégation que l'on va utiliser pour définir les intrants du problème agrégé. Ces intrants sont déduits à partir des données temporelles du problème

initial. Deuxièmement, nous passons à comparer les méthodes proposées en résolvant pour chacune de ces méthodes un modèle de construction de quarts multi-jours agrégés. L'analyse comparative se fait au biais de l'évaluation des qualités des solutions agrégées obtenues et des temps de calcul requis.

6.1.1 Méthodes d'agrégation des données

L'étape d'agrégation vise à diminuer la complexité du problème. Étant donné que la période de discrétisation est un facteur déterminant du niveau de difficulté du problème, nous proposons d'introduire un problème approximatif obtenu au biais de l'agrégation temporelle des données. Il s'agit de convertir ces données en les exprimant en fonction d'une période de discrétisation plus grossière. D'une part, cette conversion permet de réduire la taille du problème à résoudre. D'autre part, elle provoque un écart entre les données initiales et celles agrégées. Le défi est de trouver le meilleur compromis entre la réduction de la complexité du problème et la réduction de la perte de précision des données.

Notons le problème initial que l'on vise à résoudre par P et le problème agrégé que l'on vise à définir par \bar{P} . Le problème \bar{P} doit être, d'une part, de taille plus petite que celle de P et, d'autre part, capable de reporter le plus fidèlement possible les données temporelles de P . Nous distinguons deux types de données : les courbes de demande et les réglementations. Pour l'agrégation des courbes de demande, nous proposons trois méthodes : l'agrégation basée sur la moyenne, l'agrégation basée sur l'écart absolu et l'agrégation basée sur la médiane. Pour l'agrégation des réglementations, nous proposons deux méthodes : l'agrégation à couverture entière et l'agrégation à couverture continue.

Agrégation basée sur la moyenne

Rappelons que pour le problème initial P , l'horizon de planification est divisé en des périodes de durées égales à 15 minutes. Ces périodes, notées i , sont classées par ordre croissant en fonction de leurs heures de début. Pour définir le problème agrégé \bar{P} , nous divisons l'horizon de planification en 84 périodes de durées égales à 2 heures. Ces périodes agrégées, notées \bar{i} , sont classées également par ordre croissant en fonction de leurs heures de début. Soit \bar{I} l'ensemble de toutes les périodes agrégées de l'horizon de planification. Chacune de ces périodes est divisée à son tour en un ensemble de huit périodes de durées égales à 15 minutes. Nous verrons comment une donnée temporelle relative à une période agrégée sera déduite à partir des données temporelles du problème original relatives aux huit périodes désagrégées correspondantes.

Considérons la courbe de demande du département d correspondant au problème initial P . Cette courbe spécifie le nombre d'employés requis dans le département d durant chaque période désagrégée i . Rappelons que cet intrant est noté $r_{d,i}$. L'objectif est de définir la courbe de demande agrégée associée au département d qui donne le nombre requis d'employés

durant chaque période agrégée \bar{i} que l'on notera $r_{d,\bar{i}}$. Soit f l'application qui associe à chaque période agrégée \bar{i} l'ensemble des 8 périodes consécutives désagrégées correspondantes $f(\bar{i})$. On a $f(\bar{i}) = \{8(\bar{i} - 1) + 1, 8(\bar{i} - 1) + 2, \dots, 8\bar{i}\}$. La question est de trouver, pour chaque période agrégée \bar{i} , la valeur $r_{d,\bar{i}}$ la plus représentative de la série de valeurs $(r_{d,i})_{i \in f(\bar{i})}$.

Considérons un exemple illustratif. Prenons la période agrégée correspondant à l'intervalle de temps [2h00,4h00]. De plus, supposons qu'on a besoin de quatre employés durant l'intervalle de temps [2h00,3h30] et de six employés durant l'intervalle de temps [3h30,4h00]. La question est de trouver la valeur la plus représentative pour agréger la série de données (4, 4, 4, 4, 4, 4, 6, 6). Nous avons donc besoin de définir une règle qui permet de choisir cette valeur approximative.

Pour la méthode basée sur la moyenne, la demande d'une période agrégée est calculée à partir de la moyenne arithmétique des demandes désagrégées correspondantes. L'écart d'agrégation durant une période désagrégée $i \in f(\bar{i})$ est donné par l'expression $r_{d,\bar{i}} - r_{d,i}$. La moyenne arithmétique permet de minimiser la direction prédominante des écarts d'agrégation. Au lieu de prendre exactement la moyenne, nous prenons la valeur **entière** qui permet de minimiser la trajectoire des écarts.

$$r_{d,\bar{i}} = \operatorname{argmin}_{y \in \mathbb{N}} \left| \sum_{i \in f(\bar{i})} y - r_{d,i} \right|$$

Agrégation basée sur l'écart absolu

En utilisant la moyenne arithmétique, on cherche à obtenir la somme algébrique des écarts d'agrégation la plus proche de zéro. En présence d'écarts positifs et négatifs, la moyenne arithmétique a l'inconvénient de masquer l'ampleur des erreurs. L'agrégation basée sur l'écart absolu vise à minimiser la somme des valeurs absolues des écarts ce qui permet de minimiser l'amplitude des erreurs d'agrégation.

$$r_{d,\bar{i}} = \operatorname{argmin}_{y \in \mathbb{N}} \sum_{i \in f(\bar{i})} |y - r_{d,i}|$$

Agrégation basée sur la médiane

Les méthodes d'agrégation présentées précédemment sont sensibles aux valeurs les plus élevées et les moins élevées d'une série de données. Pour lisser l'effet de ces points extrêmes et avoir une meilleure idée sur la tendance centrale des données en éliminant l'effet de bruit, on propose la méthode d'agrégation basée sur la médiane qui considère une valeur centrale.

$$r_{d,\bar{i}} = \operatorname{médiane}_{i \in f(\bar{i})} r_{d,i}$$

Toutes les méthodes utilisées supposent que la demande reste stationnaire à l'intérieur des périodes de deux heures. La recherche d'un niveau ou encore d'une valeur centrale autour de laquelle la demande fluctue sensiblement reste une approximation. Il sera donc inévitable de sous- et de surestimer la demande prévue qui a une variabilité aux 15 minutes. Cette limitation sera corrigée dans l'étape de désagrégation lorsque nous passerons à raffiner la solution.

Agrégation à couverture entière

L'agrégation des réglementations se fait en exprimant les règles de travail et de transfert en fonction de la période agrégée.

L'agrégation à couverture entière agrège les réglementations en prenant les multiples de deux heures les plus proches des intrants du problème initial P . Rappelons que T désigne l'ensemble des types de quart désagrégés (initiaux). Notons par \bar{T} l'ensemble des types de quart agrégés. Pour chaque type de quart désagrégé, on lui associe un type de quart agrégé dont les heures de début et les durées admissibles sont les multiples de deux heures les plus proches des heures de début et durées admissibles du type de quart désagrégé. De même, nous définissons \bar{S}_{min} , \bar{L}_{max} et \bar{B}_{min} comme étant les multiples de deux heures les plus proches de S_{min} (durée minimale de séparation entre quarts consécutifs), L_{max} (charge horaire maximale travaillée par chaque employé) et B_{min} (durée minimale consécutive de travail dans un département donné), respectivement.

Cette méthode a la vocation de fournir un modèle dont les paramètres de couverture sont binaires. En effet, le modèle agrégé considère des courbes de demande aux deux heures. Chaque contrainte de demande s'écrit pour chaque département d et chaque période agrégée $\bar{i} \in \bar{T}$ comme suit :

$$\sum_{e \in E} \sum_{j \in J_e} \sum_{t \in \bar{T}} \sum_{q \in \bar{Q}_{e,j,t}} \delta_{q,d,\bar{i}} X_q + U_{d,\bar{i}} - O_{d,\bar{i}} = r_{d,\bar{i}}$$

avec $\bar{Q}_{e,j,t}$ l'ensemble des quarts agrégés admissibles selon le type de quart agrégé $t \in \bar{T}$. Étant donné que les débuts et durées agrégés sont des multiples de deux heures, le paramètre de couverture $\delta_{q,d,\bar{i}}$ est égal à 1 si le quart agrégé q couvre le département d durant la période de deux heures \bar{i} et 0 sinon.

Agrégation à couverture continue

L'agrégation des réglementations à couverture entière a l'inconvénient de fournir une solution agrégée qui n'est pas forcément réalisable pour le problème désagrégé. Prenons l'exemple d'un type de quart désagrégé dont les durées admissibles sont 7h, 8h et 9h. Les durées admissibles du type de quart agrégé correspondant sont les multiples de deux heures les plus proches de 7h, 8h et 9h et sont donc 6h, 8h et 10h. Un quart agrégé qui fait partie de la solution du problème agrégé peut avoir une durée de 6h qui n'est pas admissible pour le

problème initial. En conséquence, l'étape de désagrégation débute à partir d'une solution non réalisable.

Contrairement à l'agrégation à couverture entière qui agrège les données en calculant les multiples de deux heures les plus proches des données initiales, l'agrégation à couverture continue identifie les données initiales les plus proches des multiples de deux heures. L'inconvénient de cette méthode est que les paramètres de couverture sont des réels et peuvent donc être fractionnaires. Ceci pourra probablement augmenter la complexité des calculs lors de la résolution du problème agrégé.

Pour chaque période agrégée \bar{i} , pour chaque quart agrégé $q \in \bar{Q}$, le paramètre $\delta_{q,d,\bar{i}}$ calcule la proportion des périodes de 15 minutes couvertes par le quart q à l'intérieur d'un intervalle de temps de deux heures ($\delta_{q,d,\bar{i}} = \frac{\sum_{i \in f(\bar{i})} \delta_{q,d,i}}{8}$).

Prenons l'exemple d'un type de quart initial dont les heures de début sont $S_t = \{8h15, 8h30\}$ et les durées admissibles sont toutes les valeurs aux 15 minutes entre 6h et 8h.

- Si nous considérons l'agrégation à couverture continue, le type de quart agrégé possède une seule heure de début admissible qui est 8h15 (car c'est la valeur initiale la plus proche du multiple de 2h qui est 8h00) et possède deux durées admissibles qui sont 6h et 8h. Considérons le quart qui débute à 8h15 et dure 6h dans le département d . Le paramètre de couverture correspondant à ce quart, à la période agrégée associée à l'intervalle de temps $[8h00,10h00[$ et au département d est égal à $\frac{7}{8}$, soit l'intervalle de temps $[8h15,10h00[$ couvert par le quart par rapport à l'intervalle $[8h00,10h00[$.
- À l'opposé, si nous considérons l'agrégation à couverture entière, le type de quart agrégé possède une seule heure de début admissible qui est 8h00 (car c'est le multiple de 2h le plus proche de la valeur initiale 8h15) et possède deux durées admissibles qui sont 6h et 8h. Considérons le quart qui débute à 8h00 et dure 6h dans le département d . Le paramètre de couverture correspondant à ce quart, à la période agrégée associée à l'intervalle de temps $[8h00,10h00[$ et au département d est égal à 1.

L'avantage de l'agrégation à couverture continue consiste à fournir une borne supérieure au problème P . Ainsi, l'étape de désagrégation va débiter à partir d'une solution réalisable. De plus, cette méthode d'agrégation permet d'obtenir un problème \bar{P} plus proche du problème P augmentant ainsi la chance d'obtenir une solution de qualité meilleure lorsqu'elle sera utilisée dans le problème initial.

6.1.2 Choix de la méthode d'agrégation

Dans ce paragraphe, nous expliquons la démarche de sélection de la méthode d'agrégation la plus performante parmi un ensemble de méthodes testées. L'évaluation de la performance des méthodes d'agrégation se fait en se basant sur deux indicateurs : le temps nécessaire pour la réalisation de l'étape d'agrégation et la qualité du calendrier agrégé obtenu.

L'étape d'agrégation commence par l'agrégation des courbes de demande et des réglementations. Nous modélisons ensuite le problème agrégé \bar{P} en utilisant la formulation basée sur les quarts MCQ. Les variables de quart sont définies à partir des types de quart agrégés. Les contraintes de demande correspondent aux courbes de demande agrégées. Les contraintes de liaison entre quarts agrégés sont définies à partir des règles de travail et de transfert horizontales agrégées. Cette formulation est résolue à l'aide d'une heuristique basée sur la génération de colonnes qui sera décrite plus tard dans la Section 5.

Le temps de calcul correspond au temps requis pour réaliser l'étape d'agrégation. Il inclut la définition et la résolution du problème agrégé \bar{P} . Une fois ce problème résolu, nous obtenons un ensemble de variables de quarts agrégés dont la valeur est égale à 1. Un quart est sélectionné pour chaque employé durant chacun de ses jours travaillés. L'ensemble de ces quarts agrégés constitue un calendrier qu'on notera \bar{C} . Rappelons que ce calendrier respecte les réglementations agrégées. En conséquence, il n'est pas forcément réalisable pour le problème initial P étant donné que certaines règles de travail et de transfert peuvent ne pas être respectées. On ne peut donc évaluer la qualité des quarts qui constituent le calendrier agrégé \bar{C} qu'après avoir rétabli la réalisabilité des horaires personnalisés. Afin d'avoir une base de comparaison équitable entre toutes les méthodes, nous cherchons le meilleur calendrier désagrégé C réalisable pour le problème initial P et situé dans un voisinage du calendrier agrégé \bar{C} .

Ce voisinage est formé par des quarts admissibles pour le problème désagrégé P et obtenus par une perturbation des heures de début, des durées et des transferts figurant dans le calendrier agrégé \bar{C} . Pour chaque quart q du calendrier agrégé \bar{C} reçu par l'employé e , nous définissons toutes les heures de début admissibles aux réglementations originales qui appartiennent à l'intervalle [Heure de début de $q - 1H$, Heure de début de $q + 1H$]. De même, nous définissons toutes les durées admissibles aux réglementations originales qui appartiennent à l'intervalle [Durée de $q - 1H$, Durée de $q + 1H$]. Nous obtenons ainsi des quarts vides désagrégés personnalisés de l'employé e . Pour chaque bloc externe agrégé (k, d) de l'employé e , nous définissons toutes les heures de début qui appartiennent à l'intervalle [Heure de début de $k - 1H$, Heure de début de $k + 1H$] et toutes les durées qui appartiennent à l'intervalle [Durée de $k - 1H$, Durée de $k + 1H$]. Nous obtenons ainsi des blocs externes désagrégés personnalisés de l'employé e qui permettent de le transférer au département d'accueil d . Finalement, les quarts externes voisins personnalisés de l'employé e sont obtenus en remplissant ses quarts vides par ses blocs désagrégés. Les quarts internes voisins personnalisés de l'employé e sont obtenus en remplissant ses quarts vides par son département d'origine. Le voisinage $V(\bar{C})$ est l'ensemble de tous les quarts voisins internes et externes de tous les employés de l'organisation.

Ensuite, nous résolvons le problème désagrégé P restreint à l'ensemble des quarts désagrégés voisins appartenant à l'ensemble $V(\bar{C})$. Nous obtenons le calendrier désagrégé C qui

constitue une solution du problème initial P . Soit Z_C la valeur de la fonction objectif du problème désagrégé P lorsque nous retenons le calendrier C . La valeur de Z_C nous permettra d'évaluer la qualité de la solution agrégée \bar{C} .

Nous discuterons dans la Section expérimentale 6.3 du choix de la méthode d'agrégation.

6.2 Problème désagrégé

Une fois le problème agrégé \bar{P} résolu, nous obtenons une solution formée par un ensemble de quarts agrégés internes et externes. À l'étape de désagrégation, nous décomposons le problème en une série de sous-problèmes de plus petite taille. Dans un premier temps, la décomposition se fait par une procédure d'horizon roulant. Au lieu de considérer l'intégralité de l'horizon de planification, nous limitons l'optimisation à une fenêtre de temps spécifique w que l'on spécifiera plus tard. Dans un deuxième temps, pour une fenêtre de temps donnée, le problème restreint à cette fenêtre est décomposé à son tour en une série de sous-problèmes. Chaque sous-problème est associé à un département d . Ces sous-problèmes sont triés selon un ordre spécifique et résolus d'une manière séquentielle. Dans ce qui suit, un sous-problème restreint à la fenêtre w et correspondant au département d est noté $P(w, d)$.

Le principe de la méthode consiste à exploiter l'information provenant de l'étape d'agrégation. Cette information correspond à des décisions agrégées exprimées en termes de deux heures. Elles portent sur les heures de début de travail, les durées de travail et le placement des transferts. Elles ont été déterminées en se basant sur des courbes de demande et des réglementations agrégées. La perte de précision qui découle de cette agrégation doit être rectifiée à l'étape de désagrégation. La question qui se pose à cette étape porte sur la nature de l'information agrégée que l'on doit garder pour construire la solution désagrégée à une échelle temporelle plus précise. Nous allons identifier plusieurs configurations possibles selon l'information retenue. Nous allons par la suite effectuer une analyse comparative des performances de ces configurations afin d'identifier la meilleure pour notre jeu de données.

À l'étape de désagrégation, la période de discrétisation est de 15 minutes. Les données temporelles du problème, à savoir les courbes de demande et les réglementations, sont celles du problème initial P . La taille du problème exclut la résolution directe. Par conséquent, nous concevons une décomposition spatio-temporelle du problème global en faisant appel à la solution agrégée. Dans ce qui suit, nous décrivons les deux mécanismes de décomposition.

6.2.1 Décomposition temporelle

Avant de passer à définir la procédure de décomposition temporelle du problème désagrégé, nous commençons par donner les concepts nécessaires à la définition de cette procédure.

Fenêtre de temps

Une *fenêtre de temps* w est un ensemble consécutif de périodes de l'horizon. Elle est définie par son heure de début s_w et sa durée l_w . Nous définissons une restriction sur la durée minimale des fenêtres de temps. Ainsi, une fenêtre est admissible si et seulement si sa durée dépasse la durée minimale permise notée LW_{min} . La valeur la moins élevée de cette durée minimale est égale à 1 jour. En effet, la procédure d'horizon roulant consiste à décomposer l'horizon en une séquence de fenêtres dont les heures de début constituent une suite croissante monotone. Autrement dit, il faut que l'on puisse avancer dans le temps d'une fenêtre à l'autre sans rester capturé dans une partie de l'horizon de planification. De plus, le problème que nous traitons considère un horizon continu. Par conséquent, deux fenêtres de temps consécutives se chevauchent afin de permettre la réoptimisation de l'intervalle de jointure. La restriction sur la durée minimale des fenêtres de temps permet d'assurer un balayage croissant de l'horizon de planification.

Graphe de transfert

Soit $G_i^{tr}(D, A_i)$ un graphe orienté associé à la période de temps agrégée $i \in \bar{I}$. Les sommets du graphe correspondent aux départements de l'organisation formant l'ensemble D . L'ensemble A_i représente les arcs du graphe. Le poids d'un arc (d, d') est égal au nombre d'employés transférés d'un département donneur d à un département receveur d' durant la période i . Ce nombre est noté $n_{d,d'}^i$. Il est calculé à partir des blocs externes agrégés déterminés par la solution \bar{C} du problème agrégé \bar{P} . Un arc (d, d') existe lorsqu'au moins un employé est transféré de d vers d' dans la solution agrégée \bar{C} (c.-à-d. $n_{d,d'}^i \neq 0$). Le graphe G_i^{tr} est appelé *graphe de transfert relatif à la période i* .

Soit $G_w^{tr}(D, A_w)$ le graphe orienté associé à la fenêtre de temps w avec \bar{I}_w l'ensemble des périodes agrégées formant cette fenêtre. D et A_w représentent respectivement l'ensemble des sommets et des arcs de ce graphe. Le poids d'un arc (d, d') est noté $n_{d,d'}^w$. Il est égal à la somme du nombre d'employés transférés de d vers d' durant les périodes formant la fenêtre de temps w ($n_{d,d'}^w = \sum_{i \in \bar{I}_w} n_{d,d'}^i$). Un arc (d, d') existe lorsque $n_{d,d'}^w \neq 0$. Le graphe G_w^{tr} est appelé *graphe de transfert relatif à la fenêtre w* .

La Figure 6.1 illustre l'exemple de l'information que l'on peut récupérer de la solution agrégée \bar{C} portant sur les transferts entre les cinq départements de l'organisation ($D = \{d_1, d_2, \dots, d_5\}$). Un arc (d, d') énumère les blocs externes agrégés dont le département donneur est d et le département receveur est d' . Prenons l'exemple de l'arc (d_1, d_3) . Il existe quatre blocs de transfert reçus par des employés rattachés au département d_1 et redirigés vers le département d_3 . Deux employés (e_{52}, e_{120}) sont transférés de 8h00 à 10h00, un employé (e_{26}) de 18h00 à 20h00 et un employé (e_{107}) de 22h00 à minuit durant le premier jour de l'horizon.

Un graphe de transfert particulier est celui associé à tout l'horizon de planification. Il contient

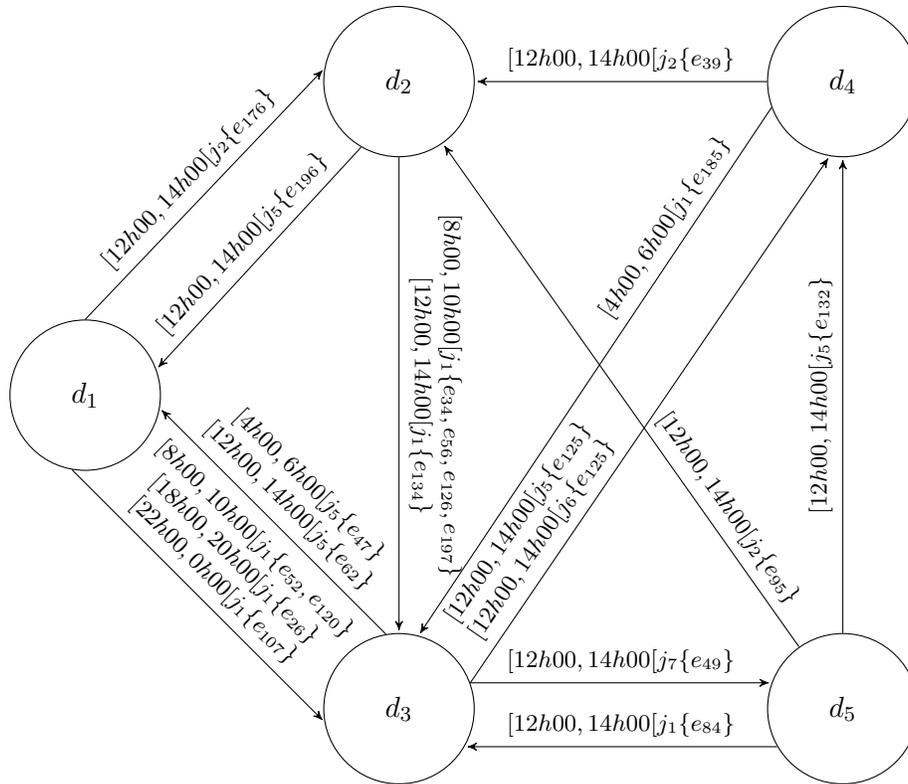


Figure 6.1 – Information sur les blocs externes agrégés

toute l'information provenant des blocs externes agrégés. La Figure 6.2 illustre le graphe G_I^{tr} relatif à l'horizon de planification correspondant à l'exemple de la Figure 6.1. Les sommets du graphe correspondent aux cinq départements. À chaque arc (d, d') , nous calculons le nombre total d'heures transférées du département donneur d au département receveur d' selon la solution agrégée. Prenons l'exemple de l'arc (d_1, d_3) dont le poids est égal à 8h. Ce poids signifie que le département d_1 prête au département d_3 huit heures réparties comme suit : deux employés (e_{52}, e_{120}) de 8h00 à 10h00, un employé (e_{26}) de 18h00 à 20h00 et un employé (e_{107}) de 22h00 à minuit.

Procédure de décomposition de l'horizon en des fenêtres de temps

La décomposition du problème désagrégé en un ensemble de fenêtre de temps se fait d'une manière itérative et dynamique. Le principe de base est de trouver, à chaque itération, la fenêtre de temps qui a la durée admissible la plus élevée définissant un graphe de transfert acyclique. Ainsi, au lieu de résoudre le problème sur tout l'horizon, nous résolvons un problème plus petit restreint à la fenêtre de temps identifiée à chaque itération. Il est à noter que la propriété de l'absence de circuits dans un graphe de transfert est très importante pour la deuxième décomposition par départements.

À chaque itération k , l'objectif est de définir la fenêtre de temps correspondante $w(k)$. Rap-

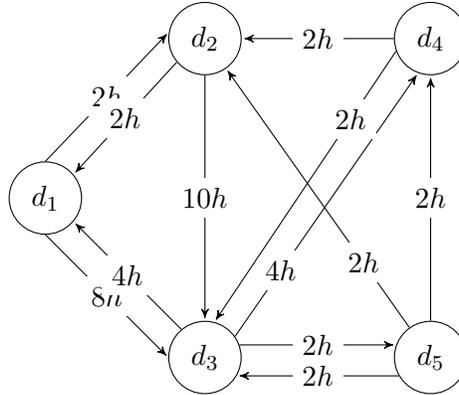


Figure 6.2 – Graphe de transfert G_I^{tr} relatif à l’horizon de planification correspondant à l’exemple de la Figure 6.1

pelons que pour déterminer cette fenêtre, il suffit de calculer son heure de début et sa durée.

À la première itération $k = 1$, la fenêtre de temps commence au début de l’horizon ($s_{w(1)} = i_1$). Nous verrons par la suite comment définir l’heure de début de la fenêtre de temps à une itération quelconque $k > 1$. Ensuite, la durée est initialisée à la durée minimale admissible qui est égale à 1 jour ($l_{w(k)} = 12$ périodes de deux heures). Si nous dépassons la fin de l’horizon, nous déplaçons l’heure de début en arrière dans le temps de manière à former une fenêtre couvrant le dernier jour de l’horizon. Si le graphe de transfert relatif à la fenêtre de temps $w(k)$ est acyclique alors nous avançons dans le temps en augmentant la durée de la fenêtre avec un pas de deux heures. Tant que cette augmentation ne fait pas apparaître de graphe cyclique, nous continuons à avancer dans l’horizon. La condition d’arrêt est d’atteindre la fin de l’horizon ou d’obtenir un circuit dans le graphe de transfert $G_{w(k)}^{tr}$. Ainsi, la durée de la fenêtre de temps sera égale à la dernière durée avant que l’on voit apparaître le ou les circuits.

Un problème se pose lorsque le graphe restreint à la fenêtre de durée minimale admissible est cyclique. Dans ce cas, l’objectif est d’interdire ces circuits en éliminant certains arcs du graphe. Le principe est d’éliminer les arcs de manière à minimiser la perte d’information relative aux transferts agrégés.

L’élimination des arcs se fait au biais d’un problème en nombres entiers EC. Nous commençons par l’énumération exhaustive de tous les circuits élémentaires dans le graphe G_w^{tr} . Soit \mathcal{C} l’ensemble de ces circuits élémentaires. À chaque arc $(d, d') \in A_w$ apparaissant au moins sur l’un des circuits, nous considérons une variable binaire $S_{d,d'}$ qui vaut 1 si l’arc (d, d') est éliminé du graphe et 0 sinon. Un paramètre binaire $\delta_{d,d',c}$ vaut 1 si l’arc (d, d') fait partie du circuit c et 0 sinon.

Modèle (EC)

$$\min \sum_{(d,d') \in A_w} n_{d,d'}^w S_{d,d'} \quad (6.1)$$

$$\text{s.c.} \sum_{(d,d') \in A_w} \delta_{d,d',c} S_{d,d'} \geq 1, \quad \forall c \in \mathcal{C} \quad (6.2)$$

$$S_{d,d'} \in \{0, 1\}, \forall (d, d') \in A_w \quad (6.3)$$

La fonction objectif (6.1) minimise le poids des arcs éliminés. Ce poids calcule le nombre total des périodes transférées déduites de la solution agrégée \bar{C} que l'on va sacrifier. Les contraintes (6.2) assurent l'élimination de tous les circuits. Les contraintes (6.3) imposent l'intégralité des variables.

Pour terminer la description de la procédure de détermination d'une fenêtre de temps $w(k)$, il reste à définir l'heure de début lorsqu'il ne s'agit pas de la première itération ($k > 1$). Ceci se fait en se basant sur les résultats provenant de la fenêtre précédente $w(k-1)$. Lorsque nous définissons une fenêtre de temps, nous résolvons un problème de construction de quarts restreint à cette fenêtre. Une fois la résolution terminée, nous disposons d'un ensemble de quarts personnalisés. Pour définir l'heure de début $s_{w(k)}$, il suffit de se baser sur l'ensemble des quarts sélectionnés au niveau de la fenêtre précédente $w(k-1)$. De cet ensemble, nous identifions les quarts qui débutent à l'intérieur de la fenêtre $w(k-1)$ et se terminent à l'extérieur de la fenêtre $w(k-1)$. Ainsi, nous obtenons l'heure de début de la fenêtre $w(k)$ qui est égale au minimum des débuts de tous ces quarts qui auraient chevauché la fenêtre $w(k)$ si son heure de début était juste après la fin de la fenêtre $w(k-1)$. Le fait de reculer dans l'horizon de planification permet de réoptimiser la construction des quarts sur l'intervalle de temps qui implique les deux fenêtres $w(k-1)$ et $w(k)$. Ceci a la vocation de réduire la sous-optimalité provenant de la décomposition temporelle du problème.

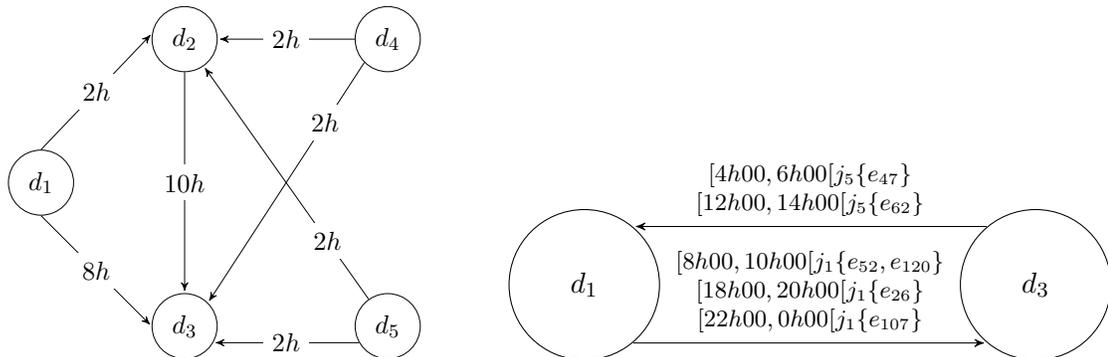


Figure 6.3 – Construction du premier graphe de transfert

En appliquant cette procédure à l'exemple montré dans la Figure 6.1, nous obtenons la première fenêtre de temps w_1 qui commence au début de l'horizon et se termine à 4h00 le jour j_5 . La figure 6.3 donne le graphe de transfert relatif à la fenêtre w_1 . Lors de la détermination de cette fenêtre de temps, le premier circuit qui apparaît correspond au chemin $\langle d_1, d_3, d_1 \rangle$. Dans cet exemple, aucun arc n'a été éliminé du moment que la fenêtre w_1 respecte la contrainte sur la durée minimale.

Une fois la fenêtre de temps w déterminée, nous passons à la résolution du problème de construction de quarts restreint à cette fenêtre.

6.2.2 Décomposition spatiale

Dans cette section, nous décrivons la procédure de résolution du problème désagrégé restreint à une fenêtre de temps w . La méthode exploite la propriété d'absence des circuits dans le graphe de transfert G_w^{tr} pour faire ressortir un ordre spécifique sur l'ensemble de ses sommets. Ainsi, le problème restreint à la fenêtre w est décomposé à son tour en une séquence de problèmes de plus petite taille, un problème par département. Un sous-problème désagrégé $P(w, d)$ est défini pour chaque département d restreint à la fenêtre w . Les sous-problèmes obtenus sont traités selon l'ordre spécifique provenant de la structure du graphe de transfert.

En algorithmique des graphes, il est toujours possible de définir une relation d'ordre partiel sur l'ensemble des sommets d'un graphe acyclique orienté. Cette relation d'ordre permet de respecter la propriété suivante : « *chaque sommet doit toujours être visité avant ses successeurs* ». Cet ordre de visite spécifique est le *tri topologique* du graphe.

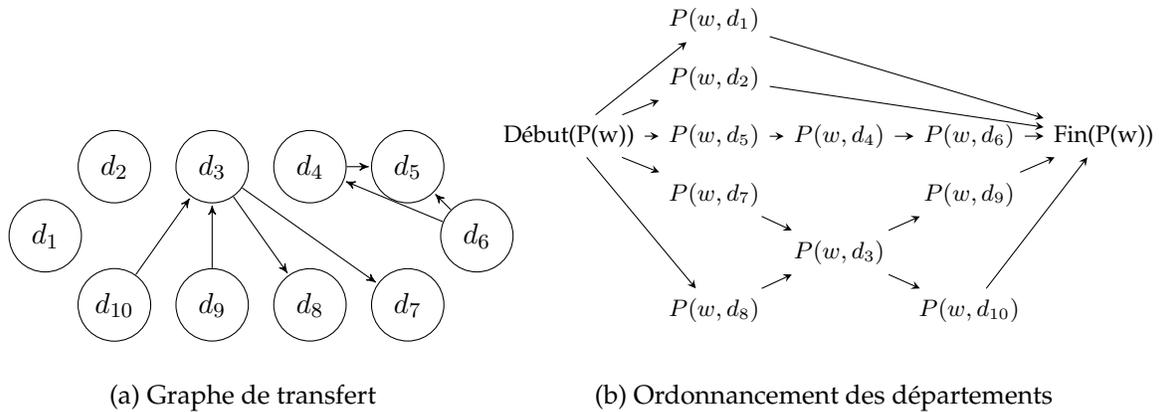
Le *tri topologique inverse* correspond donc à visiter chaque sommet avant ses prédécesseurs. Notre procédure de décomposition spatiale se base sur le tri topologique inverse. Considérons le graphe de transfert acyclique G_w^{tr} . Soit d un prédécesseur immédiat de d' dans le graphe. Le département d est alors un département donneur du département d' du moment qu'il lui transfère des employés. L'ensemble des *départements donneurs* d'un département d correspond à l'ensemble de ses prédécesseurs immédiats dans le graphe de transfert. De même, l'ensemble des *départements receveurs* d'un département d correspond à l'ensemble de ses successeurs immédiats dans le graphe de transfert. Lorsque les départements sont triés selon l'ordre topologique inverse, ils sont visités selon la *règle du receveur avant*.

Le résultat du tri des sommets du graphe de transfert G_w^{tr} constitue l'*ordonnancement des départements*. Étant donné que le tri topologique inverse est une relation d'ordre partiel, le résultat de ce tri ne peut pas être décrit par une simple séquence mais nous utilisons plutôt un *graphe d'ordonnancement* dont les sommets sont les sous-problèmes $\{P(w, d), d \in D\}$ et les arcs sont les relations de précédence. Prenons l'exemple illustré par la Figure 6.4. Initialement, nous disposons du graphe de transfert formé par 10 départements $D = \{d_1, d_2, \dots, d_{10}\}$. L'objectif est de faire ressortir le graphe d'ordonnancement correspondant.

Avant de passer à illustrer l’algorithme de tri permettant d’obtenir un tel graphe, nous commençons par une description de ses sommets et de ses arcs en utilisant notre exemple.

Le graphe d’ordonnement décrit l’ordre de traitement des sous-problèmes $\{P(w, d_1), P(w, d_2), \dots, P(w, d_{10})\}$. À chaque département d correspond un sommet $P(w, d)$ qui consiste à définir et résoudre ce sous-problème. Deux sommets additionnels sont ajoutés et correspondent au début et à la fin de traitement du problème désagrégé $P(w)$ restreint à la fenêtre w . Un sous-problème $P(w, d)$ ne peut être traité que si tous les sous-problèmes $P(w, d')$ prédécesseurs immédiats de $P(w, d)$ soient déjà traités. Ainsi, un arc correspond à une relation de précédence. Dans l’exemple, les premiers sous-problèmes qui constituent le début de la résolution du problème $P(w)$ sont ceux directement liés au sommet de début. Il s’agit des départements d_1, d_2, d_5, d_7 et d_8 . En effet, ces départements sont exclusivement receveurs et nous voulons visiter les départements selon la règle du receveur avant. L’arc qui lie le sous-problème $P(w, d_4)$ au sous-problème $P(w, d_5)$ signifie que la résolution du sous-problème du département d_4 ne peut débuter que lorsque le sous-problème du département d_5 soit résolu. Ceci se traduit par le fait que d_5 est le seul département receveur de d_4 . Chaque chemin du graphe d’ordonnement fait ressortir un ensemble de sous-problèmes qui doivent être résolus séquentiellement.

Pour illustrer l’algorithme de tri, nous utilisons l’exemple de la Figure 6.4. Pour chaque sommet d du graphe de transfert, notons $n^+(d)$ son *demi-degré extérieur* qui calcule le nombre d’arcs partant de ce sommet (c’est à dire le nombre de ses départements receveurs). Le principe de l’algorithme est le suivant : on ne peut placer un sous-problème dans le graphe d’ordonnement que lorsque son demi-degré extérieur devient nul, c’est à dire que tous ses départements receveurs soient déjà placés). Notons L l’ensemble des départements dont les sous-problèmes correspondants ne sont pas encore placés dans le graphe d’ordonnement. Initialement, $L = \{d_1, d_2, \dots, d_{10}\}$. Pour chaque département $d \in L$, nous calculons son demi-degré extérieur $n^+(d)$. Nous identifions l’ensemble des départements ayant un demi-degré extérieur nul donc les départements exclusivement receveurs. Pour l’exemple de la Figure 6.4, cet ensemble est $\{d_1, d_2, d_5, d_7, d_8\}$. Les sous-problèmes associés à ces départements sont placés dans le graphe d’ordonnement. Ils sont tous liés au sommet de début. À la deuxième itération, l’ensemble des départements dont les sous-problèmes ne sont pas encore placés dans le graphe d’ordonnement devient $L = \{d_3, d_4, d_6, d_9, d_{10}\}$. Pour chaque département $d \in L$, nous calculons le nouveau demi-degré extérieur (après suppression des départements receveurs de d dont les sous-problèmes étaient placés dans le graphe d’ordonnement à l’itération précédente). Si le demi-degré extérieur de d est nul alors le sous-problème $P(w, d)$ est placé dans le graphe d’ordonnement comme successeur immédiat des sous-problèmes placés à l’itération précédente et correspondant à des départements receveurs de d . Prenons d_3 comme exemple. Les départements receveurs de d_3 placés à la première itération sont d_7 et d_8 . Le demi-degré extérieur de d_3 (qui était égal



(a) Graphe de transfert

(b) Ordonnancement des départements

Itération 1	<table border="1"> <thead> <tr> <th>d</th> <th>d_1</th> <th>d_2</th> <th>d_3</th> <th>d_4</th> <th>d_5</th> <th>d_6</th> <th>d_7</th> <th>d_8</th> <th>d_9</th> <th>d_{10}</th> </tr> </thead> <tbody> <tr> <td>$n^+(d)$</td> <td>0</td> <td>0</td> <td>2</td> <td>1</td> <td>0</td> <td>2</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> </tbody> </table>	d	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	$n^+(d)$	0	0	2	1	0	2	0	0	1	1
d	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}													
$n^+(d)$	0	0	2	1	0	2	0	0	1	1													
Itération 2	<table border="1"> <thead> <tr> <th>d</th> <th>d_3</th> <th>d_4</th> <th>d_6</th> <th>d_9</th> <th>d_{10}</th> </tr> </thead> <tbody> <tr> <td>$n^+(d)$</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> <td>1</td> </tr> <tr> <td>Départements receveurs de d tel que $P(w, d)$ soit placé à l'itération 1</td> <td>$\{d_7, d_8\}$</td> <td>$\{d_5\}$</td> <td>-</td> <td>-</td> <td>-</td> </tr> </tbody> </table>	d	d_3	d_4	d_6	d_9	d_{10}	$n^+(d)$	0	0	1	1	1	Départements receveurs de d tel que $P(w, d)$ soit placé à l'itération 1	$\{d_7, d_8\}$	$\{d_5\}$	-	-	-				
d	d_3	d_4	d_6	d_9	d_{10}																		
$n^+(d)$	0	0	1	1	1																		
Départements receveurs de d tel que $P(w, d)$ soit placé à l'itération 1	$\{d_7, d_8\}$	$\{d_5\}$	-	-	-																		
Itération 3	<table border="1"> <thead> <tr> <th>d</th> <th>d_6</th> <th>d_9</th> <th>d_{10}</th> </tr> </thead> <tbody> <tr> <td>$n^+(d)$</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>Départements receveurs de d tel que $P(w, d)$ soit placé à l'itération 2</td> <td>$\{d_4\}$</td> <td>$\{d_3\}$</td> <td>$\{d_3\}$</td> </tr> </tbody> </table>	d	d_6	d_9	d_{10}	$n^+(d)$	0	0	0	Départements receveurs de d tel que $P(w, d)$ soit placé à l'itération 2	$\{d_4\}$	$\{d_3\}$	$\{d_3\}$										
d	d_6	d_9	d_{10}																				
$n^+(d)$	0	0	0																				
Départements receveurs de d tel que $P(w, d)$ soit placé à l'itération 2	$\{d_4\}$	$\{d_3\}$	$\{d_3\}$																				

(c) Algorithme de tri

Figure 6.4 – Tri des départements

à 2) est donc diminué de deux. Étant donné que $n^+(d_3) = 0$, nous plaçons $P(w, d_3)$ dans le graphe d'ordonnancement comme successeur immédiat de $P(w, d_7)$ et $P(w, d_8)$. L'ensemble des départements dont les sous-problèmes sont placés à la deuxième itération correspond à $\{d_3, d_4\}$.

Ce processus itératif est répété jusqu'à ce que tous les sous-problèmes soient placés dans le graphe d'ordonnancement. Finalement, tous les sous-problèmes sans successeurs dans le graphe d'ordonnancement sont liés au sommet de fin.

Notons que des sous-problèmes ayant le même prédécesseur immédiat peuvent être traités simultanément. En informatique, le parallélisme est le paradigme dominant permettant de réaliser un ensemble d'opérations simultanément. La conception d'un algorithme dont l'architecture se prête à une implémentation parallèle est très importante pour diminuer les temps de calcul. Dans le cadre de cette thèse, nous avons utilisé un processus séquentiel lors de l'implémentation de l'heuristique et l'élaboration des expérimentations. Ceci dit, il est

important de souligner que notre heuristique de prétransfert a la vocation de permettre un tel parallélisme.

Dans l'implémentation séquentielle adoptée lors des expérimentations pour notre exemple, nous pouvons considérer la séquence $\langle P(w, d_1), P(w, d_2), P(w, d_5), P(w, d_4), P(w, d_6), P(w, d_7), P(w, d_8), P(w, d_3), P(w, d_9), P(w, d_{10}) \rangle$.

Dans ce qui suit, nous supposons que les départements sont triés selon un ordre topologique inverse et qu'ils sont visités dans cet ordre en donnant la priorité au département receveur avant.

6.2.3 Énumération des quarts désagrégés

Le principe de l'heuristique de prétransfert est d'exploiter l'information provenant du problème agrégé \bar{P} pour définir et résoudre le problème désagrégé P . Nous avons vu précédemment comment utiliser les blocs externes agrégés pour décomposer le problème global en des sous-problèmes plus faciles à résoudre. Dans cette section, nous expliquons comment exploiter la solution agrégée pour définir un ensemble restreint de quarts désagrégés qui seront considérés par le problème désagrégé P . Rappelons que Q correspond à l'ensemble de tous les quarts admissibles du problème global P . L'objectif est de définir l'ensemble restreint des quarts désagrégés $\tilde{Q} \subseteq Q$ qui sont considérés lors de la résolution du problème désagrégé P . L'énumération de ces quarts se fait en deux étapes : la sélection d'un ensemble de quarts vides puis le placement des départements à l'intérieur de ces quarts.

Sélection des quarts vides

Rappelons qu'un quart vide q est caractérisé par son heure de début s_q et sa durée l_q . Un quart vide débutant le jour $j \in J$ est admissible s'il existe un type de quart désagrégé $t \in T$ tel que $s_q \in S_{t,j}$ et $l_q \in L_t$. La sélection des quarts vides admissibles se fait de deux manières : selon la solution agrégée ou indépendamment de cette solution.

Dans le premier cas, nous nous basons sur l'information provenant de la solution agrégée pour délimiter l'ensemble restreint des quarts vides à considérer pour le problème désagrégé. Rappelons que le problème agrégé \bar{P} est modélisé par une formulation basée sur les quarts MCQ. Le calendrier \bar{C} correspond à la meilleure solution réalisable trouvée pour ce problème. Elle peut être décrite par l'ensemble des quarts agrégés $q \in \bar{Q}$ ayant des valeurs non nuls ($X_q = 1$). Notons $\bar{q}_{e,j}$ le quart agrégé sélectionné pour l'employé $e \in E$ durant son jour de travail $j \in J_e$. Le calendrier agrégé peut être décrit formellement comme suit : $\bar{C} = (\bar{q}_{e,j})_{e \in E, j \in J_e}$. Les quarts vides voisins sont obtenus par une perturbation des heures de début et des durées de travail formant la solution agrégée. Soit θ^l l'amplitude de cette perturbation. L'ensemble des quarts vides situés dans un voisinage du quart agrégé $\bar{q}_{e,j}$ est formé par tous les quarts vides admissibles tels que leurs heures de début soient comprises

entre $s_{\bar{q}_{e,j}} - \theta^l$ et $s_{\bar{q}_{e,j}} + \theta^l$ et leurs durées soient comprises entre $l_{\bar{q}_{e,j}} - \theta^l$ et $l_{\bar{q}_{e,j}} + \theta^l$. Dans ce cas, l'ensemble des quarts vides peut être exprimé comme suit : $Q^{vides} = \cup_{e \in E} \cup_{j \in J_e} Q_{e,j}^{vides}$ avec $Q_{e,j}^{vides} = \cup_{j' \in J_e} \{(s_q, l_q) | q \in Q_{e,j}, s_{\bar{q}_{e,j'}} - \theta^l \leq s_q \leq s_{\bar{q}_{e,j'}} + \theta^l, l_{\bar{q}_{e,j'}} - \theta^l \leq l_q \leq l_{\bar{q}_{e,j'}} + \theta^l\}$.

Dans le deuxième cas, tous les quarts vides admissibles sont considérés. Une énumération exhaustive de toutes les heures de début et toutes les durées de travail admissibles est réalisée à partir de l'ensemble des types de quart désagrégés T . Dans ce cas, l'ensemble des quarts vides est exprimé comme suit : $Q^{vides} = \cup_{e \in E} \cup_{j \in J_e} Q_{e,j}^{vides}$ avec $Q_{e,j}^{vides} = Q_{e,j}$ l'ensemble de tous les quarts admissibles de l'employé e le jour travaillé $j \in J_e$.

Placement des départements

L'ensemble restreint des quarts internes désagrégés \tilde{Q}^{int} est obtenu en remplissant les quarts vides par les départements d'origine. L'ensemble de ces quarts est décrit par l'expression suivante : $\tilde{Q}^{int} = \{(s_q, l_q, 0, d_e, d_0) | q \in Q^{vides}\}$. Notons que lorsque les quarts vides sont obtenus par une énumération de tous les débuts et les durées admissibles sans égard à la solution agrégée, l'ensemble \tilde{Q}^{int} contient tous les quarts internes désagrégés admissibles ($\tilde{Q}^{int} = Q^{int}$).

L'ensemble restreint des quarts externes désagrégés \tilde{Q}^{ext} est défini en se basant sur la solution agrégée \bar{C} selon plusieurs possibilités. Nous commençons par identifier l'ensemble des blocs externes agrégés noté \bar{B} . Chaque bloc externe personnalisé $(\bar{k}, d)_e \in \bar{B}$ est défini par son heure de début $s_{\bar{k}}$, sa durée $l_{\bar{k}}$, l'employé e recevant ce bloc dans la solution agrégée et le département receveur dans lequel ce bloc sera réalisé d . Soit $\bar{B}((\bar{k}, d)_{d'}) = \{(\bar{k}, d)_e \in \bar{B} | e \in E_{d'}\}$ l'ensemble de tous les blocs personnalisés ayant la même heure de début et la même durée correspondante au segment \bar{k} , se réalisant dans le même département receveur d et transférés par le même département donneur d' . Les ensembles $\bar{B}((\bar{k}, d)_{d'})$ constituent une partition de l'ensemble \bar{B} . Un bloc $(\bar{k}, d)_{d'}$ est dit anonyme du moment que l'on ne spécifie pas l'employé qui le reçoit. Un bloc anonyme est défini donc par son heure de début $s_{\bar{k}}$, sa durée $l_{\bar{k}}$, le département donneur d' transférant l'employé durant ce bloc et le département receveur dans lequel il sera réalisé d . Soit $\bar{B}^{anonymes}$ l'ensemble des blocs externes agrégés anonymes. À chaque bloc anonyme $(\bar{k}, d)_{d'} \in \bar{B}^{anonymes}$, notons $n_{\bar{k},d,d'}$ le nombre d'employés recevant ce bloc dans le calendrier agrégé \bar{C} .

Dans un premier temps, nous commençons par énumérer l'ensemble des blocs externes désagrégés anonymes $B^{anonymes}$ à partir de l'ensemble des blocs externes agrégés anonymes $\bar{B}^{anonymes}$ selon un critère que l'on définira. Dans un deuxième temps, nous affectons chaque bloc externe désagrégé anonyme à un ensemble spécifique d'employés selon un critère que l'on spécifiera.

Pour chaque bloc $(\bar{k}, d)_{d'} \in \bar{B}^{anonymes}$, nous énumérons un ensemble de blocs externes désagrégés anonymes $B^{anonymes}((\bar{k}, d)_{d'})$ selon deux possibilités :

- Nous retenons l'information provenant de la solution agrégée en gardant le même bloc agrégé $(\bar{k}, d)_{d'}$ sans aucune perturbation.
- Les débuts et les durées du bloc externe agrégé sont perturbés. Soit θ^{tr} l'amplitude de cette perturbation. L'ensemble des blocs externes désagrégés dans un voisinage de $(\bar{k}, d)_{d'}$ est formé par tous les blocs anonymes fournis par le département d' au département d et dont les heures de début sont situées entre $s_{\bar{k}} - \theta^{tr}$ et $s_{\bar{k}} + \theta^{tr}$ et les durées sont comprises entre $\max\{B_{min}, l_{\bar{k}} - \theta^{tr}\}$ et $l_{\bar{k}} + \theta^{tr}$.

Une fois les blocs externes désagrégés anonymes énumérés, nous passons à les affecter à un sous-ensemble spécifique d'employés. Deux possibilités sont considérées selon l'information provenant de la solution agrégée que l'on décide de retenir.

- *Identité de l'employé* : Pour chaque bloc externe agrégé personnalisé $(\bar{k}, d)_e \in \bar{B}$, il existe un et un seul bloc externe agrégé anonyme $(\bar{k}, d)_{d_e} \in \bar{B}^{anonymes}$. L'ensemble des blocs externes désagrégés personnalisés de l'employé e est donné par l'expression $B_e = \{(k, d)_e | (k, d)_{d_e} \in B^{anonymes}((\bar{k}, d)_{d_e})\}$.
- *Identité du département donneur* : Pour chaque bloc externe agrégé anonyme $(\bar{k}, d)_{d'} \in \bar{B}^{anonymes}$ et pour chaque employé e rattaché au département d' et qualifié pour travailler dans le département d , nous considérons l'ensemble des blocs externes désagrégés personnalisés relativement au bloc agrégé anonyme $(\bar{k}, d)_{d'}$ donné par l'expression $B_e((\bar{k}, d)_{d'}) = \{(k, d)_e | (k, d)_{d'} \in B^{anonymes}((\bar{k}, d)_{d'})\}$. Ainsi, nous obtenons l'ensemble B_e comme étant l'union de tous les ensembles $B_e((\bar{k}, d)_{d'})$.

Finalement, nous plaçons les blocs externes désagrégés personnalisés B_e à l'intérieur des quarts vides Q_e^{vides} énumérés précédemment. Considérons le bloc $(k, d)_e \in B_e$ et le quart vide $q \in Q_e^{vides}$. Uniquement les trois cas suivants mènent à l'obtention d'un quart externe désagrégé :

- Le bloc $(k, d)_e$ et le quart vide q débutent à la même période et $l_q - l_k \geq B_{min}$. Nous obtenons le quart désagrégé $(s_q, l_k, l_q - l_k, d, d_e)$ débutant à l'externe et se terminant à l'interne.
- Le bloc $(k, d)_e$ et le quart vide q se terminent à la même période et $l_q - l_k \geq B_{min}$. Nous obtenons le quart désagrégé $(s_q, l_q - l_k, l_k, d_e, d)$ débutant à l'interne et se terminant à l'externe.
- Le bloc $(k, d)_e$ et le quart vide q débutent et se terminent à la même période. Nous obtenons le quart désagrégé $(s_q, l_q, 0, d, d_0)$ se déroulant entièrement à l'externe (finissant dans le département fictif d_0).

Ainsi, nous obtenons l'ensemble restreint des quarts externes désagrégés \tilde{Q}^{ext} .

Lorsque nous choisissons de retenir l'identité des départements donneurs, les quarts externes sont partitionnés selon le bloc externe agrégé anonyme qui a servi à les construire. Nous verrons par la suite comment utiliser cette partition pour incorporer une contrainte

additionnelle dans le modèle correspondant à un sous-problème $P(w, d)$. Pour chaque bloc externe agrégé anonyme $(\bar{k}, d)_{d'} \in \bar{B}^{anonymes}$, notons $\tilde{Q}^{ext}((\bar{k}, d)_{d'})$ l'ensemble des quarts externes obtenus en considérant les blocs externes désagrégés personnalisés appartenant à l'un des ensembles $B_e((\bar{k}, d)_{d'})$.

6.2.4 Procédure de résolution espace-temps

Une fois l'ensemble restreint des quarts internes et externes désagrégés, qu'on a noté \tilde{Q} , énuméré, nous passons à la résolution du problème désagrégé P restreint à \tilde{Q} en le décomposant en des sous-problèmes de plus petite taille. Chaque sous-problème $P(w, d)$, correspondant à une fenêtre de temps w et un département d , est modélisé à l'aide de la formulation basée sur les quarts MCQ. Ce sous-problème est restreint à un sous-ensemble de quarts $\tilde{Q}(w, d) \subseteq \tilde{Q}$ que l'on spécifiera dans la partie 6.2.5.

La solution finale du problème désagrégé P est obtenue par la résolution d'une procédure de décomposition espace-temps. Cette solution peut être décrite par un calendrier désagrégé $C = (q_{e,j})_{e \in E, j \in J_e}$ avec $q_{e,j}$ le quart désagrégé reçu par l'employé e le jour travaillé $j \in J_e$. Notre méthode construit d'une manière gloutonne ce calendrier C . Ainsi, pour un sous-problème $P(w, d)$, nous distinguons deux cas : un employé peut avoir déjà reçu un quart durant son jour travaillé ou le contraire. Au fur et à mesure que l'on avance dans le temps, l'horaire de l'employé se construit jusqu'à ce qu'il reçoit un quart durant chaque jour travaillé.

Pour décrire la procédure de décomposition espace-temps, revenons à l'exemple de la Figure 6.3. Il s'agit du graphe de transfert acyclique associé à la fenêtre de temps w_1 qui débute à minuit le premier jour de l'horizon et se termine à 4h00 du jour j_5 . Initialement, aucun quart n'a été sélectionné pour aucun employé. Pour la résolution du sous-problème $P(w_1)$, il faut commencer par trier l'ensemble des départements selon la règle du receveur avant. Une séquence possible est $\langle P(w_1, d_3), P(w_1, d_2), P(w_1, d_1), P(w_1, d_4), P(w_1, d_5) \rangle$. Nous commençons par résoudre le sous-problème $P(w_1, d_3)$. Le résultat de ce sous-problème consiste à sélectionner un ensemble de quarts débutant à l'intérieur de la fenêtre w_1 pour les employés internes de d_3 et certains employés provenant des départements donneurs de d_3 dans le graphe de transfert (qui sont d_1, d_2, d_4 et d_5). Ce sous-problème permet donc de satisfaire aux besoins du département d_3 en utilisant sa main d'œuvre interne et en réservant une main d'œuvre externe dont la provenance est déduite du graphe de transfert agrégé. Une fois le sous-problème résolu, nous passons à exécuter la même procédure au niveau du département d_2 . Le seul changement est qu'il se peut que certains employés internes de d_2 ont été transférés (réservés) précédemment au département d_3 lors de la résolution du sous-problème $P(w_1, d_3)$. Dans ce cas, il faut respecter les restrictions quant aux périodes transférées des quarts sélectionnés au niveau du sous-problème $P(w_1, d_3)$. Nous révisons alors les blocs internes de ces quarts pour les ajuster à leur tour à la demande au niveau

du département d_2 et nous gardons figés les blocs externes. Nous faisons de même pour le reste des départements. Au fur et à mesure, le calendrier se construit jusqu'à ce que tous les employés possèdent des quarts sélectionnés débutant à l'intérieur de w_1 . Nous calculons la nouvelle fenêtre w_2 selon la procédure de décomposition temporelle. Nous déterminons le nouveau graphe de transfert $G_{w_2}^{tr}$ qui lui correspond. La résolution du sous-problème $P(w_2)$ est contrainte par tous les quarts qui ont été déjà sélectionnés précédemment. On itère jusqu'à ce que l'horizon de planification soit totalement parcouru et tous les départements au niveau de la dernière fenêtre soient visités. À la fin de l'algorithme, chaque employé doit recevoir un quart durant chaque jour travaillé.

Passons à décrire un sous-problème $P(w, d)$ restreint à une fenêtre de temps w et associé à un département d .

6.2.5 Sous-problème restreint à une fenêtre de temps et associé à un département

Chaque sous-problème $P(w, d)$ est modélisé à l'aide de la formulation basée sur les quarts MCQ. Dans la suite, nous spécifions les variables de décision et les contraintes de ce sous-problème.

Variables de décision

Étant donné que la formulation choisie est basée sur les quarts, nous considérons des variables de quart. L'objectif de cette partie est de spécifier l'ensemble des quarts désagrégés énumérés restreint au sous-problème $P(w, d)$, qu'on a noté $\tilde{Q}(w, d)$ avec $\tilde{Q}(w, d) \subseteq \tilde{Q}$. Pour chaque quart $q \in \tilde{Q}(w, d)$, nous définissons une variable de décision binaire X_q qui est égale à 1 si le quart q est sélectionné et 0 sinon.

Pour définir l'ensemble $\tilde{Q}(w, d)$ des quarts considérés par le sous-problème $P(w, d)$, nous distinguons deux types d'employés : les employés internes E_d et un sous-ensemble spécifique d'employés externes $E_d^{ext}(w)$.

Pour chaque employé interne $e \in E_d$, pour chaque jour travaillé $j \in J_e$ de cet employé, les cas suivants se présentent :

- L'employé e a déjà reçu un quart débutant avant la fenêtre w . Ceci signifie que ce quart a été sélectionné lors de la résolution d'un sous-problème associé à une fenêtre de temps précédente à w . Dans ce cas, le quart est ajouté à l'ensemble des quarts $\tilde{Q}(w, d)$. Par ailleurs, il sera le seul quart personnalisé correspondant à ce jour.
- L'employé e a déjà reçu un quart débutant à l'intérieur de la fenêtre w . Ce quart est nécessairement un quart externe utilisé pour transférer l'employé e à un département d' receveur de d . Ce quart provient de la résolution du sous-problème $P(w, d')$ avec d' un prédécesseur immédiat de d dans le graphe de transfert G_w^{tr} . Nous réoptimisons le

bloc interne formant le quart $q_{e,j}$ tout en gardant le bloc externe inchangé. Pour chaque quart désagrégé externe énuméré $q \in \tilde{Q}$ correspondant à l'employé e et au jour j , si le quart q débute à l'intérieur de la fenêtre w et si le bloc externe du quart q est égal au bloc externe du quart $q_{e,j}$ alors nous ajoutons le quart q à l'ensemble des quarts $\tilde{Q}(w, d)$.

- L'employé e n'a pas encore reçu de quart. Pour chaque quart interne $q \in \tilde{Q}$ correspondant à l'employé e et au jour j , si ce quart débute à l'intérieur de la fenêtre w alors nous l'ajoutons à l'ensemble des quarts $\tilde{Q}(w, d)$. Notons que cet ensemble de quarts internes débutant à l'intérieur de la fenêtre w sert à optimiser le problème désagrégé P au niveau du sous-problème $P(w, d)$. En d'autres termes, cet ensemble de quarts internes sert à s'ajuster à un sous-ensemble de courbes de demande correspondantes au sous-problème $P(w, d)$ que l'on précisera plus tard.
- L'employé e n'a pas encore reçu de quart. Pour chaque quart interne $q \in \tilde{Q}$ correspondant à l'employé e et au jour j , si ce quart débute après la fin de la fenêtre w alors nous l'ajoutons à l'ensemble des quarts $\tilde{Q}(w, d)$. Notons que cet ensemble de quarts internes débutant après la fin de la fenêtre w sert à respecter les contraintes horizontales de travail et de transfert afin d'assurer l'admissibilité des horaires.

L'ensemble spécifique des employés externes considérés dans le sous-problème $P(w, d)$ diffère selon que l'on considère l'identité de l'employé donné ou l'identité du département donneur. Pour identifier cet ensemble $E_d^{ext}(w)$, nous procédons comme suit. D'abord, nous identifions l'ensemble des blocs externes agrégés anonymes $(\bar{k}, d)_{d'}$ qui vérifient les deux conditions suivantes : ils interceptent la fenêtre de temps w et ils correspondent à un département donneur d' tel que l'arc (d', d) existe dans le graphe de transfert G_w^{tr} . La deuxième condition est importante du moment que certains arcs peuvent être supprimés du graphe de transfert en présence de circuits par le modèle d'élimination de circuits EC.

Pour chaque employé externe $e \in E_d^{ext}$ et chaque jour durant lequel cet employé n'a pas encore reçu de quart dans le calendrier désagrégé C , nous cherchons l'ensemble des quarts externes $q \in \tilde{Q}$ reçus par l'employé e le jour j vérifiant les deux conditions suivantes : le quart q débute à l'intérieur de la fenêtre w et le quart q est obtenu à partir d'un bloc anonyme agrégé permettant de le transférer de son département d'origine vers le département d . Si l'ensemble de ces quarts est non vide alors l'employé e appartient à l'ensemble des employés externes $E_d^{ext}(w)$ correspondant au sous-problème $P(w, d)$.

Pour chaque employé externe $e \in E_d^{ext}(w)$, pour chaque jour travaillé $j \in J_e$ de cet employé, les cas suivants se présentent :

- L'employé e a déjà reçu un quart. Ce quart débute nécessairement avant la fenêtre w . Dans ce cas, nous ajoutons ce quart à l'ensemble $\tilde{Q}(w, d)$.
- L'employé e n'a pas encore reçu de quart. Pour chaque bloc anonyme agrégé interceptant la fenêtre w et correspondant à un département donneur d' dans le graphe de

transfert G_w^{tr} , pour chaque quart externe obtenu à partir de ce bloc et débutant à l'intérieur de la fenêtre w , nous ajoutons ce quart dans l'ensemble $\tilde{Q}(w, d)$. Notons que cet ensemble est non vide. Il sert à optimiser le problème désagrégé P au niveau du sous-problème $P(w, d)$. En d'autres termes, cet ensemble de quarts externes sert à s'ajuster à un sous-ensemble de courbes de demande correspondantes au sous-problème $P(w, d)$ que l'on précisera plus tard.

- L'employé e n'a pas encore reçu de quart. Pour chaque quart interne $q \in \tilde{Q}$ correspondant à l'employé e et au jour j , si ce quart débute à l'intérieur de la fenêtre w ou après la fin de cette fenêtre alors nous l'ajoutons à l'ensemble des quarts $\tilde{Q}(w, d)$. Notons que cet ensemble de quarts internes sert à respecter les contraintes horizontales de travail et de transfert afin d'assurer l'admissibilité des horaires.

En plus des variables de quart X , nous définissons un ensemble de variables de sous-couverture U et de surcouverture O correspondant à un ensemble restreint de contraintes de demande associées à la fenêtre w et au département d . Dans ce qui suit, nous passons à spécifier ces courbes de demande.

Contraintes

Les contraintes de la formulation MCQ du sous-problème $P(w, d)$ sont les contraintes de demande, les contraintes d'admissibilité des horaires et d'autres contraintes additionnelles permettant le respect de certaines informations sur les transferts provenant du calendrier agrégé \tilde{C} .

Dans la suite, on note $\tilde{Q}_{e,j}(w, d)$ l'ensemble des quarts appartenant à $\tilde{Q}(w, d)$ et correspondant à l'employé e et à son jour travaillé $j \in J_e$.

Le sous-problème $P(w, d)$ sert à couvrir les besoins du département d durant la fenêtre de temps w . Ainsi, les contraintes de demande sont celles correspondant au département d et aux périodes désagrégées I_w associées à la fenêtre w . De plus, on ajoute un ensemble de périodes débutant après la fin de la fenêtre w afin de permettre de s'ajuster à la demande du département d durant la plage de temps couverte par des quarts de l'ensemble $\tilde{Q}(w, d)$ qui débutent à l'intérieur de w et se terminent après w . Pour définir cet ensemble, cherchons la période maximale couverte par au moins un quart $q \in \tilde{Q}(w, d)$ débutant à l'intérieur de la fenêtre w et couvrant le département d durant cette période. Considérons la fenêtre de temps w' possédant la même heure de début de la fenêtre w et se terminant à la période maximale identifiée. Soit $I_{w'}$ l'ensemble des périodes désagrégées associées à la fenêtre w' . Les contraintes de demande correspondant au sous-problème $P(w, d)$ s'écrivent comme suit :

$$\sum_{e \in E_d \cup E_d^{ext}(w)} \sum_{j \in J_e} \sum_{q \in \tilde{Q}_{e,j}(w, d)} \delta_{q,d,i} X_q + U_{d,i} - O_{d,i} = r_{d,i}, \quad \forall i \in I_{w'} \quad (6.4)$$

Les contraintes d'admissibilité des horaires s'écrivent pour chaque employé $e \in E_d \cup E_d^{ext}(w)$ en se limitant à l'ensemble restreint des quarts désagrégés correspondant au sous-problème $P(w, d)$. Ces contraintes s'écrivent en remplaçant l'ensemble $Q_{e,j}$ par $\tilde{Q}_{e,j}(w, d)$ dans les équations (4.8)-(4.11). Notons que les règles de travail et de transfert horizontales pour la liaison entre les quarts désagrégés correspondent aux réglementations du problème désagrégé P . Les règles de travail et de transfert ponctuelles sont explicitement respectées lors de l'énumération des quarts désagrégés $\tilde{Q}_{e,j}$.

Lorsque les quarts externes désagrégés \tilde{Q} sont construits en gardant l'identité du département donneur, il faut rajouter un ensemble de contraintes additionnelles afin de limiter le nombre d'employés transférés d'un département donneur d' au département receveur d pour chaque bloc externe anonyme interceptant la fenêtre w et correspondant à l'arc (d', d) du graphe de transfert G_w^{tr} (Notons $\bar{B}^{anonymes}(w, d)$ l'ensemble de ces blocs). En effet, sans l'ajout de cette contrainte, nous risquons de surestimer les transferts afin de couvrir les courbes de demande du sous-problème $P(w, d)$ au dépend de la couverture des courbes de demande du sous-problème $P(w, d')$ qui sera traité ultérieurement. Pour chaque bloc anonyme agrégé $(\bar{k}, d)_{d'} \in \bar{B}^{anonymes}(w, d)$ et pour chaque quart $q \in \tilde{Q}(w, d)$, on pose le paramètre $\delta_{q,(\bar{k},d)_{d'}}$ qui vaut 1 si le quart q est obtenu à partir du bloc $(\bar{k}, d)_{d'}$ et 0 sinon.

$$\sum_{q \in \tilde{Q}(w, d)} \delta_{q,(\bar{k},d)_{d'}} X_q \leq n_{\bar{k},d,d'}, \quad \forall (\bar{k}, d)_{d'} \in \bar{B}^{anonymes}(w, d) \quad (6.5)$$

6.3 Analyse de performance de l'approche de résolution prétransfert

Le but de cette partie expérimentale est de faire les meilleurs choix quant aux modules composant l'heuristique de prétransfert. Cette heuristique d'optimisation distribuée qui intègre plusieurs problèmes décisionnels dans le même modèle fait appel à deux étapes : l'agrégation et la désagrégation. Chacune de ces étapes implique un ensemble de paramètres et de techniques d'optimisation qui doivent être étudiés. Des analyses comparatives sont effectuées afin de choisir les valeurs de ces paramètres et les techniques offrant les meilleures performances.

Cette section est divisée en deux parties. La première partie concerne l'étape d'agrégation. Elle constitue une analyse de l'effet des outils d'agrégation utilisés sur la qualité des calendriers agrégés obtenus. La deuxième partie étudie l'étape de désagrégation. Étant donné que la performance de cette étape dépend étroitement de l'information à retenir du niveau agrégé, l'analyse porte sur l'effet de cette information sur la qualité des calendriers désagrégés obtenus.

6.3.1 Problème agrégé

Le problème \bar{P} est défini à partir du problème P en agrégeant deux types de données : les réglementations et les courbes de demande. Rappelons que nous proposons trois méthodes pour agréger les courbes de demande : l'agrégation basée sur la moyenne (AV), l'agrégation basée sur l'écart absolu (ABS) et l'agrégation basée sur la médiane (MED). Rappelons également que nous proposons deux méthodes pour agréger les réglementations : l'agrégation à couverture entière (2H) et l'agrégation à couverture continue (NEAR2H).

Cette partie de la thèse constitue une analyse comparative entre les différentes manières d'agréger le problème. Elle a pour but de choisir la meilleure méthode d'agrégation à retenir et à incorporer dans notre heuristique de prétransfert. Le Tableau 6.1 montre les six différentes méthodes étudiées. Chaque méthode est désignée par la manière d'agréger les courbes de demande d'une part et la manière d'agréger les réglementations d'autre part. Par exemple, la méthode (AV,2H) agrège les courbes de demande en se basant sur la moyenne et agrège les réglementations en utilisant des paramètres de couverture entiers.

Tableau 6.1 – Méthodes d'agrégation des données temporelles

Méthode	Courbes de demande	Réglementations
1	AV	2H
2	AV	NEAR2H
3	ABS	2H
4	ABS	NEAR2H
5	MED	2H
6	MED	NEAR2H

La figure 6.5 illustre la démarche d'évaluation d'une méthode d'agrégation. Une fois les données temporelles agrégées, le problème \bar{P} est résolu par notre procédure GenColBB. La solution du problème agrégé est un calendrier agrégé \bar{C} . Étant donné que ce calendrier n'est pas forcément réalisable pour le problème P , il faut ramener l'admissibilité des horaires le constituant afin d'avoir une base de comparaison équitable entre l'ensemble des méthodes. Un voisinage du calendrier $V(\bar{C})$ est alors construit. Ce voisinage est formé par des quarts désagrégés qui sont proches des quarts agrégés formant le calendrier \bar{C} . Finalement, le problème désagrégé restreint aux quarts voisins est résolu par notre procédure GenColBB. Nous obtenons ainsi le calendrier désagrégé C qui renseigne sur la qualité de l'agrégation. Rappelons que Z_C est la valeur de l'objectif correspondant à ce calendrier.

Reprenons les instances du JEU1 (voir Figure 2.4). Nous utilisons ces instances pour effectuer les tests quant au choix de la méthode d'agrégation.

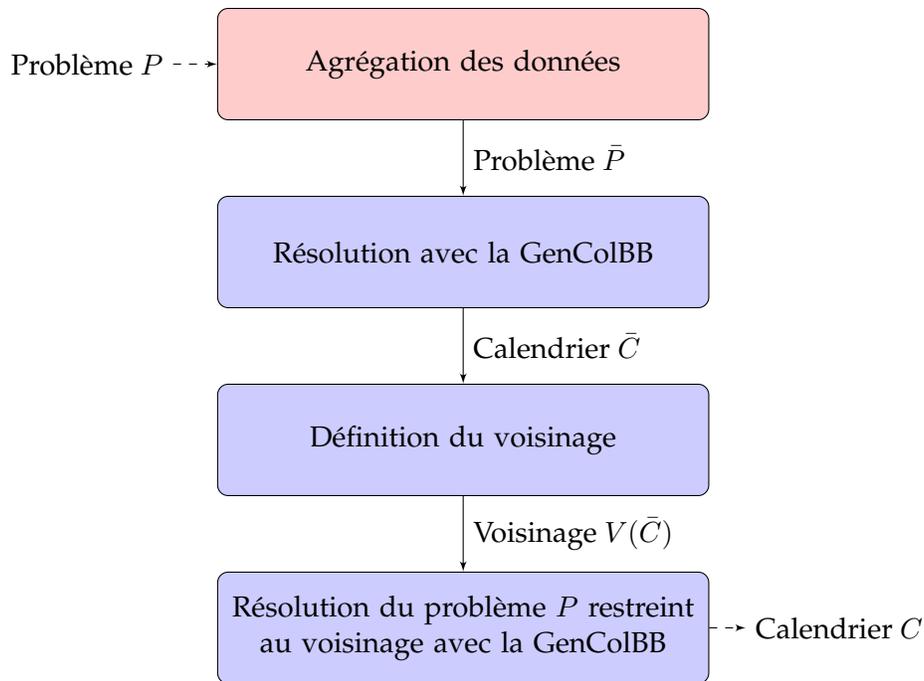


Figure 6.5 – Évaluation d’une méthode d’agrégation

Effet de l’agrégation sur la précision des courbes de demande

L’agrégation des courbes de demande influence la précision des données qui seront utilisées pour la prise de décision à l’étape d’agrégation. Cette précision dépend non seulement de la méthode d’agrégation mais aussi de la nature des courbes de demande. Rappelons que la génération des scénarios fait appel à un paramètre dont le rôle est d’ajuster la variabilité des courbes de demande. Ce paramètre correspond à l’indice du profil définissant chaque instance. Il existe quatre profils de demande : P1, P2, P3 et P4 qui correspondent à une variabilité moyenne à toutes les huit heures, à toutes les quatre heures, à toutes les deux heures et à toutes les heures respectivement.

La Figure 6.6 illustre deux scénarios qui diffèrent par le profil de demande. Dans les deux cas, nous avons choisi un département de la même organisation. Dans le premier cas, nous avons considéré sa courbe de demande correspondant au profil P1 et dans le deuxième cas, nous avons considéré sa courbe de demande correspondant au profil P4. L’axe des abscisses représente la première journée de travail et il est discrétisé en des périodes de 15 minutes. En ordonnée, nous représentons quatre courbes de demande. La première courbe appelée « *Original* » correspond à la demande réelle en effectifs pour chaque période de 15 minutes au sein du département considéré. Les trois autres courbes correspondent à la demande agrégée pour chaque période de deux heures en utilisant les trois méthodes d’agrégation des courbes de demande. Il est clair que lorsque la variabilité de la demande augmente, la perte de précision des données due à l’agrégation augmente.

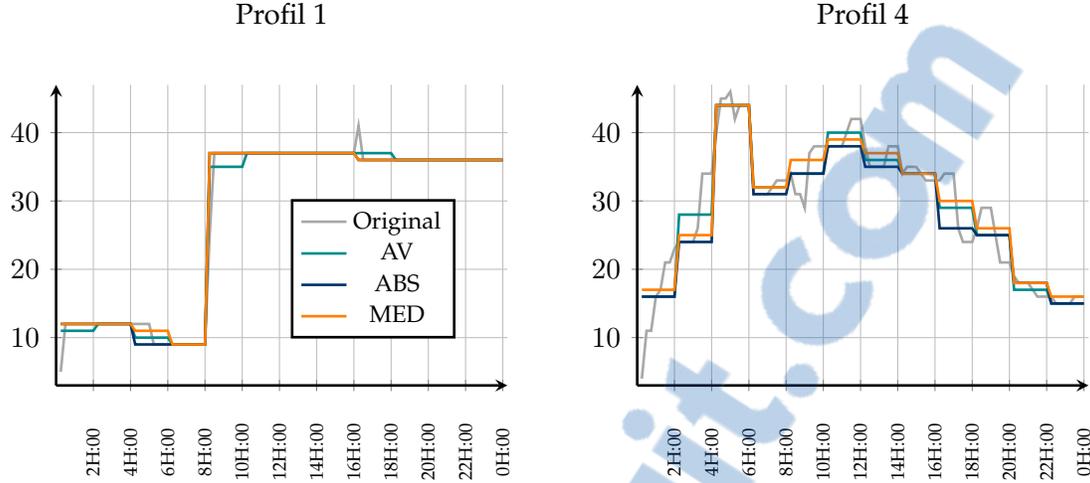


Figure 6.6 – Effet du profil et de la méthode d’agrégation sur la précision des courbes de demande

Rappelons que l’écart d’agrégation pour chaque période désagrégée $i \in f(\bar{i})$ est donné par l’expression $r_{d,\bar{i}} - r_{d,i}$. Afin de mesurer la perte de précision due à l’agrégation, nous calculons l’écart moyen (EM), l’écart moyen absolu (EMA) et le pourcentage moyen d’écart absolu (PMEA) sur l’horizon de planification et pour tous les départements de l’organisation.

Ces indicateurs s’expriment comme suit :

$$EM = \frac{1}{|D||I|} \sum_{d \in D} \sum_{\bar{i} \in \bar{I}} \sum_{i \in f(\bar{i})} (r_{d,\bar{i}} - r_{d,i}) \quad (6.6)$$

$$EMA = \frac{1}{|D||I|} \sum_{d \in D} \sum_{\bar{i} \in \bar{I}} \sum_{i \in f(\bar{i})} |r_{d,\bar{i}} - r_{d,i}| \quad (6.7)$$

$$PMEA = \frac{1}{|D||I|} \sum_{d \in D} \sum_{\bar{i} \in \bar{I}} \sum_{i \in f(\bar{i})} \frac{|r_{d,\bar{i}} - r_{d,i}|}{\max(r_{d,i}, 1)} \quad (6.8)$$

La Figure 6.7 illustre l’effet de la méthode d’agrégation et de la variabilité de la demande sur la perte de précision suite à l’agrégation. La figure inclut trois graphiques associés aux indicateurs d’écart EM, EMA et PMEA respectivement. Chaque graphique comporte trois séries de données correspondant aux trois méthodes d’agrégation des courbes de demande AV, ABS et MED. En axe des abscisses, les instances sont triées selon l’indice du profil puis selon la taille de l’organisation. En d’autres termes, nous trouvons d’abord les 12 instances correspondant au profil P1 puis les 12 instances correspondant au profil P2 et ainsi de suite. Remarquons que les profils sont triés selon l’ordre croissant de la variabilité de demande. Pour chaque profil, nous disposons de 12 instances (voir Figure 2.4).

Observons d’abord l’écart moyen d’agrégation EM. La méthode basée sur la moyenne (AV)

permet de minimiser la valeur absolue de la moyenne arithmétique des écarts d'agrégation. En d'autres termes, cette méthode donne toujours la valeur de EM la plus proche de 0. Ce résultat est prévisible de par la définition de la méthode AV. La méthode basée sur l'écart absolu (ABS) sous-estime en moyenne la demande désagrégée contrairement à la méthode basée sur la médiane (MED) qui surestime en moyenne cette demande. Cette dernière constatation découle de la nature des courbes de demande de notre jeu de données JEU1. Notons que l'écart moyen d'agrégation est beaucoup plus important pour le profil P4 caractérisé par la variabilité de demande la plus élevée par rapport au reste des profils.

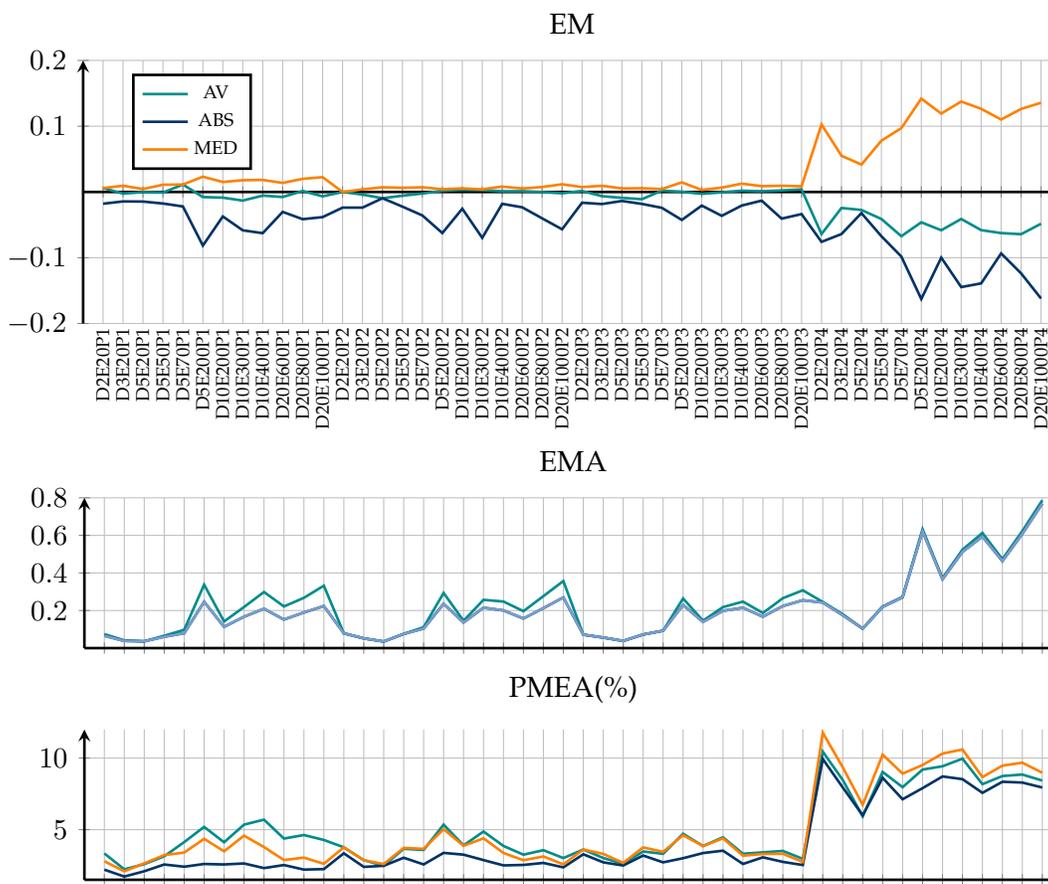


Figure 6.7 – Indicateurs d'écart par instance et par méthode d'agrégation

À l'opposé des méthodes ABS et MED qui minimisent l'amplitude moyenne des écarts d'agrégation, la méthode AV induit l'écart absolu moyen d'agrégation le plus élevé. De par la définition de la méthode ABS, il est prévisible qu'elle permet de minimiser l'indicateur EMA. Par contre, cette constatation pour la méthode MED découle uniquement de la nature des courbes de demande générées. La méthode AV affiche la pire performance. Son inconvénient majeur est le fait de sommer algébriquement les écarts négatifs et positifs d'agrégation. La surestimation de demande durant certaines périodes est compensée par une sous-estimation durant les autres périodes. Par conséquent, elle est la plus mauvaise

du point de vue de l'indicateur EMA. Tout comme l'écart moyen, l'écart moyen absolu se détériore significativement pour toutes les méthodes d'agrégation au niveau des instances correspondant au profil P4.

Finalement, c'est la méthode ABS qui minimise le pourcentage moyen d'écart absolu d'agrégation. Il n'y a pas de dominance entre les méthodes AV et MED. Cet indicateur est le plus significatif car il possède deux avantages. Contrairement à l'écart moyen qui donne une idée sur la trajectoire moyenne des écarts, l'indicateur PMEA a la vocation de tenir compte des écarts qu'ils soient positifs ou négatifs. Ainsi, il renseigne sur l'amplitude moyenne des écarts d'agrégation. En addition, contrairement à l'écart absolu moyen qui donne une idée sur l'amplitude des écarts sans tenir compte de la demande requise, l'indicateur PMEA a la vocation de donner une information sur l'erreur commise par l'agrégation relativement au niveau de la demande désagrégée. Ainsi, il renseigne sur l'amplitude relative moyenne des écarts d'agrégation exprimée en pourcentage. Encore une fois, la performance de toutes les méthodes d'agrégation se détériore pour le profil P4. Ceci dit, le pourcentage moyen d'écart absolu reste toujours inférieur à 10% pour la méthode ABS.

Effet de l'agrégation sur la précision des réglementations

Les réglementations correspondent aux règles de travail et de transfert à respecter pour construire des horaires admissibles. Le Tableau 6.2 recense les différences entre les trois types de réglementations : originales, agrégées par la méthode 2H et agrégées par la méthode NEAR2H.

Tableau 6.2 – Agrégation des réglementations

Type de réglementations	Période de discrétisation	Heures de début	Durées	B_{min}
Original	15MIN	De 0H:00 à 4H:00	De 7H à 9H	1H
		De 8H:00 à 12H:00		
		De 16H:00 à 20H:00		
2H	2H	De 0H:00 à 4H:00	6H, 8H, 10H	2H
		De 8H:00 à 12H:00		
		De 16H:00 à 20H:00		
NEAR2H	2H	De 0H:00 à 4H:00	7H, 8H, 9H	1H
		De 8H:00 à 12H:00		
		De 16H:00 à 20H:00		

Les deux méthodes d'agrégation possèdent les mêmes heures de début agrégées étant donné que les heures de début originales les plus proches de deux heures sont aussi des multiples de deux heures. Il existe neuf débuts agrégés au lieu des 51 débuts admissibles au problème original. Ainsi, l'agrégation permet d'éliminer 82.35% des heures de début diminuant ainsi la complexité des calculs. Les durées de travail agrégées diffèrent selon la méthode d'agrégation.

gation. Pour la méthode 2H, il existe deux durées qui ne respectent pas les réglementations du problème original. Ainsi, la précision des durées de la méthode NEAR2H est meilleure que celle de la méthode 2H. Notons qu’il existe également une différence très importante entre les deux méthodes d’agrégation relativement aux règles de transfert. Rappelons que la durée minimale d’un bloc de début ou de fin ne doit pas être inférieure à 1 heure pour notre jeu de données JEU1. La méthode NEAR2H permet de conserver cette précision étant donné que l’on accepte des paramètres de couverture continus. À l’opposé, la méthode 2H fixe la restriction sur la durée minimale d’un bloc à deux heures afin d’obtenir des paramètres de couverture binaires. Ainsi, la capacité du problème agrégé à détecter les transferts les plus courts est meilleure pour la méthode NEAR2H que pour la méthode 2H.

Choix de la méthode d’agrégation

Nous utilisons les 24 premières instances du JEU1 pour effectuer les tests permettant de choisir la méthode d’agrégation à incorporer dans l’heuristique de prétransfert. Chaque méthode d’agrégation est évaluée selon la démarche illustrée dans la Figure 6.5 en l’appliquant à chacune des 24 instances considérées.

Tableau 6.3 – Taille des problèmes utilisés pour le choix de la méthode d’agrégation

Organisation	Problème agrégé \bar{P}							
	# Quarts				# Quarts GenColBB			
	NEAR2H		2H		NEAR2H		2H	
	Moy.	É.T.	Moy.	É.T.	Moy.	É.T.	Moy.	É.T.
D2E20	9,198	0	7,099	0	3,103	70	3,061	64
D3E20	11,915	0	8,943	0	3,202	80	3,045	96
D5E20	22,564	0	16,181	0	3,559	246	3,383	230
D5E50	58,320	0	41,755	0	9,750	327	8,954	253
D5E70	78,831	0	56,512	0	13,304	504	12,146	217
D5E200	221,023	0	158,640	0	38,851	1,129	34,809	1,218

Organisation	Problème désagrégé P restreint au voisinage $V(\bar{C})$							
	# Quarts voisins				# Quarts voisins GenColBB			
	NEAR2H		2H		NEAR2H		2H	
	Moy.	É.T.	Moy.	É.T.	Moy.	É.T.	Moy.	É.T.
D2E20	8,973	736	6,435	966	6,582	146	5,023	345
D3E20	9,536	1,633	6,685	794	6,523	182	5,093	381
D5E20	9,611	1,024	6,514	780	6,773	195	5,171	252
D5E50	29,663	3,485	20,385	1,947	17,606	556	13,486	464
D5E70	41,198	2,994	26,613	2,531	24,687	497	18,211	687
D5E200	106,192	7,623	71,280	3,204	69,216	2,025	52,150	2,088

Le Tableau 6.3 renseigne sur la complexité des problèmes utilisés lors de l’évaluation des méthodes d’agrégation. Rappelons qu’il existe deux problèmes à résoudre : le problème agrégé

\bar{P} qui considère des données temporelles approximatives et le problème désagrégé P restreint au voisinage $V(\bar{C})$ avec \bar{C} le calendrier agrégé solution du problème \bar{P} . Chacun de ces deux problèmes a été résolu d'une manière heuristique en utilisant la procédure GenColBB (voir Section 5.2). Le tableau est divisé en deux parties : la première partie est associée au problème \bar{P} et la deuxième partie est associée au problème P restreint à $V(\bar{C})$. Les 24 instances considérées sont regroupées en six classes. Chaque classe est caractérisée par l'ensemble des départements et l'ensemble des employés qui la composent et elle contient quatre instances qui diffèrent uniquement par les profils de demande.

Les quatre premières colonnes (# Quarts et # Quarts voisins) donnent le nombre total de variables de quart des problèmes \bar{P} et P restreint à $V(\bar{C})$, respectivement, si la génération de colonnes n'est pas utilisée. Étant donné que la résolution se fait au biais de la procédure GenColBB, il y a uniquement un sous-ensemble restreint de variables de quart qui sera considéré. Les quatre dernières colonnes (# Quarts GenColBB et # Quarts voisins GenColBB) donnent le nombre de variables de quart retenues à la dernière itération de la génération de colonnes. Il s'agit donc du nombre de variables de quart du problème en nombres entiers résolu au nœud racine à l'aide de la procédure de séparation et d'évaluation progressive.

Chaque groupe de quatre colonnes (# Quarts, # Quarts GenColBB, # Quarts voisins et # Quarts voisins GenColBB) est divisé à son tour en deux parties selon la méthode d'agrégation des réglementations (NEAR2H et 2H). Pour chacune de ces méthodes et pour chaque classe d'instances, nous calculons la moyenne (Moy.) et l'écart type (É.T.) du nombre de quarts sur un total de 12 valeurs. Rappelons que chaque classe inclut quatre instances et que chaque instance est traitée à l'aide de trois méthodes d'agrégation des courbes de demande (AV, ABS et MED) pour la même méthode d'agrégation des réglementations. Ce regroupement par classe d'instances et par méthode d'agrégation des réglementations provient du fait que la taille des problèmes \bar{P} et P restreint à $V(\bar{C})$ dépend principalement de l'organisation et des règles de travail et de transfert agrégées. D'ailleurs, le nombre total de quarts agrégés admissibles du problème \bar{P} dépend uniquement de la classe et la méthode d'agrégation des réglementations (Écart type = 0).

Examinons d'abord la taille du problème agrégé \bar{P} . Le nombre de quarts admissibles définis par les règles de travail et de transfert agrégées par la méthode NEAR2H est plus élevé que celui lorsque l'agrégation se fait par la méthode 2H. Cette différence provient notamment de la restriction quant à la durée minimale d'un bloc de travail qui est plus précise lorsque nous considérons la méthode NEAR2H que la méthode 2H. La Figure 6.8 énumère les possibilités des durées des blocs de début et de fin correspondant aux quarts à deux blocs pour les deux méthodes d'agrégation des réglementations. Nous concluons que la méthode NEAR2H définit un problème agrégé incluant plus de variables que celui de la méthode 2H en permettant plus de flexibilité sur les transferts.

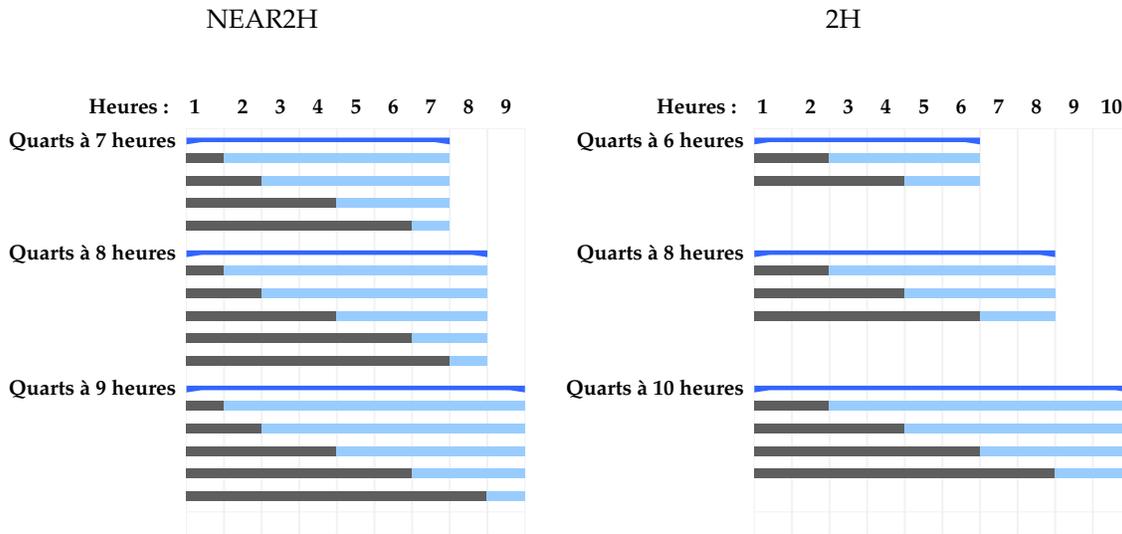


Figure 6.8 – Durées des blocs de début et de fin pour les quarts multidépartements

Observons maintenant la taille du problème désagrégé P restreint au voisinage $V(\bar{C})$. La taille du voisinage obtenu par la perturbation du calendrier agrégé défini en utilisant la méthode NEAR2H est plus grande que la taille du voisinage associé à la méthode 2H. La méthode NEAR2H induit un voisinage plus large pour deux raisons principales : le niveau de précision des durées des quarts et le niveau de flexibilité des transferts. En effet, la perturbation d'amplitude 1 heure des durées agrégées de la méthode NEAR2H permet de définir 19 durées désagrégées admissibles (entre 7H et 8H, entre 7H et 9H et entre 8H et 9H pour les durées agrégées 7H, 8H et 9H respectivement). Quant à la méthode 2H, la même perturbation permet de définir uniquement 11 durées désagrégées admissibles (7H, entre 7H et 9H et 9H pour les durées agrégées 6H, 8H et 10H respectivement). La flexibilité portant sur les durées des transferts joue aussi un rôle important influençant la taille du voisinage. Pour le prouver, nous avons extrait pour chaque instance et chaque méthode d'agrégation des courbes de demande parmi les 72 combinaisons considérées (24 instances \times 3 méthodes d'agrégation des courbes de demande) le nombre d'heures transférées dans le calendrier agrégé \bar{C} lorsque nous utilisons la méthode NEAR2H et lorsque nous utilisons la méthode 2H. Nous avons calculé pour chaque paire de valeurs l'écart exprimé en nombre d'heures entre les transferts identifiés par la méthode NEAR2H et par la méthode 2H. Dans 7% des cas, l'écart est nul. Dans 90% des cas, la méthode NEAR2H identifie plus de transferts que la méthode 2H. L'écart moyen est égal à 36.85 heures. Uniquement dans 3% des cas, l'inverse se produit avec un écart moyen de 3 heures transférées. De par la définition du voisinage, un nombre plus élevé d'heures transférées peut expliquer l'augmentation de la taille du voisinage, et plus spécifiquement l'augmentation du nombre de quarts voisins externes.

Le Tableau 6.4 permet de comparer entre les six méthodes d'agrégation en se basant sur la qualité du meilleur calendrier C trouvé dans un voisinage du calendrier agrégé \bar{C} . Les

Tableau 6.4 – Comparaison des différentes méthodes d’agrégation sur la base du coût du meilleur calendrier trouvé dans le voisinage de la solution agrégée

Instance	(AV,2H) (ABS,2H) (MED,2H) (AV,NEAR2H) (ABS,NEAR2H) (MED,NEAR2H)					
	Z_C					
D2E20P1	298.3	298.3	288.9	287.7	290.0	290.0
D2E20P2	524.3	536.5	524.3	513.7	501.1	513.7
D2E20P3	531.6	534.5	524.7	472.9	483.3	493.2
D2E20P4	455.9	432.8	451.0	416.2	405.9	461.9
D3E20P1	323.0	300.8	325.4	298.1	292.5	299.3
D3E20P2	718.6	727.5	716.0	716.3	713.7	697.2
D3E20P3	680.6	685.6	701.5	647.0	641.4	643.8
D3E20P4	704.4	750.1	761.8	728.4	709.3	681.1
D5E20P1	462.2	474.8	470.4	418.6	418.6	403.6
D5E20P2	681.7	681.7	617.3	595.2	595.2	608.8
D5E20P3	730.2	725.1	727.4	652.7	654.0	636.9
D5E20P4	843.3	861.8	883.8	803.0	813.4	801.8
D5E50P1	706.7	655.4	757.2	587.4	556.6	588.7
D5E50P2	1,181.5	1,064.0	1,118.4	989.3	968.0	993.9
D5E50P3	1,017.8	1,013.7	988.5	894.0	920.0	883.5
D5E50P4	1,127.0	1,176.0	1,170.6	960.3	1,030.8	992.6
D5E70P1	812.5	779.2	796.5	681.4	661.7	667.1
D5E70P2	1,479.8	1,457.9	1,416.4	1,332.5	1,340.3	1,306.4
D5E70P3	1,396.0	1,434.1	1,528.1	1,176.9	1,223.0	1,185.0
D5E70P4	1,374.1	1,348.9	1,301.1	1,107.8	1,036.2	1,126.2
D5E200P1	1,767.7	1,843.9	1,825.8	1,681.4	1,568.5	1,517.7
D5E200P2	2,715.1	2,680.6	2,719.4	2,532.6	2,552.2	2,443.5
D5E200P3	3,443.6	3,342.7	3,399.6	2,987.6	2,968.7	3,091.1
D5E200P4	3,412.8	3,363.6	3,430.2	3,138.0	3,032.0	3,218.8
Intervalle	% Instances : Déviation = $\frac{Z_C - \min Z_C}{\min Z_C} (\%) \in$ Intervalle					
[0, 1]	0.0	0.0	4.2	33.3	50.0	54.2
]1, 3]	0.0	4.2	4.2	37.5	20.8	16.7
]3, 5]	16.7	8.3	8.3	12.5	25.0	12.5
]5, 10]	8.3	25.0	8.3	12.5	4.2	12.5
]10, 15]	37.5	33.3	41.7	4.2	0.0	4.2
]15, 40]	37.5	29.2	33.3	0.0	0.0	0.0

colonnes correspondent aux six méthodes d’agrégation. Le tableau est divisé en deux parties. La première donne la valeur de l’objectif du calendrier C (Z_C). Les lignes correspondent aux premières 24 instances du JEU1 utilisées pour effectuer les tests quant au choix de la méthode d’agrégation. Les cellules en gras correspondent à la meilleure solution trouvée sur l’ensemble des méthodes pour chaque instance. Comme précédemment, considérons le pourcentage de déviation d’un calendrier quelconque ayant le coût Z_C par rapport au

meilleur coût trouvé $\min Z_C$ (Déviation = $\frac{Z_C - \min Z_C}{\min Z_C}$). La deuxième partie du tableau donne le pourcentage d'instances telles que cette déviation soit située dans un intervalle donné. Les intervalles et les déviations sont exprimés en pourcentage.

Il est clair que l'agrégation des réglementations en prenant les données temporelles les plus proches de deux heures (NEAR2H) domine l'agrégation des réglementations en prenant les multiples de deux heures les plus proches des données temporelles (2H). Ce constat provient du fait que la première option conserve plus de précision quant aux données initiales du problème. En effet, le problème agrégé est traité avec une connaissance plus précise des contraintes portant sur les horaires, et plus spécifiquement sur les durées des quarts et les durées des transferts. Observons maintenant l'effet de la méthode d'agrégation des courbes de demande sur la qualité du calendrier obtenu. Au total sur les 48 tests effectués en utilisant chaque méthode d'agrégation des courbes de demande, nous obtenons un calendrier qui est à 1% ou moins du meilleur calendrier trouvé dans 16.67%, 25% et 29.17% des cas pour les méthodes basées sur la moyenne, l'écart absolu et la médiane, respectivement. C'est l'agrégation basée sur la moyenne qui montre la pire performance étant donné qu'elle est sensible aux données extrêmes lorsque l'étendue de variabilité des courbes de demande est élevée. Concentrons-nous maintenant sur les méthodes basées sur l'écart absolu (ABS) et la médiane (MED) qui sont proches en termes de performances. Le pourcentage de traitements tels que la déviation soit \leq à 5% est égal à 54.17% et 50.00% pour les méthodes ABS et MED, respectivement. En d'autres termes, la méthode ABS est sensiblement meilleure que la méthode MED si nous prenons comme critère de comparaison ce pourcentage.

Le Tableau 6.5 permet de comparer les temps de calcul nécessaires pour compléter l'étape d'agrégation par les six méthodes testées. Il est composé de deux parties principales. Pour ces deux parties, les colonnes correspondent aux méthodes d'agrégation. Les lignes de la première partie du tableau correspondent aux instances. Pour chaque instance et pour chaque méthode d'agrégation, nous présentons le temps requis pour agréger les données et résoudre le problème agrégé par la procédure GenColBB. Rappelons que pour chaque instance, la résolution du problème \bar{P} obtenu par chaque méthode d'agrégation permet d'obtenir le calendrier agrégé \bar{C} . Rappelons aussi que l'évaluation de la qualité des solutions agrégées se fait selon la qualité du meilleur calendrier C trouvé dans le voisinage $V(\bar{C})$. Ainsi, une méthode d'agrégation permet d'obtenir une solution de qualité acceptable si le coût du calendrier C (Z_C) est de qualité acceptable relativement au meilleur coût trouvé par toutes les méthodes ($\min Z_C$). Supposons que cette condition est satisfaite lorsque la déviation en pourcentage entre les deux coûts ($\frac{Z_C - \min Z_C}{\min Z_C}$) reste inférieure ou égale à 5%. Les cellules en gris correspondent à des tests pour lesquels les méthodes d'agrégation réussissent à identifier une solution de qualité acceptable. Soit $\min CPU$ le meilleur temps requis pour identifier une solution de qualité relative acceptable. Les lignes de la deuxième partie du tableau correspondent à des intervalles de temps. Cette partie donne le pourcentage d'instances (parmi les

Tableau 6.5 – Comparaison des différentes méthodes d’agrégation en se basant sur le temps de calcul requis de l’étape d’agrégation

Instance	(AV,2H) (ABS,2H) (MED,2H) (AV,NEAR2H) (ABS,NEAR2H) (MED,NEAR2H)					
	CPU du problème agrégé \bar{P} (secondes)					
D2E20P1	3	4	2	3	6	5
D2E20P2	21	15	21	141	41	140
D2E20P3	8	6	209	151	83	103
D2E20P4	53	3	21	11	13	47
D3E20P1	2	4	2	16	4	27
D3E20P2	2	4	3	10	13	6
D3E20P3	6	20	12	54	83	72
D3E20P4	3	4	6	17	4	9
D5E20P1	2	2	2	22	7	43
D5E20P2	18	19	6	30	31	39
D5E20P3	5	12	5	108	133	64
D5E20P4	5	4	3	7	7	20
D5E50P1	136	23	62	2923	3600	1009
D5E50P2	331	97	57	1959	603	791
D5E50P3	1384	917	2326	3600	3600	3600
D5E50P4	82	16	69	3600	1180	3600
D5E70P1	34	33	139	3600	3600	3600
D5E70P2	109	144	58	3600	3600	3600
D5E70P3	3600	496	3600	3600	3600	3600
D5E70P4	190	95	721	3600	3600	3600
D5E200P1	1201	569	1288	3600	3600	3600
D5E200P2	3290	410	774	3601	3600	3600
D5E200P3	3600	3428	3600	3600	3600	3600
D5E200P4	2837	1589	3600	3600	3601	3600
Intervalle	%Ins. : $\frac{Z_C - \min Z_C}{\min Z_C} \leq 5\%$ et $\Delta \text{CPU} = \text{CPU} - \min \text{CPU} \in \text{Intervalle}$					
[0, 0]	12.5	4.2	12.5	41.7	58.3	37.5
]0,60]	4.2	8.3	4.2	29.2	33.3	37.5
]60,120]	0.0	0.0	0.0	8.3	4.2	4.2
]180,240]	0.0	0.0	0.0	0.0	0.0	4.2
]1320,1380]	0.0	0.0	0.0	4.2	0.0	0.0

24 considérées) pour lesquelles une méthode d’agrégation réussit à trouver une solution de qualité acceptable dans un temps CPU qui ne dépasse pas un certain intervalle par rapport au meilleur temps écoulé minCPU ($\Delta \text{CPU} = \text{CPU} - \min \text{CPU} \in \text{Intervalle}$ donné). Notons que les intervalles qui ne figurent pas dans le tableau correspondent à des pourcentages nuls pour toutes les méthodes d’agrégation.

Dans 58.3% des cas, la méthode (ABS,NEAR2H) mène à une solution de qualité relative ac-

ceptable en nécessitant le temps de calcul le moins élevé. Dans 37.5% des cas, cette méthode permet d'identifier une solution de qualité relative acceptable dans un délai supplémentaire maximal de deux minutes par rapport au meilleur temps écoulé parmi ceux menant à des solutions de qualité acceptable. En conclusion, nous sélectionnons la méthode (ABS, NEAR2H) pour l'étape d'agrégation car elle permet de réaliser un bon compromis entre perte de précision des données et complexité de calcul.

Concentrons-nous donc sur cette méthode. Observons que la limite d'une heure est atteinte pour la plupart des plus grandes instances testées. Rappelons que la procédure GenColBB définit un problème en nombres entiers restreint à un ensemble de quarts identifiés par la génération de colonnes. Dans cette procédure, ce problème en nombres entiers est résolu par le Branch and Bound de CPLEX avec un gap limite d'optimalité égal à 1%. Un temps limite d'une heure est aussi imposé sur toute la procédure GenColBB. Autrement dit, pour les instances de grande taille, CPLEX n'a pas terminé l'exploration implicite de l'arbre de branchement. Le Tableau A.1 en annexes donne les gaps d'optimalité correspondant au problème agrégé \bar{P} restreint aux quarts sélectionnés par la génération de colonnes. Le gap d'optimalité est grand dans certains cas et atteint 19.91% pour l'instance D5E200P3. Dans cette thèse, nous allons nous contenter du meilleur calendrier agrégé \bar{C} identifié par CPLEX avec les limites imposées sur le temps et le gap. Par ailleurs, les gaps d'optimalité affichés évaluent la qualité du calendrier agrégé \bar{C} par rapport aux courbes de demande agrégées.

Les détails concernant le temps de calcul et le gap d'optimalité relativement au problème désagrégé P restreint au voisinage $V(\bar{C})$ sont donnés à titre d'indication en annexes dans les tableaux A.2 et A.3 respectivement.

6.3.2 Problème désagrégé

Le niveau désagrégé exploite l'information provenant du niveau agrégé de différentes manières. Dans cette partie, nous commençons par analyser l'effet de la nature de l'information retenue sur la performance de notre heuristique de prétransfert.

Analyse des composantes du problème désagrégé

Cette partie a pour objectif d'analyser les différentes composantes du problème désagrégé et de sélectionner la meilleure configuration testée en vue de maximiser la performance de l'heuristique de prétransfert. Ce choix se fait en considérant neuf configurations possibles pour la définition et la résolution du problème désagrégé. La Figure 6.9 illustre la démarche d'évaluation d'une méthode de désagrégation. Notons que l'étape d'agrégation est réalisée avec la méthode choisie précédemment (ABS, NEAR2H). Afin de comparer les neuf configurations considérées, nous avons utilisé les 36 premières instances de notre jeu de données JEU1 (voir Figure 2.4). Ces instances sont divisées en trois catégories : les petites instances

issues des classes D2E20, D3E20 et D5E20, les moyennes instances issues des classes D5E50, D5E70 et D5E200 et les grandes instances issues des classes D10E200, D10E300 et D10E400.

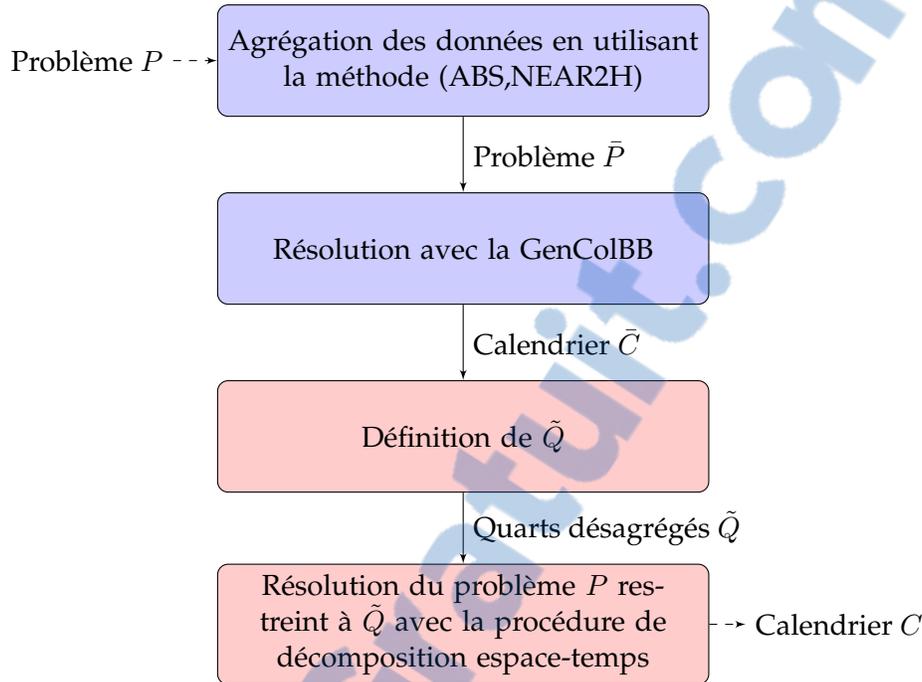


Figure 6.9 – Évaluation d’une méthode de désagrégation

Rappelons que nous avons défini dans la Section 6.2.3 un ensemble restreint de quarts désagrégés \tilde{Q} . La résolution du problème désagrégé P restreint à \tilde{Q} se fait d’une manière heuristique selon la procédure décrite à la Section 6.2. La résolution s’appuie autour d’une décomposition espace-temps du problème global. Elle utilise une procédure d’horizon roulant pour identifier une suite de fenêtres de temps respectant chacune la restriction sur la durée totale minimale. Pour chaque fenêtre de temps w , le problème P restreint à w est décomposé à son tour en un ordonnancement de sous-problèmes associé chacun à un département d . Ainsi, pour chaque fenêtre w et chaque département d , nous résolvons un problème désagrégé $P(w, d)$ restreint à un ensemble de quarts $\tilde{Q}(w, d) \subseteq \tilde{Q}$ permettant de couvrir la demande du département d au niveau de la fenêtre w en utilisant les employés internes et un ensemble spécifique des employés externes de d . Le choix des fenêtres de temps, l’ordre de traitement des départements et l’ensemble spécifique des employés externes est dicté par un ensemble de règles provenant du graphe de transfert G_I^{tr} déduit du calendrier \bar{C} .

La résolution du problème désagrégé P restreint à \tilde{Q} fait appel à l’ensemble des paramètres suivants :

- Durée minimale des fenêtres de temps : Rappelons que ce paramètre est noté LW_{min} et que sa valeur minimale est égale à 1 jour. Nous considérons les trois valeurs suivantes : 1 jour, 3 jours et 7 jours. W1, W3 et W7 correspondent respectivement à ces

trois options. Notons que la troisième option correspond à éliminer la décomposition temporelle du problème car notre jeu de données considère un horizon d'une semaine.

- Début et durées de travail : La construction de l'ensemble des quarts vides Q_{vides} se fait selon deux manières différentes : « V » et « A ». L'option « V » pour dire « voisins » utilise le calendrier agrégé \bar{C} et énumère l'ensemble des heures de début et des durées voisins avec une amplitude de perturbation de deux heures. L'option « A » pour dire « admissibles » énumère tous les débuts et les durées admissibles selon les types de quart T du problème global, et ce indépendamment de l'information provenant du niveau agrégé.
- Début et durées de transfert : La construction de l'ensemble des blocs anonymes désagrégés $B^{anonymes}$ se fait soit en gardant exactement les mêmes blocs externes figurant dans le calendrier agrégé \bar{C} ou en perturbant les débuts et les durées de ces blocs externes avec une amplitude d'une heure. La première option sera notée « F » pour dire « fixés » et la deuxième sera notée « P » pour dire « perturbés ».
- Identité des transferts : Rappelons que l'ensemble B correspond aux blocs personnalisés désagrégés figurant dans l'ensemble des quarts externes \tilde{Q}^{ext} . Lors de l'affectation des blocs désagrégés anonymes de l'ensemble $B^{anonymes}$ aux employés, nous considérons deux options selon l'information que l'on retient du calendrier agrégé \bar{C} . Dans un premier temps, nous affectons les blocs désagrégés anonymes aux employés transférés dans \bar{C} durant les plages horaires correspondant à ces blocs obtenant ainsi l'option « E » pour dire « employé ». Dans un deuxième temps, nous perturbons l'identité des employés en l'élargissant pour couvrir l'ensemble des employés admissibles provenant du département donneur dans \bar{C} obtenant ainsi l'option « D » pour dire « département ».
- Procédure de résolution des sous-problèmes $P(w, d)$: Nous avons comparé entre les deux options sans et avec la génération de colonnes désignées par « B » et « G » pour dire « BB » et « GenColBB » respectivement. Le BB correspond à la procédure de séparation et d'évaluation progressive de CPLEX avec un nouveau temps limite de 10 minutes et un gap d'optimalité limite de 1%. CPLEX est utilisé pour résoudre le problème en nombres entiers au nœud racine avec un temps limite de 10 minutes et un gap d'optimalité limite de 1%.

Chaque configuration testée pour résoudre le problème désagrégé P restreint à \tilde{Q} est donnée par le quintuplet (R1,R2,R3,R4,R5) avec R1, R2, R3, R4 et R5 les valeurs associées respectivement aux options envisagées pour les composantes LW_{min} , Q_{vides} , $B^{anonymes}$, B et la procédure de résolution des sous-problèmes.

Le plan d'expériences complet contient 3×2^4 configurations possibles. Nous avons choisi uniquement neuf configurations parmi les 48 possibles (voir Tableau 6.6). Nous avons commencé par la combinaison 1 la plus simple (W7,V,F,E,B) qui vérifie les trois conditions sui-

Tableau 6.6 – Configurations testées pour la résolution du problème désagrégé P restreint à \tilde{Q}

N°	Désignation	Durée min. des fenêtres	Débuts et durées de travail	Débuts et durées de transfert	Identité des transferts	GenColBB ou BB
1	(W7,V,F,E,B)	Horizon	Voisins	Fixés	Employé	BB
2	(W7,V,P,E,B)	Horizon	Voisins	Perturbés	Employé	BB
3	(W7,V,P,E,G)	Horizon	Voisins	Perturbés	Employé	GenColBB
4	(W7,A,P,E,B)	Horizon	Admissibles	Perturbés	Employé	BB
5	(W1,A,F,E,B)	1 jour	Admissibles	Fixés	Employé	BB
6	(W1,A,P,E,G)	1 jour	Admissibles	Perturbés	Employé	GenColBB
7	(W1,V,P,D,B)	1 jour	Voisins	Perturbés	Dép. donneur	BB
8	(W3,V,P,D,B)	3 jours	Voisins	Perturbés	Dép. donneur	BB
9	(W3,A,P,D,B)	3 jours	Admissibles	Perturbés	Dép. donneur	BB

vantes : elle exclut la décomposition temporelle du problème désagrégé P , elle est la plus fidèle au calendrier agrégé \bar{C} lors de la définition des quarts désagrégés \tilde{Q} et elle utilise le Branch and Bound de CPLEX pour la résolution des sous-problèmes $P(\bar{I}, d)$. Ensuite, nous avons élargi l'ensemble des quarts désagrégés \tilde{Q} en perturbant les débuts et les durées des blocs externes pour former la 2^{ème} combinaison (W7,V,P,E,B). Nous avons remarqué que dans 69% des cas, l'utilisation des blocs externes perturbés donne des solutions de meilleure qualité que celles obtenues par des blocs externes fixés. Par conséquent, nous avons décidé de garder l'option « perturbés » et de définir une 3^{ème} combinaison qui diffère uniquement de la 2^{ème} par la procédure de résolution des sous-problèmes. Au lieu d'utiliser le BB, nous avons utilisé la GenColBB pour résoudre les sous-problèmes définis sur l'horizon de planification et restreints aux départements de l'organisation. Cette 3^{ème} combinaison (W7,V,P,E,G) détériore légèrement la qualité des solutions obtenues dans 72% des cas en moyenne de 3% par rapport à la 2^{ème} combinaison (W7,V,P,E,B).

Nous avons donc opté pour l'option « BB » et nous avons décidé d'élargir l'ensemble des heures de début et des durées définissant les quarts désagrégés \tilde{Q} en énumérant tous les quarts vides admissibles au lieu de nous contenter des quarts vides voisins. Pour cette raison, nous avons testé la 4^{ème} configuration (W7,A,P,E,B) et nous l'avons comparée à la 2^{ème} combinaison (W7,V,P,E,B) qui a permis d'identifier les meilleures solutions jusque-là. Nous avons remarqué que la 4^{ème} combinaison (W7,A,P,E,B) domine la 2^{ème} combinaison testée (W7,V,P,E,B) en termes de qualité des solutions obtenues dans 83% des cas pour les organisations les plus petites. La 2^{ème} combinaison (W7,V,P,E,B) devient meilleure dans 100% des cas pour les organisations les plus grandes. Ce résultat s'explique par la taille des sous-problèmes $P(\bar{I}, d)$ qui devient plus grande dans le cas de l'énumération exhaustive de tous les débuts et les durées admissibles. Cette taille devient un obstacle à la résolution efficace en utilisant le Branch and Bound de CPLEX. D'ailleurs, le gap moyen d'optimalité grimpe à 21% en moyenne sur les 12 grandes instances. Cette qualité médiocre de résolution des



sous-problèmes détériore la performance de la 4^{ème} combinaison (W7,A,P,E,B) en faveur de la 2^{ème} combinaison (W7,V,P,E,B).

Afin d'explorer davantage l'option « admissibles », nous avons décidé de restreindre la taille des sous-problèmes $P(w, d)$ en considérant des fenêtres de temps de durée minimale égale à 1 jour et des blocs de transfert fixés obtenant ainsi la 5^{ème} combinaison (W1,A,F,E,B). Le nombre moyen de fenêtres de temps est égal à 4.1 pour les 12 petites instances, 8.1 pour les 12 moyennes instances et 8.7 pour les 12 grandes instances ce qui prouve que le modèle d'élimination de circuits (EC) a été utilisé afin de respecter la restriction sur la durée minimale des fenêtres décomposant ainsi le problème selon la composante temporelle. Nous avons remarqué que la combinaison 5 conduit à des solutions de qualité moindre pour les petites instances et de qualité meilleure pour les grandes instances que celles de la 4^{ème} combinaison (W7,A,P,E,B). Par contre, la 2^{ème} combinaison (W7,V,P,E,B) demeure meilleure que la 5^{ème} pour la totalité des 12 grandes instances. Ce résultat ne provient pas des gaps d'optimalité comme précédemment pour la 4^{ème} combinaison (W7,A,P,E,B) car le gap moyen d'optimalité correspondant à la 5^{ème} combinaison (W1,A,F,E,B) et aux 12 grandes instances est égal à 0.1%. Ceci signifie que la décomposition temporelle du problème est à l'origine de la mauvaise performance de la 5^{ème} combinaison (W1,A,F,E,B). En fait, les contraintes horizontales quant à la durée maximale travaillée durant l'horizon et le pourcentage minimal de non transfert sont mieux gérées par la méthode de désagrégation lorsque l'horizon est considéré dans son ensemble que lorsqu'il est divisé en des fenêtres. Rappelons que le traitement du problème durant l'horizon en entier « W7 » provoque l'élimination de certains transferts entre les départements. Cette élimination est moins importante lorsque nous considérons une restriction sur la durée des fenêtres de temps de 1 jour. Il est donc question de compromis entre la gestion des contraintes horizontales et la conservation de l'information sur les transferts agrégés. La procédure d'horizon roulant, en présence de contraintes horizontales, tend à favoriser le début de l'horizon au détriment de la fin.

La combinaison 6 est obtenue à partir de la combinaison 5 en changeant l'option des transferts « fixés » par des transferts « perturbés » et l'option « BB » par l'option « GenColBB ». Ce choix permet de tester si la détérioration de la décomposition temporelle du problème peut être rattrapée par une définition plus large de quarts externes \tilde{Q}^{ext} en perturbant les heures de début et les durées des blocs de transfert. Ceci dit, pour éviter l'obtention de sous-problèmes $P(w, d)$ très larges contenant un nombre élevé de quarts externes, nous avons eu recours à la procédure GenColBB au lieu du BB. Quoique l'option « GenColBB » a diminué légèrement la qualité des solutions dans le cas de la combinaison 3, son utilisation peut être pertinente lorsque la flexibilité sur les heures de début et les durées devient plus importante que celle de la combinaison 3. La combinaison 6 a été meilleure que la combinaison 5 dans 69% des cas avec une amélioration moyenne de 7% de la valeur de l'objectif. Une fois de plus, nous confirmons que l'option des transferts « perturbés » est meilleure que l'option

des transferts « fixés ». Ceci dit, les inconvénients provenant de la décomposition temporelle persistent pour les grandes instances car la combinaison 2 reste la meilleure. Par contre, la combinaison 6 est meilleure que la combinaison 2 pour les petites instances. En effet, le nombre moyen des fenêtres de temps pour les petites instances est égal 4.1 alors qu'il est égal à 8.7 pour les grandes. Les effets négatifs de la décomposition temporelle sont donc atténués pour les petites instances.

Les trois dernières combinaisons diffèrent des six premières par l'identité des transferts. L'option « département donneur » remplace l'option « employé ». Uniquement les restrictions sur la durée des fenêtres de temps égales à 1 et 3 jours sont considérées. Ce choix permet d'éviter l'obtention de sous-problèmes $P(w, d)$ qui sont très larges et ne peuvent pas être résolus efficacement dans la limite de temps de 10 minutes. La résolution des sous-problèmes se fait par le BB dans tous les cas pour maximiser les chances d'identifier les meilleures solutions possibles aux sous-problèmes. Dans un premier temps, nous considérons des quarts vides voisins et des transferts perturbés. La seule différence entre les combinaisons 7 et 8 réside dans la décomposition temporelle du problème. Alors que le nombre moyen des fenêtres de temps pour la combinaison 7 associée à W1 est égal à 7, la combinaison 8 associée à W3 fait intervenir uniquement en moyenne 2.8 fenêtres de temps. Comme attendu, la combinaison 8 qui implique un nombre moins élevé de fenêtres de temps conduit à de meilleures solutions que celles identifiées par la combinaison 7 dans 58% des cas. Nous obtenons la même valeur de l'objectif dans 8% des cas. Afin de voir l'effet d'élargir davantage la taille de l'ensemble des quarts désagrégés \tilde{Q} , nous avons modifié la 8^{ème} combinaison (W3,V,P,D,B) par la 9^{ème} combinaison (W3,A,P,D,B). Observons que l'option « voisins » est remplacée par l'option « admissibles ». Pour les 12 petites instances, cette modification permet d'améliorer la valeur de l'objectif de 7.3% en moyenne dans 67% des cas et de conserver la même qualité dans 8% des cas. Pour les grandes instances, la tendance s'inverse. En effet, cette modification augmente la valeur de l'objectif en moyenne de 15.1% dans 83% des cas. La détérioration provient de la difficulté à résoudre efficacement certains sous-problèmes par le BB. Quoique le gap moyen d'optimalité soit égal à 2.2%, la moyenne des 12 gaps maximaux est égale à 33.6%.

La Figure 6.10 donne le nombre de quarts désagrégés formant l'ensemble \tilde{Q} pour chaque méthode de désagrégation. En axe des abscisses, nous représentons les 36 premières classes d'instances du JEU1. En axe des ordonnées, nous considérons six séries de données correspondant aux neuf méthodes de désagrégation. Les séries de données triées selon un ordre croissant de la cardinalité de l'ensemble \tilde{Q} sont données comme suit : (V,F,E), (V,P,E), (V,P,D),(A,F,E),(A,P,E) et finalement (A,P,D).

La Figure 6.11 donne le nombre moyen de variables de décision des sous-problèmes $P(w, d)$ restreints aux sous-ensembles $\tilde{Q}(w, d)$. L'annexe A.2 donne le nombre moyen, minimal et maximal de variables de décision des sous-problèmes (voir Tableaux A.4, A.5 et A.6 res-

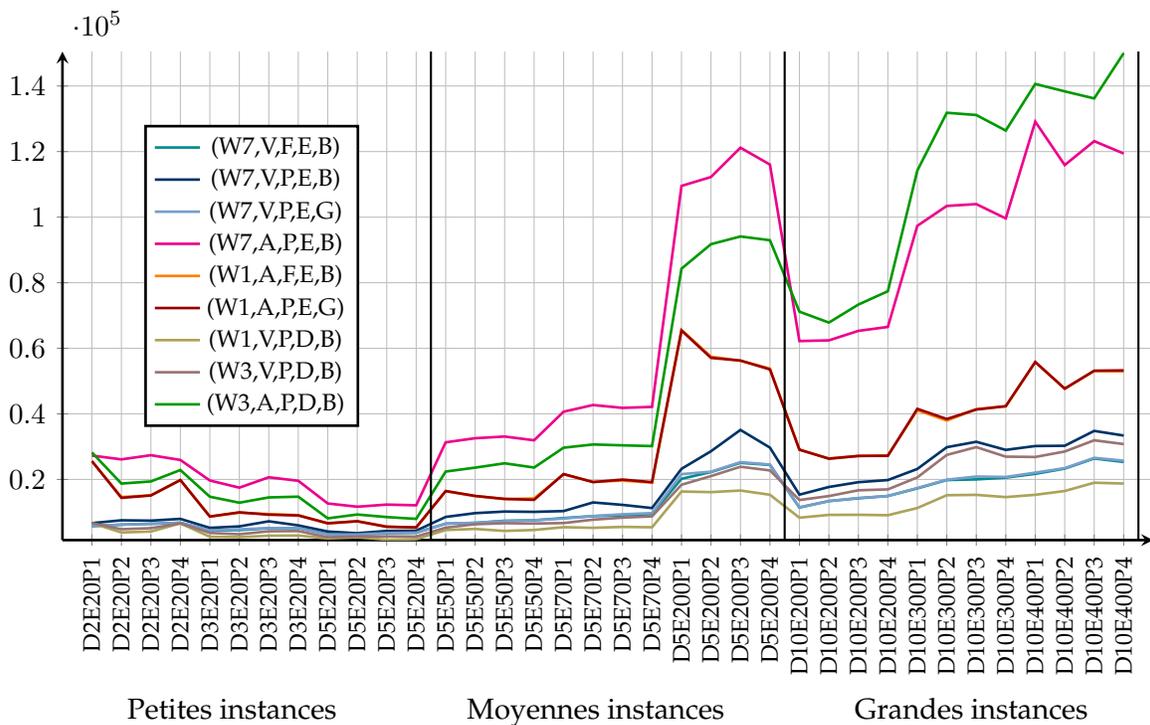
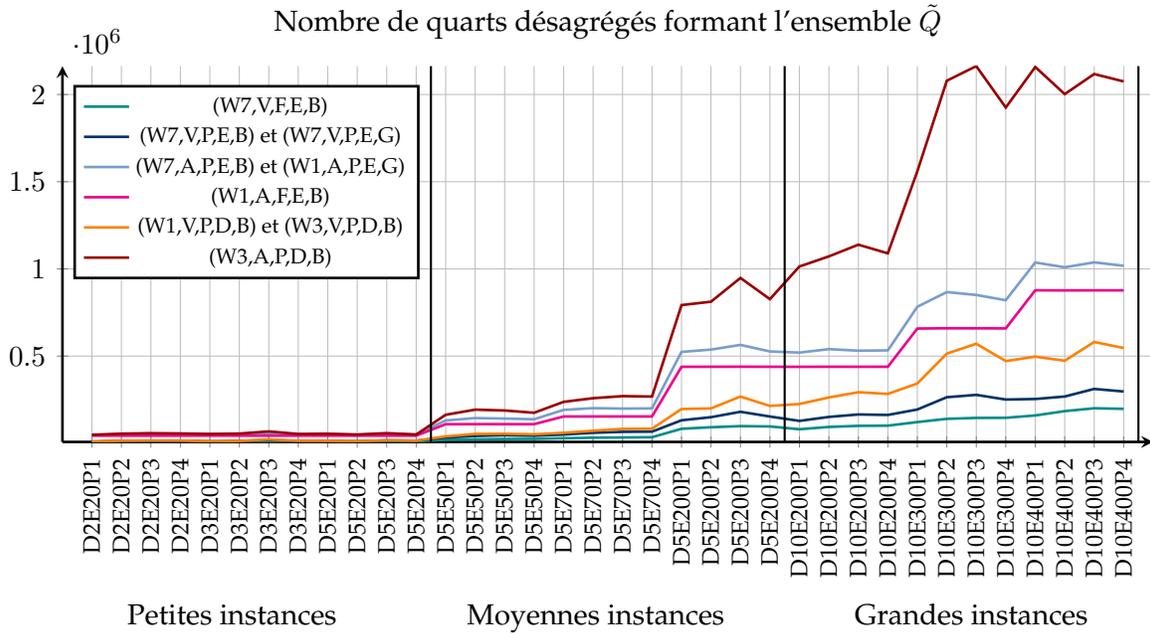


Figure 6.11 – Nombre moyen de variables de décision des sous-problèmes $P(w, d)$ restreints aux sous-ensembles $\tilde{Q}(w, d)$

pectivement). L'axe des abscisses représente les 36 instances considérées pour le choix de la méthode de désagrégation. L'axe des ordonnées correspond au nombre de variables de dé-

cision. Nous considérons neuf séries de données associées aux méthodes de désagrégation.

Tableau 6.7 – Comparaison des différentes méthodes de désagrégation en se basant sur la qualité des solutions obtenues

Intervalle	(W7,V,FE,B)	(W7,V,PE,B)	(W7,V,PE,G)	(W7,A,PE,B)	(W1,A,FE,B)	(W1,A,PE,G)	(W1,V,PD,B)	(W3,V,PD,B)	(W3,A,PD,B)
Petites classes									
≤ 1	0.00	8.33	0.00	8.33	0.00	16.67	50.00	33.33	66.67
]1,2]	0.00	0.00	8.33	8.33	8.33	8.33	0.00	8.33	8.33
]2,3]	0.00	0.00	0.00	0.00	0.00	16.67	0.00	0.00	16.67
]3,4]	0.00	0.00	0.00	0.00	8.33	0.00	8.33	8.33	8.33
]4,5]	0.00	0.00	0.00	0.00	0.00	0.00	8.33	0.00	0.00
≤ 5 (Cumul)	0.00	8.33	8.33	16.67	16.67	41.67	66.67	50.00	100.00
]5,10]	25.00	25.00	16.67	33.33	33.33	16.67	0.00	16.67	0.00
]10,15]	16.67	8.33	8.33	16.67	25.00	8.33	25.00	25.00	0.00
]15,20]	16.67	16.67	25.00	0.00	0.00	8.33	0.00	0.00	0.00
> 20	41.67	41.67	41.67	33.33	25.00	25.00	8.33	8.33	0.00
Moyennes classes									
≤ 1	8.33	16.67	16.67	33.33	0.00	0.00	33.33	0.00	8.33
]1,2]	0.00	0.00	0.00	0.00	0.00	0.00	8.33	0.00	0.00
]2,3]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
]3,4]	0.00	0.00	8.33	0.00	0.00	0.00	0.00	8.33	0.00
]4,5]	0.00	0.00	0.00	8.33	8.33	0.00	8.33	8.33	0.00
≤ 5 (Cumul)	8.33	16.67	25.00	41.67	8.33	0.00	50.00	16.67	8.33
]5,10]	25.00	16.67	25.00	0.00	0.00	0.00	25.00	33.33	25.00
]10,15]	8.33	8.33	8.33	8.33	16.67	16.67	8.33	8.33	0.00
]15,20]	0.00	8.33	0.00	8.33	8.33	25.00	0.00	8.33	0.00
> 20	58.33	50.00	41.67	41.67	66.67	58.33	16.67	33.33	66.67
Grandes classes									
≤ 1	0.00	75.00	41.67	0.00	0.00	0.00	0.00	8.33	8.33
]1,2]	0.00	8.33	0.00	0.00	0.00	0.00	0.00	8.33	0.00
]2,3]	8.33	8.33	8.33	0.00	0.00	0.00	0.00	0.00	0.00
]3,4]	8.33	0.00	16.67	0.00	0.00	0.00	0.00	25.00	0.00
]4,5]	0.00	0.00	16.67	0.00	0.00	0.00	0.00	8.33	8.33
≤ 5 (Cumul)	16.67	91.67	83.33	0.00	0.00	0.00	0.00	50.00	16.67
]5,10]	25.00	8.33	16.67	8.33	0.00	8.33	50.00	25.00	33.33
]10,15]	41.67	0.00	0.00	0.00	0.00	0.00	16.67	8.33	0.00
]15,20]	0.00	0.00	0.00	0.00	8.33	8.33	16.67	0.00	8.33
> 20	16.67	0.00	0.00	91.67	91.67	83.33	16.67	16.67	41.67

Le Tableau 6.7 compare entre la valeur de l'objectif des solutions calculées à l'aide des neuf méthodes de désagrégation. Le tableau est divisé en trois parties qui correspondent aux petites, moyennes et grandes instances. Les colonnes du tableau correspondent aux configurations de la procédure de décomposition espace-temps. Les lignes correspondent à des intervalles délimitant la valeur de la déviation par rapport à la meilleure solution trouvée. Pour chaque instance, nous calculons le coût du meilleur calendrier trouvé par chaque configuration (Z_C) et nous déduisons le meilleur coût parmi les neuf identifiés ($\min Z_C$). Ensuite, nous

calculons pour chaque instance et chaque configuration la déviation relative $(\frac{Z_C - \min Z_C}{\min Z_C})$ exprimée en pourcentage. Pour chaque catégorie d’instances (petites, moyennes et grandes) et chaque intervalle, nous calculons le nombre total d’instances qui possèdent une déviation relative appartenant à l’intervalle considéré. Finalement, nous calculons un pourcentage d’instances par rapport au total des 12 instances composant chaque catégorie.

Prenons comme critère de sélection de la méthode de désagrégation le pourcentage d’instances telles que la déviation par rapport à la meilleure solution trouvée est au plus égale à 5%. Nous identifions en gras le pourcentage le plus élevé parmi toutes les configurations pour chaque catégorie d’instances. Pour les petites classes, la configuration (W3,A,P,D,B) réussit à identifier pour toutes les 12 instances de cette catégorie une solution de bonne qualité relative. La déviation maximale n’excède pas 4%. À l’opposé, la méthode (W7,V,F,E,B) ne réussit à identifier aucune solution de qualité relative acceptable. Elle affiche la pire performance. Regardons le Tableau A.4 qui donne le nombre moyen de variables de décision des sous-problèmes $P(w, d)$ restreints aux sous-ensembles $\tilde{Q}(w, d)$. La meilleure configuration (W3,A,P,D,B) implique un nombre moyen de variables entre 7,975 et 28,231 alors que la pire configuration (W7,V,F,E,B) implique un nombre moyen de variables beaucoup moins élevé variant entre 3,190 et 6,781. Il est donc normal d’expliquer la différence de performance par le nombre de quarts désagrégés explorés par les deux méthodes de désagrégation. La méthode qui explore le plus de quarts est celle qui obtient des calendriers de meilleur coût. Pour les classes moyennes, la meilleure configuration correspond à (W1,V,P,D,B) qui réussit à identifier une solution de qualité relative acceptable dans 50% des cas et une solution ayant une déviation relative $\leq 10\%$ dans 75% des cas. En effet, à partir d’un certain nombre moyen de variables de décision, la configuration (W3,A,P,D,B) qui était auparavant la meilleure, perd sa performance à cause de la difficulté de résolution des sous-problèmes $P(w, d)$. Pour chaque instance et chaque méthode de désagrégation, nous avons retenu le gap d’optimalité maximal et moyen sur l’ensemble des sous-problèmes $P(w, d)$ résolus par la procédure de décomposition espace-temps. Quoique le gap moyen d’optimalité pour la configuration (W3,A,P,D,B) soit entre 0% et 3.7%, le gap maximum atteint une valeur maximale de 35.9% (voir Tableaux A.8 et A.10 en annexe A.2). En d’autres termes, les sous-problèmes deviennent grands et la résolution par le Branch and Bound de CPLEX à l’intérieur d’un intervalle de temps de 10 minutes peut entraîner dans certains cas des solutions de mauvaise qualité ce qui limite la performance des configurations définissant des ensembles $Q(w, d)$ de grande taille. La meilleure configuration pour la catégorie des classes moyennes qui est (W1,V,P,D,B) contourne le problème de la complexité des sous-problèmes $P(w, d)$ par l’exploration de fenêtres de temps relativement courtes découlant de la restriction minimale égale à 1 jour. La Figure 6.11 montre clairement que cette configuration définit des sous-problèmes de taille moindre. Finalement, la deuxième configuration (W7,V,P,E,B) domine l’ensemble des méthodes de désagrégation pour le cas des grandes instances. En effet, elle réussit à identi-

fier des solutions de bonne qualité relative dans 91.67% des cas avec une déviation $\leq 5\%$. Cette deuxième configuration a la vocation d'éviter la décomposition temporelle du problème mais limite la taille de l'ensemble \tilde{Q} en restant autour d'un voisinage des quarts vides agrégés et en conservant l'identité des employés transférés dans le calendrier \tilde{C} .

Notons que le point commun entre les trois méthodes les plus performantes est d'identifier des solutions avec un nombre d'heures transférées important comparé aux autres configurations. Quoique ce nombre ne soit pas le plus élevé pour toutes les instances, il fait tout le temps partie de ceux les plus élevés sur l'ensemble des neuf configurations testées (voir Tableau A.11 en annexe A.2).

Tableau 6.8 – Répartition des temps de calcul des méthodes de désagrégation

Intervalle	% d'ins. tel que Déviation $\leq 5\%$ et $\Delta\text{CPU} \in$ Intervalle (Déviation moyen en %)								
(secondes)	(W7,V,FE,B)	(W7,V,PE,B)	(W7,V,PE,G)	(W7,A,PE,B)	(W1,A,FE,B)	(W1,A,PE,G)	(W1,V,PD,B)	(W3,V,PD,B)	(W3,A,PD,B)
Petites classes									
[0,0]	0.0	8.3(0.8)	0.0	8.3(1.6)	0.0	0.0	58.3(1.2)	8.3(1.0)	16.7(0.0)
]0,60]	0.0	0.0	8.3(1.2)	0.0	8.3(3.4)	25(1.0)	8.3(0.8)	41.7(0.9)	16.7(0.0)
]60,120]	0.0	0.0	0.0	8.3(0.4)	8.3(1.0)	8.3(2.1)	0.0	0.0	16.7(0.4)
]120,300]	0.0	0.0	0.0	0.0	0.0	8.3(1.8)	0.0	0.0	25(0.8)
]420,720]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	25(2.5)
Moyennes classes(%)									
[0,0]	0.0	16.7(0.0)	0.0	16.7(0.0)	8.3(4.7)	0.0	41.7(0.3)	16.7(4.4)	0.0
]0,60]	0.0	0.0	16.7(0.2)	0.0	0.0	0.0	8.3(4.2)	0.0	0.0
]120,180]	0.0	0.0	0.0	8.3(4.3)	0.0	0.0	0.0	0.0	0.0
]180,720]	0.0	0.0	0.0	16.7(0.4)	0.0	0.0	0.0	0.0	0.0
]1080,1500]	8.3(0.0)	0.0	8.3(3.0)	0.0	0.0	0.0	0.0	0.0	8.3(0.0)
Grandes classes (%)									
[0,0]	8.3(2.7)	41.7(0.6)	8.3(0.0)	0.0	0.0	0.0	0.0	41.7(2.5)	0.0
]0,60]	0.0	8.3(1.4)	16.7(1.8)	0.0	0.0	0.0	0.0	8.3(3.2)	0.0
]60,120]	0.0	8.3(0.0)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
]120,300]	0.0	25(0.0)	25(3.7)	0.0	0.0	0.0	0.0	0.0	0.0
]300,660]	8.3(3.7)	8.3(0.0)	16.7(2.0)	0.0	0.0	0.0	0.0	0.0	0.0
]720,900]	0.0	0.0	16.7(0.7)	0.0	0.0	0.0	0.0	0.0	0.0
]2460,6900]	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.7(2.2)

Le Tableau 6.8 analyse la répartition des temps de calcul des neuf méthodes de désagrégation. Les colonnes correspondent aux différentes configurations de la procédure de décomposition espace-temps. Pour chaque instance, nous identifions les méthodes menant à une solution de déviation $\leq 5\%$. Rappelons que la déviation est calculée comme étant l'écart en pourcentage entre le coût du calendrier identifié par chaque méthode et le meilleur coût de tous les calendriers identifiés par l'ensemble des méthodes. Lorsque la déviation est $\leq 5\%$ le calendrier est considéré de qualité relative acceptable. Nous cherchons donc le temps de

calcul minimal sur l'ensemble des méthodes qui ont réussi à trouver un calendrier de qualité acceptable. Notons $\min \text{CPU}$ ce temps minimal de résolution. Ensuite, nous calculons pour chacune des méthodes qui ont bien performé l'écart entre le temps de résolution de la méthode (CPU) et le meilleur temps de résolution ($\min \text{CPU}$). Cet écart s'exprime donc comme suit : $\Delta \text{CPU} = \text{CPU} - \min \text{CPU}$. Enfin, nous regroupons pour chaque méthode les instances résolues avec une qualité acceptable dans un temps CPU tel que l'écart ΔCPU appartienne à un intervalle donné. Les lignes du tableau correspondent à ces intervalles. Pour chaque catégorie d'instances, nous calculons la proportion par rapport aux 12 instances appartenant à cette catégorie et nous indiquons entre parenthèse la déviation moyenne en termes de qualité de solution. Par exemple, la configuration 2 (W7,V,P,E,B) trouve dans 41.7% des cas une solution acceptable pour des instances de grande taille dans le meilleur temps de résolution. De plus, la déviation moyenne correspondante à ces instances est égale à 0.6%. Notons que les chiffres en gras correspondent aux méthodes qui ont performé le mieux en termes de qualité de solution.

Pour les petites classes, bien que la méthode (W3,A,P,D,B) identifie les calendriers les moins coûteux, elle requiert dans certains cas un temps relativement élevé par rapport aux autres méthodes identifiant des calendriers de coût acceptable. La configuration 7 (W1,V,P,D,B) calcule dans 58.3% des cas des solutions de déviation moyenne égale à 1.2% dans un temps de résolution minimal offrant ainsi un bon compromis entre la qualité et le temps pour la moitié des instances de petite taille. Au contraire, les configurations 7 (W1,V,P,D,B) et 2 (W7,V,P,E,B) dominent non seulement en termes de qualité de la solution mais aussi en termes de temps de résolution pour les moyennes et grandes instances, respectivement. Pour plus de détails, le Tableau A.12 en annexe donne les temps de calcul requis des méthodes de désagrégation en secondes. Le temps de calcul est composé essentiellement de trois termes : le temps d'énumération des quarts désagrégés formant l'ensemble \tilde{Q} , le temps d'énumération des circuits élémentaires et de résolution du modèle EC pour l'élimination de ces circuits et le temps de résolution des problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$ à l'aide du BB ou de la GenColBB. Les Tableaux A.13, A.14 et A.15 donnent respectivement ces trois termes.

En conclusion, nous retenons la configuration 9 (W3,A,P,D,B) pour les petites instances, la configuration 7 (W1,V,P,D,B) pour les moyennes instances et la configuration 2 (W7,V,P,E,B) pour les grandes instances. Nous décidons également de garder la configuration 2 pour les instances de très grande taille issues des classes D20E600, D20E800 et D20E1000 étant donné que l'on observe que cette configuration devient meilleure lorsque la taille des sous-problèmes augmente avec la complexité des instances.

Comparaison entre la procédure de décomposition espace-temps et la procédure de GenColBB

Dans cette partie, nous désignons la procédure de décomposition espace-temps par « *DecompET* ». De plus, nous considérons l'intégralité des 48 instances formant le JEU1 (voir Figure 2.4). Ces instances seront divisées en petites, moyennes, grandes et très grandes organisations.

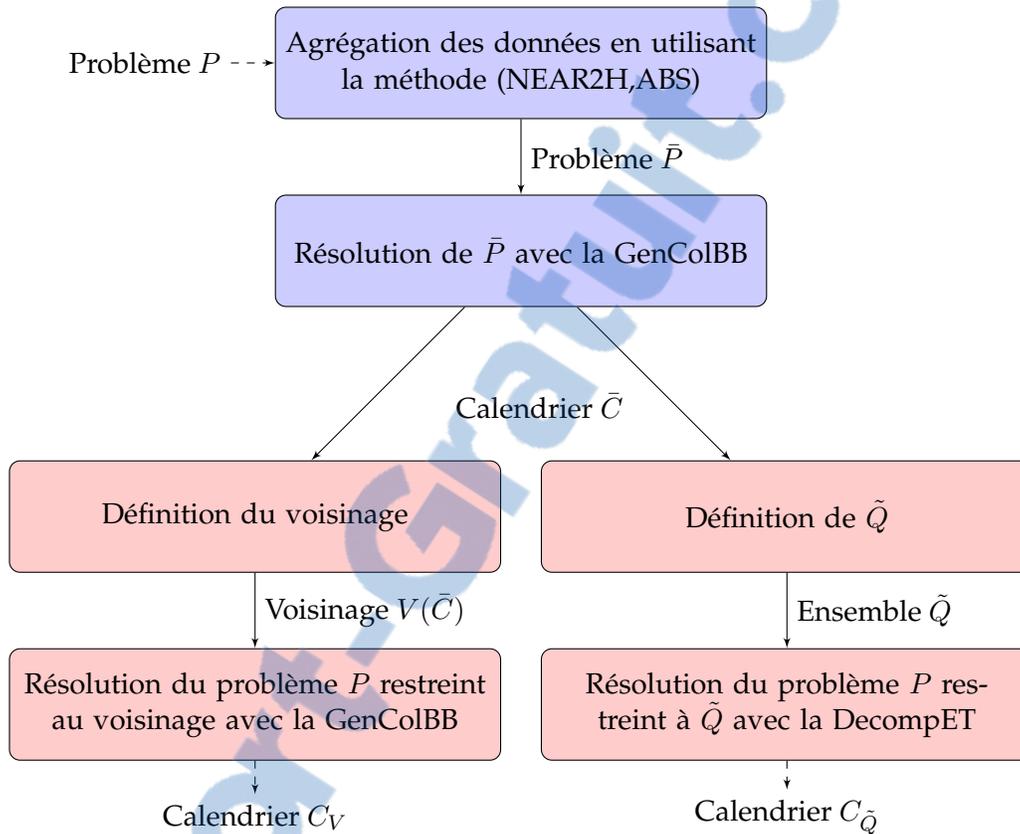


Figure 6.12 – Comparaison des deux procédures GenColBB et DecompET pour la résolution du problème désagrégé P

La Figure 6.12 illustre l'analyse comparative élaborée dans cette partie. Rappelons que l'heuristique de prétransfert est formée par deux étapes : l'agrégation et la désagrégation. L'agrégation se fait en définissant un problème agrégé \bar{P} à l'aide de la méthode (NEAR2H,ABS) et en le résolvant par notre procédure GenColBB. La solution correspond à un calendrier agrégé \bar{C} . La désagrégation peut se faire selon deux procédures distinctes :

- La première procédure, la plus simple, définit un voisinage du calendrier agrégé que nous avons noté $V(\bar{C})$. Le problème désagrégé P restreint à $V(\bar{C})$ est résolu par la GenColBB. Nous obtenons la solution donnée par le calendrier C_V .
- La deuxième procédure, la plus sophistiquée, définit un ensemble \tilde{Q} de quarts en retenant certaines informations du calendrier agrégé \bar{C} . Le problème désagrégé P restreint

à \tilde{Q} est résolu par la procédure de décomposition espace-temps DecompET. Nous obtenons la solution donnée par le calendrier $C_{\tilde{Q}}$. La procédure DecompET est configurée dépendamment de la taille de l'organisation :

- DecompET(W3,A,P,D,B) pour les petites organisations D2E20, D3E20 et D5E20,
- DecompET(W1,V,P,D,B) pour les moyennes organisations D5E50, D5E70 et D5E200,
- DecompET(W7,V,P,E,B) pour les grandes organisations D10E200, D10E300 et D10E400,
- et DecompET(W7,V,P,E,B) pour les très grandes organisations D20E600, D20E800 et D20E1000.

Le Tableau 6.9 compare les performances des procédures GenColBB et DecompET pour toutes les instances du JEU1. Le tableau donne le nombre de quarts désagrégés formant le sous-ensemble restreint de quarts exploré par chacune des deux procédures. Ce nombre correspond à la cardinalité du voisinage $V(\bar{C})$ pour la GenColBB et la cardinalité de l'ensemble \tilde{Q} pour la DecompET. Observons que $|V(\bar{C})| \leq |\tilde{Q}|$ pour toutes les instances. Le tableau donne également le nombre de variables de décision du problème désagrégé P restreint à $V(\bar{C})$ et le nombre moyen de variables des sous-problèmes $P(w, d)$ restreints aux $\tilde{Q}(w, d)$. Le nombre total de sous-problèmes est indiqué entre parenthèses. Par exemple, la procédure DecompET(W1,V,P,D,B) divise le problème P restreint à \tilde{Q} en 30 sous-problèmes $P(w, d)$ obtenus par six fenêtres de temps et cinq départements pour l'instance D5E50P1. Les colonnes suivantes correspondent aux coûts des calendriers obtenus par les deux procédures (Z_V pour la GenColBB et $Z_{\tilde{Q}}$ pour la DecompET). Les deux dernières colonnes correspondent aux temps total requis pour compléter chacune des deux procédures GenColBB et DecompET notés respectivement CPU_V et $\text{CPU}_{\tilde{Q}}$.

Nous analysons les résultats fournis par le tableau 6.9 pour le cas des petites, moyennes et grandes instances d'une part, et pour le cas des très grandes instances d'autre part. Pour le premier cas, cette analyse permet de distinguer les quatre situations suivantes :

- Uniquement pour deux instances de grande taille (marquées par (1)), la procédure DecompET affiche une meilleure performance que la procédure GenColBB en termes de la valeur de l'objectif et du temps de calcul. La déviation moyenne $\frac{Z_V - Z_{\tilde{Q}}}{Z_V}$ est égale à 4% avec un gain moyen de temps considérable ($\text{CPU}_V - \text{CPU}_{\tilde{Q}}$) égal à 3,129.59 secondes.
- Uniquement pour deux instances de petite taille (marquées par (2)), la procédure DecompET permet de calculer des solutions de qualité légèrement meilleure que celles de la procédure GenColBB mais en consommant un temps de calcul plus élevé. La déviation moyenne est égale à 0.8% avec une augmentation moyenne importante du temps égale à 310.5 secondes.
- Pour 22 instances de moyenne et grande taille (marquées par (3)), la procédure DecompET converge rapidement vers une solution de mauvaise qualité avec une détérioration moyenne de l'objectif égale à 37.8% et une diminution moyenne du temps

égale à 2,314.87 secondes. Dans ce cas, la procédure DecompET affiche une mauvaise performance par rapport à la désagrégation par la procédure GenColBB.

- Pour 10 instances de petite taille (marquées par (4)), la procédure DecompET affiche une pire performance que la procédure GenColBB en termes de la valeur de l'objectif et du temps de calcul. La perte de la qualité de la solution est chiffrée à 12.4% d'augmentation du coût. La perte moyenne du temps de calcul est égale à 349.59 secondes.

Pour le deuxième cas, celui des très grandes instances, le Tableau 6.9 montre que la tendance identifiée pour le cas des petites, moyennes et grandes organisations s'inverse. C'est la procédure DecompET qui domine largement la procédure GenColBB. L'analyse des résultats permet de distinguer les deux situations suivantes :

- Pour 10 instances de très grande taille parmi les 12 considérées (marquées par (1)), la procédure DecompET affiche largement une meilleure performance que la procédure GenColBB en termes de la valeur de l'objectif et du temps de calcul. La déviation moyenne est égale à 68.5% avec un gain moyen de temps considérable égal à 2,526.56 secondes.
- Pour les deux instances de très grande taille restantes (marquées par (3)), la procédure DecompET converge rapidement vers une solution de mauvaise qualité avec une détérioration moyenne de l'objectif égale à 11.9% et une diminution moyenne du temps égale à 3,036.51 secondes.

Notons que pour le cas des très grandes instances, le temps d'énumération des circuits élémentaires du graphe de transfert associé à l'horizon croît exponentiellement. En conséquence, nous utilisons une heuristique simple pour l'élimination des circuits. Au lieu d'énumérer tous les circuits dans le graphe, nous effectuons une recherche en profondeur d'abord des sommets du graphe. À chaque fois qu'un circuit est détecté, nous supprimons l'arc de poids maximal dans ce circuit. Le processus itératif s'arrête lorsque tous les circuits sont éliminés.

En conclusion, nous avons décidé de retenir pour les petites, moyennes et grandes instances la procédure de désagrégation basée sur la GenColBB alors que pour les très grandes instances, nous avons décidé de retenir la procédure de décomposition espace-temps avec la configuration (W7,V,P,E,B). En effet, tant que la taille du voisinage $V(\bar{C})$ reste raisonnable, l'exploration implicite de l'arbre de branchement permet d'obtenir des calendriers de coût meilleur que ceux trouvés par la décomposition espace-temps. Lorsque la taille du voisinage $V(\bar{C})$ devient élevée pour le cas des très grandes organisations, la décomposition permet de trouver des solutions de qualité meilleure en contournant la complexité élevée des problèmes à résoudre. Notons que la configuration DecompET(W7,V,P,E,B) exclut la procédure de l'horizon roulant et se contente de diviser le problème par département.

Notons que la procédure de décomposition espace-temps peut fournir des solutions réalisables en des temps largement inférieurs à ceux de la procédure GenColBB pour le cas des

moyennes et grandes instances. Ceci permet ainsi de fournir des calendriers admissibles lorsqu'on a besoin d'une solution dans des temps courts.

Afin de conclure ce chapitre, nous soulignons les contributions de notre travail et nous proposons certains axes d'amélioration :

- La procédure de décomposition espace-temps offre une alternative pour l'optimisation distribuée des problèmes de construction d'horaires avec partage de ressources en effectifs entre départements pour les organisations de très grande taille.
- La procédure de décomposition espace-temps a la vocation de permettre une résolution en parallèle de certains sous-problèmes ce qui réduirait davantage les temps de résolution (voir Figure 6.4). À l'opposé, la procédure GenColBB ne permet pas une telle parallélisation.
- L'amélioration de la procédure de décomposition espace-temps peut se faire en proposant une procédure de résolution plus performante des sous-problèmes $P(w, d)$ autre que le BB et la GenColBB. Cette procédure peut être basée sur un algorithme de Branch-and-Price.
- Les sous-problèmes $P(w, d)$ peuvent être modélisés à l'aide de la formulation basée sur les blocs de type MCB2 étant donné qu'ils vérifient l'hypothèse liée à cette formulation. En effet, au moment de définir un sous-problème, nous connaissons d'avance que tous les employés peuvent être transférés vers un seul département d'accueil. Étant donné que dans le Chapitre 4 nous avons remarqué que le modèle MCB2 affiche une meilleure performance que le modèle MCQ, cette idée peut amener à une amélioration de la performance de notre procédure de décomposition espace-temps.
- Il est possible de tester la procédure de désagrégation basée sur la GenColBB en utilisant différentes configurations lors de la définition du voisinage $V(\bar{C})$. Dans le présent travail, cette procédure était à la base utilisée comme moyen simple d'évaluation des qualités des calendriers agrégés obtenus par les différentes méthodes d'agrégation. Malgré qu'elle permet d'éviter la détérioration de la qualité des solutions provoquée par la décomposition, elle reste limitée par la taille des organisations et perd de son efficacité pour les organisations les plus grandes. La procédure de décomposition permet d'outrepasser la complexité algorithmique en séparant le problème en plusieurs sous-problèmes de difficulté moindre.

Tableau 6.9 – Comparaison des procédures GenColBB et DecompET pour le JEU1

Instance		Nombre de quarts		Nombre de variables		Coût du calendrier désagrégé Z_C		Temps de résolution CPU (secondes)		
		GenCol -BB	Decomp -ET	# Vars.	# Moyen Vars.	GenCol -BB	Decomp -ET	GenCol -BB	Decomp -ET	
D2E20P1	(2)	8,091	49,080	9,172	28,231	(2)	290.0	287.8	12.1	90.5
D2E20P2	(4)	9,935	55,218	9,378	18,742	(6)	501.1	651.6	26.9	436.0
D2E20P3	(4)	9,502	58,028	9,107	19,394	(6)	483.3	586.3	11.0	97.3
D2E20P4	(2)	9,167	56,438	9,425	22,873	(4)	405.9	402.3	79.9	622.5
D3E20P1	(4)	9,678	53,642	10,808	14,732	(6)	292.5	314.7	8.6	37.5
D3E20P2	(4)	9,725	55,780	10,659	12,908	(9)	713.7	820.4	8.9	258.8
D3E20P3	(4)	12,471	68,183	10,774	14,517	(9)	641.4	756.7	15.4	697.3
D3E20P4	(4)	7,858	53,606	10,456	14,744	(6)	709.3	728.7	9.6	455.4
D5E20P1	(4)	9,874	54,841	13,574	8,144	(15)	418.6	459.3	12.9	172.8
D5E20P2	(4)	8,627	50,410	13,318	9,308	(10)	595.2	619.4	8.7	632.9
D5E20P3	(4)	11,272	58,223	13,840	8,530	(15)	654.0	741.1	19.1	212.3
D5E20P4	(4)	9,569	50,315	13,569	7,975	(15)	813.4	833.0	53.0	669.9
D5E50P1	(3)	25,084	39,841	24,136	4,581	(30)	556.6	743.1	469.5	19.6
D5E50P2	(3)	30,013	53,717	23,851	4,906	(25)	968.0	1,310.7	38.2	16.9
D5E50P3	(3)	32,246	54,674	24,542	4,298	(40)	920.0	1,473.3	202.8	22.4
D5E50P4	(3)	28,981	51,669	23,982	4,591	(35)	1,030.8	1,430.1	51.7	21.9
D5E70P1	(3)	38,249	60,224	31,716	5,442	(40)	661.7	1,078.5	439.1	27.8
D5E70P2	(3)	43,231	71,948	31,468	5,247	(40)	1,340.3	2,029.2	426.8	26.4
D5E70P3	(3)	45,425	82,521	31,952	5,530	(50)	1,223.0	2,016.9	2,524.4	35.4
D5E70P4	(3)	42,576	83,751	31,504	5,414	(50)	1,036.2	1,801.2	528.6	35.2
D5E200P1	(3)	100,800	196,589	75,247	16,324	(35)	1,568.5	2,122.8	3,600.6	84.0
D5E200P2	(3)	104,286	200,035	75,756	16,131	(40)	2,552.2	3,367.2	3,600.6	119.0
D5E200P3	(3)	118,035	268,001	77,829	16,622	(50)	2,968.7	4,445.4	3,600.5	115.0
D5E200P4	(3)	101,371	214,324	73,571	15,333	(45)	3,032.0	4,451.8	3,600.5	101.9
D10E200P1	(3)	100,046	128,484	81,904	15,360	(10)	1,448.0	2,102.0	3,600.5	85.6
D10E200P2	(3)	103,030	151,360	82,272	17,705	(10)	2,952.0	3,820.1	3,600.6	303.7
D10E200P3	(3)	109,191	165,444	85,009	19,164	(10)	3,437.1	4,057.3	3,600.5	274.5
D10E200P4	(3)	104,374	162,590	82,979	19,813	(10)	3,549.4	4,420.7	3,600.4	260.9
D10E300P1	(3)	146,556	193,810	115,428	23,203	(10)	3,193.6	4,263.8	3,601.1	127.8
D10E300P2	(3)	186,794	264,270	121,600	29,831	(10)	3,705.8	4,835.1	3,600.7	592.3
D10E300P3	(3)	184,163	277,726	120,622	31,487	(10)	5,528.2	5,750.9	3,600.4	401.8
D10E300P4	(3)	165,051	251,063	116,466	29,015	(10)	4,802.6	6,064.0	3,600.6	495.8
D10E400P1	(1)	206,048	254,085	151,434	30,194	(10)	4,101.4	3,902.8	3,600.8	217.5
D10E400P2	(1)	192,334	268,273	150,238	30,274	(10)	4,697.0	4,550.9	3,600.7	724.9
D10E400P3	(3)	206,158	311,879	152,672	34,807	(10)	5,943.0	7,207.5	3,600.8	575.8
D10E400P4	(3)	194,487	297,052	148,495	33,389	(10)	5,830.1	6,688.9	3,603.3	421.6
D20E600P1	(1)	277,401	357,791	231,000	20,842	(20)	12,270.9	5,934.2	3,600.7	658.0
D20E600P2	(1)	296,516	427,489	229,953	23,676	(20)	18,023.9	8,648.0	3,600.8	1,391.2
D20E600P3	(3)	299,560	444,217	231,702	23,631	(20)	9,482.8	11,477.8	3,600.9	676.2
D20E600P4	(1)	278,300	425,181	227,353	24,128	(20)	216,449.0	11,310.6	3,600.7	524.8
D20E800P1	(3)	364,973	460,876	294,702	26,885	(20)	8,415.4	8,652.1	3,601.7	453.4
D20E800P2	(1)	419,700	608,703	302,153	34,268	(20)	25,106.9	11,631.4	3,601.0	1,510.0
D20E800P3	(1)	389,082	579,493	295,887	32,314	(20)	305,035.0	16,752.0	3,602.5	578.7
D20E800P4	(1)	373,142	563,984	293,749	32,415	(20)	24,661.6	14,946.4	3,600.9	674.3
D20E1000P1	(1)	414,818	522,607	357,462	29,522	(20)	22,435.6	9,582.0	3,601.7	799.5
D20E1000P2	(1)	528,314	765,176	363,238	43,167	(20)	33,262.6	16,146.3	3,601.4	1,366.9
D20E1000P3	(1)	504,617	748,699	365,755	39,341	(20)	378,340.0	21,110.1	3,602.7	1,652.5
D20E1000P4	(1)	434,123	676,729	359,026	37,571	(20)	379,595.0	16,813.2	3,602.3	1,593.2

Chapitre 7

Approche de résolution post-transfert

Ce chapitre décrit l'heuristique de post-transfert pour la résolution du problème de construction de quarts dans un contexte multidépartement. L'heuristique que nous proposons est conforme à une architecture qui divise le problème en deux niveaux décisionnels : la construction des quarts internes personnalisés et le placement des blocs externes à l'intérieur de ces quarts. Notons que la séparation décisionnelle n'est pas totale du moment que la décision portant sur les heures de début et les durées de travail est légèrement révisée au moment du placement des blocs de transfert. D'une manière plus détaillée, le problème est scindé en deux étapes résolues séquentiellement :

- Dans la première étape, nous traitons le problème de construction de quarts sans partage de ressources en effectifs entre les départements. Dans ce cas, le problème global est séparable par département. Pour chaque département d , nous résolvons un problème agrégé \bar{P}_d qui permet de construire un ensemble optimisé de quarts en utilisant exclusivement la main d'œuvre interne. L'extrait de cette étape est un calendrier agrégé interne \bar{C}^{int} .
- Dans la deuxième étape, nous définissons un problème désagrégé P qui intègre l'ensemble des départements de l'organisation dans un même processus d'optimisation. Les heures de début et les durées de travail du calendrier \bar{C}^{int} sont légèrement perturbées et un ensemble particulier de blocs externes sont considérés. Le problème désagrégé P restreint à un sous-ensemble spécifique de quarts est résolu permettant ainsi d'obtenir le calendrier C solution du problème global.

Plusieurs techniques introduites dans les Chapitres 4 et 6 seront utilisées dans l'élaboration de l'heuristique de post-transfert.

7.1 Problème agrégé séparable

Cette partie décrit la première étape de l'heuristique de post-transfert. Le principe de cette heuristique repose sur la contrainte horizontale de transfert qui impose que la proportion

des périodes travaillées dans le département d'origine soit au minimum égale à $T_{min} = 70\%$ du nombre total des périodes travaillées. Étant donné que l'horaire de l'employé est principalement constitué par les blocs internes, l'heuristique commence par traiter cet aspect décisionnel du problème en priorité puis passe à prendre les décisions sur les transferts en fonction des quarts préétablis.

Considérons la formulation basée sur les quarts MCQ. Lorsque l'ensemble des quarts personnalisés est formé uniquement par les quarts internes, les contraintes de demande (4.7) ne sont plus des contraintes liantes. Le problème devient séparable par département. Comme précédemment, la technique d'agrégation est utilisée pour réduire la complexité algorithmique des problèmes à résoudre.

Pour chaque département d , nous considérons le problème agrégé \bar{P}_d qui s'énonce comme suit :

Formulation MCQ du problème \bar{P}_d

$$\min \gamma^- \sum_{\bar{i} \in \bar{I}} c_d^- U_{d,\bar{i}} + \gamma^+ \sum_{d \in D} \sum_{\bar{i} \in \bar{I}} c_d^+ O_{d,\bar{i}} + \gamma^l \sum_{e \in E_d} \sum_{j \in J_e} \sum_{q \in \bar{Q}_{e,j}^{int}} c_q^l X_q \quad (7.1)$$

$$\text{s.c.} \sum_{e \in E_d} \sum_{j \in J_e} \sum_{q \in \bar{Q}_{e,j}^{int}} \delta_{q,d,\bar{i}} X_q + U_{d,\bar{i}} - O_{d,\bar{i}} = r_{d,\bar{i}}, \quad \bar{i} \in \bar{I} \quad (7.2)$$

$$\sum_{q \in \bar{Q}_{e,j}^{int}} X_q = 1, \quad \forall e \in E, j \in J_e \quad (7.3)$$

$$\sum_{q' \in \bar{Q}_{e,j+1}^{int}} s_{q'} X_{q'} - \sum_{q \in \bar{Q}_{e,j}^{int}} (s_q + l_q) X_q \geq S_{min}, \quad \forall e \in E, (j, j+1) \in J_e^2 \quad (7.4)$$

$$\sum_{j \in J_e} \sum_{q \in \bar{Q}_{e,j}^{int}} l_q X_q \leq L_{max}, \quad \forall e \in E \quad (7.5)$$

$$X_q \in \{0, 1\}, \forall e \in E, j \in J_e, q \in \bar{Q}_{e,j}^{int} \quad (7.6)$$

$$U_{d,\bar{i}}, O_{d,\bar{i}} \geq 0, \bar{i} \in \bar{I} \quad (7.7)$$

La fonction objectif (7.1) vise à minimiser le coût total de la main d'œuvre formé par le coût total de sous-couverture, de surcouverture et de travail. Les contraintes de demande (7.2) permettent d'ajuster l'offre à la demande en effectifs. L'offre inclut uniquement les employés internes du département d (E_d) et on considère uniquement l'ensemble des quarts internes de ces employés ($\cup_{e \in E_d} \bar{Q}_{e,j}^{int}$). Les contraintes (7.3) assurent le respect des cycles pré-assignés aux employés. Les contraintes (7.4) et (7.5) imposent l'admissibilité des horaires aux réglementations horizontales de travail. Notons que la contrainte horizontale de transfert est supprimée du moment qu'elle est vérifiée pour tout calendrier interne.

Comme pour l'heuristique de prétransfert, l'agrégation des données se fait à l'aide de la méthode (NEAR2H,ABS). Chaque problème \bar{P}_d est résolu par le Branch and Bound de CPLEX.

Les problèmes $(\bar{P}_d)_{d \in D}$ peuvent être traités en parallèle ce qui constitue un avantage de la méthode lors de son implémentation. Soit \bar{C}^{int} le calendrier agrégé interne formé par l'ensemble des quarts sélectionnés par tous les problèmes $(\bar{P}_d)_{d \in D}$ associés aux départements de l'organisation. Étant donné que l'agrégation des réglementations se fait en gardant les données initiales les plus proches des multiples de deux heures, le calendrier agrégé \bar{C}^{int} constitue une solution admissible pour le problème global P .

7.2 Problème désagrégé

Une fois le calendrier interne \bar{C}^{int} obtenu, nous passons à définir et résoudre un problème désagrégé P restreint à un ensemble de quarts spécifiques internes et externes. Cet ensemble de quarts dépend de la méthode de désagrégation à utiliser. Dans le cadre de cette thèse, nous proposons trois méthodes de désagrégation pour le placement à posteriori des transferts :

- La première méthode utilise la formulation basée sur les quarts MCQ pour formuler le problème désagrégé P restreint à un ensemble de quarts \tilde{Q}_{MCQ} . Ce problème est résolu par la procédure GenColBB et cette méthode sera désignée par la GenColBB-MCQ.
- La deuxième et troisième méthodes utilisent la formulation basée sur les blocs de type MCB24 pour formuler le problème désagrégé P restreint à un ensemble de quarts \tilde{Q}_{MCB} . Ce problème est résolu par les deux procédures BB et GenColBB. Les méthodes qui en résultent seront désignées par le BB-MCB et la GenColBB-MCB.

Dans la suite, nous expliquons les détails relatifs aux trois méthodes GenColBB-MCQ, BB-MCB et GenColBB-MCB représentées dans la Figure 7.1 et dont l'extrait est donné par les calendriers C_{MCQ}^G , C_{MCB}^B et C_{MCB}^G , respectivement.

7.2.1 Désagrégation basée sur les quarts GenColBB-MCQ

Dans cette partie, il suffit de décrire la démarche de construction de l'ensemble restreint \tilde{Q}_{MCQ} . Une fois cet ensemble défini, nous utilisons la procédure GenColBB pour résoudre la formulation basée sur les quarts du problème désagrégé P restreint à \tilde{Q}_{MCQ} .

À partir du calendrier agrégé interne \bar{C}^{int} , nous disposons d'un quart agrégé interne $\bar{q}_{e,j}$ pour chaque employé durant chaque jour travaillé $j \in J_e$. Nous définissons un ensemble de quarts vides personnalisés pour chaque employé e le jour travaillé j qui s'exprime comme suit : $Q_{e,j}^{vides} = \cup_{j' \in J_e} \{(s_q, l_q) | q \in Q_{e,j}, s_{\bar{q}_{e,j'}} - \theta^l \leq s_q \leq s_{\bar{q}_{e,j'}} + \theta^l, l_{\bar{q}_{e,j'}} - \theta^l \leq l_q \leq l_{\bar{q}_{e,j'}} + \theta^l\}$. Cet ensemble correspond aux quarts vides voisins qui sont obtenus par une perturbation des heures de début et des durées de travail agrégées. Comme précédemment, θ^l représente l'amplitude de cette perturbation.

Après la construction des quarts vides, nous passons au placement des départements à l'intérieur de ces quarts. L'ensemble des quarts internes et externes \tilde{Q}_{MCQ} qui sera exploré par

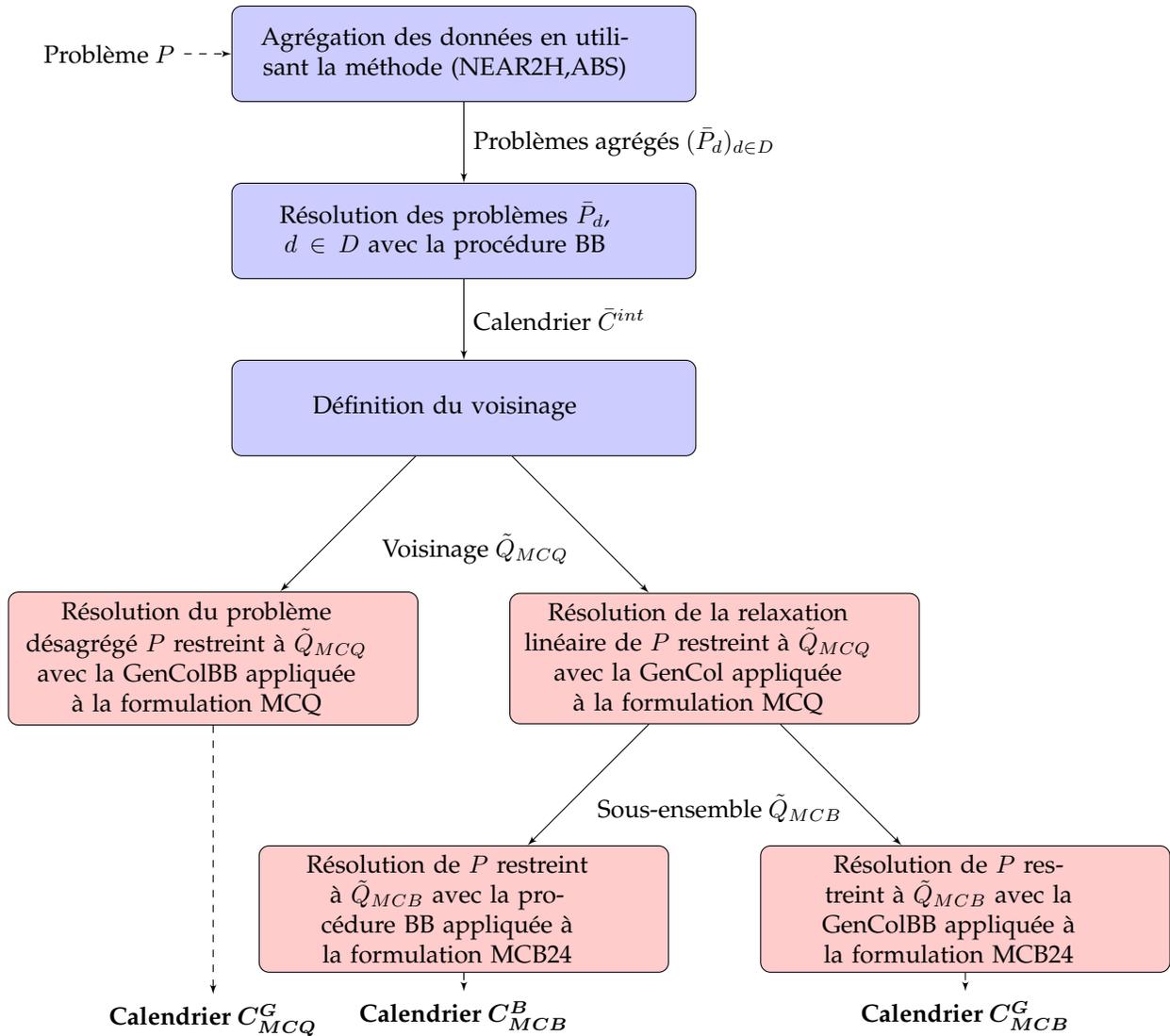


Figure 7.1 – Méthodes de désagrégation pour l’heuristique de post-transfert

la procédure GenColBB est formé par toutes les possibilités de placer les départements d’origine et d’accueil dans les quarts vides. Formellement, cet ensemble est exprimé comme suit : $\tilde{Q} = \cup_{e \in E} \cup_{j \in J_e} \tilde{Q}_{e,j}$ avec $\tilde{Q}_{e,j} = \{(s_q, l_b, l_f, d_b, d_f) \in Q_{e,j} | d_b, d_f \in D_e \cup \{d_e\}\}$. En d’autres termes, nous considérons tous les quarts admissibles dont les heures de début et les durées sont dans le voisinage du calendrier agrégé interne.

7.2.2 Désagrégation basée sur les blocs BB-MCB

Rappelons que l’utilisation du modèle réduit de blocs MCB2 suppose que la décision portant sur le placement des départements d’accueil pour chaque employé e durant chaque jour travaillé $j \in J_e$ soit partiellement établie. En effet, de par la définition du problème, nous connaissons que chaque employé peut être transféré à au plus un département d’accueil

durant chaque jour travaillé. Si nous décidons de l'identité du département receveur, il ne reste qu'à décider de l'heure de début et la durée du bloc externe pour déterminer les périodes transférées.

Au plus un département d'accueil sera donc considéré pour chaque paire (e, j) lors de la définition du modèle MCB2. La détermination du département receveur se fait simplement par la résolution de la relaxation linéaire du problème désagrégé P restreint à \tilde{Q}_{MCQ} modélisé à l'aide de la formulation basée sur les quarts. Il s'agit de la relaxation linéaire du modèle résolu par la méthode GenColBB-MCQ (voir Section 7.2.1). La résolution se fait en utilisant la procédure de génération de colonnes au nœud racine. Ainsi, nous obtenons un ensemble de variables de quarts externes ayant des valeurs fractionnaires non nulles. Nous sélectionnons pour chaque employé e et chaque jour $j \in J_e$ la variable de quart externe ayant la valeur non nulle la plus élevée si elle existe. À la fin de cette étape, nous disposons pour chaque employé e et chaque jour travaillé $j \in J_e$ d'au plus un département d'accueil. Notons $\tilde{D}_{e,j}$ l'ensemble des départements potentiels dans lesquels l'employé e peut travailler le jour j . Ces départements sont le département d'origine d_e et le département d'accueil receveur $d_{e,j}$ s'il y en a d'après la résolution de la relaxation linéaire du modèle basé sur les quarts restreint à \tilde{Q}_{MCQ} . En d'autres termes, deux cas sont possibles : $\tilde{D}_{e,j} = \{d_e\}$ ou $\tilde{D}_{e,j} = \{d_e, d_{e,j}\}$ avec $d_{e,j}$ le seul département receveur s'il y en a.

Comme pour la méthode GenColBB-MCQ, la méthode BB-MCB commence par énumérer tous les quarts vides voisins au calendrier agrégé interne Q^{vides} . Ensuite, le placement des départements se fait en énumérant toutes les possibilités d'affecter les départements admissibles $\tilde{D}_{e,j}$ aux quarts voisins vides.

Il est clair que l'ensemble $\tilde{Q}_{MCB} \subseteq \tilde{Q}_{MCQ}$. En effet, il s'agit de limiter l'ensemble des quarts externes \tilde{Q}_{MCQ} à ceux contenant des transferts vers les départements d'accueil spécifiques identifiés par la résolution de la relaxation linéaire de la formulation basée sur les quarts.

Une fois l'ensemble \tilde{Q}_{MCB} construit, nous passons à définir le modèle réduit de construction de blocs MCB24 correspondant au problème désagrégé P restreint à \tilde{Q}_{MCB} . Rappelons que nous avons considéré dans le Chapitre 4 des instances formant le JEU3 qui se rapprochent des instances du JEU1 que l'on vise à traiter lors de la désagrégation post-transfert. Rappelons que nous avons montré expérimentalement avec un ensemble de tests que la formulation MCB24 est la plus performante pour l'ensemble de ces instances. Ainsi, nous retenons la version MCB24 sans blocs fictifs et sans contraintes de liaison entre départements.

Finalement, le problème P restreint à \tilde{Q}_{MCB} modélisé par la formulation MCB24 est résolu par le Branch and Bound de CPLEX.

7.2.3 Désagrégation basée sur les blocs GenCol-MCB

La troisième méthode diffère de la deuxième par la procédure d'exploration du même ensemble de quarts \tilde{Q}_{MCB} . La procédure de GenColBB est utilisée avec le même paramétrage établi dans le Chapitre 6. Dans ce qui suit, nous présentons les modifications qui doivent être introduites afin d'appliquer la procédure au modèle basé sur les blocs.

L'ensemble des variables initialement considéré est formé par les quarts internes de tous les employés et toutes les variables d'écart \bar{V} et \bar{W} correspondantes aux nouvelles contraintes forward et backward (voir Équations (4.44)-(4.47), le surlignage des variables de type \bar{V} et \bar{W} est pour dire qu'il faut considérer les graphes de transport modifiés $\bar{G} = (\bar{K}_{e,j,t}^1, \bar{K}_{e,j,t}^2, \bar{A}_{e,j,t})$ correspondant à la version du modèle réduit de construction de blocs MCB2). À chaque itération, pour chaque employé e et chaque jour travaillé $j \in J_e$, nous ajoutons au maximum les 10 colonnes ayant les coûts réduits strictement négatifs les moins élevés. Ces colonnes peuvent être des blocs de début, de fin ou des quarts complètement externes. Le coût réduit des quarts complètement externes est déjà donné dans l'application de la procédure GenColBB à la formulation basée sur les quarts (voir l'équation (5.17)).

Calculons le coût réduit d'un bloc de début interne de l'employé e débutant durant le jour $j \in J_e$ et admissible selon le type de quart t . Ce bloc interne est noté (b, d_e) avec $b \in \bar{K}_{e,j,t}^1$.

$$\begin{aligned}
 c_{(b,d_e)} = & \gamma^l l_b c_e^l - \sum_{i \in I} \delta_{b,d,i} \Pi_{d,i}^{dem} \\
 & - \sum_{m=0}^{\bar{M}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,m}^{1F}}(b) \Pi_{e,j,t,m}^{for,\bar{V}} - \sum_{r=0}^{\bar{R}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,r}^{1B}}(b) \Pi_{e,j,t,r}^{bac,\bar{V}} \\
 & - s_b \delta_{j-1,j} \Pi_{e,j-1,j}^{sep} + (s_b + l_b) \delta_{j,j+1} \Pi_{e,j,j+1}^{sep} - l_b \Pi_e^l - l_b (1 - T_{min}) \Pi_e^{tr} - \Pi_{e,j}^{jour}
 \end{aligned} \tag{7.8}$$

, avec :

- $\gamma^l l_b c_e^l$: le coût des périodes travaillées couvertes par le bloc b .
- $-\sum_{i \in I} \delta_{b,d,i} \Pi_{d,i}^{dem}$: le coût réduit correspondant à la couverture des courbes de demande.
- $-\sum_{m=0}^{\bar{M}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,m}^{1F}}(b) \Pi_{e,j,t,m}^{for,\bar{V}}$: le coût réduit correspondant aux contraintes forward pour la liaison des blocs de début internes aux blocs de fin externes où $\mu_{\bar{K}_{e,j,t,m}^{1F}}(b) = 1$ si $b \in \bar{K}_{e,j,t,m}^{1F}$ et 0 sinon.
- $-\sum_{r=0}^{\bar{R}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,r}^{1B}}(b) \Pi_{e,j,t,r}^{bac,\bar{V}}$: le coût réduit correspondant aux contraintes backward pour la liaison des blocs de début internes aux blocs de fin externes où $\mu_{\bar{K}_{e,j,t,r}^{1B}}(b) = 1$ si $b \in \bar{K}_{e,j,t,r}^{1B}$ et 0 sinon.
- $-s_b \delta_{j-1,j} \Pi_{e,j-1,j}^{sep} + (s_b + l_b) \delta_{j,j+1} \Pi_{e,j,j+1}^{sep}$: le coût réduit correspondant aux deux contraintes de séparation entre les jours consécutifs $j-1, j$ et $j, j+1$.

- $-l_b \Pi_e^l$: le coût réduit correspondant à la contrainte de la charge de travail maximale.
- $-l_b(1-T_{min})\Pi_e^{tr}$: le coût réduit correspondant à la contrainte du pourcentage de travail minimal sans transfert.
- $-\Pi_{e,j}^{jour}$: le coût réduit correspondant à la contrainte du jour travaillé j selon le cycle reçu par l'employé.

Calculons le coût réduit d'un bloc de début externe de l'employé e correspondant au jour $j \in J_e$ et admissible selon le type de quart t . Ce bloc externe est noté $(b, d_{e,j})$ avec $b \in \bar{K}_{e,j,t}^1$ et $d_{e,j}$ le département d'accueil receveur de l'employé e le jour j s'il existe.

$$\begin{aligned}
c_{(b,d_{e,j})} = & \gamma^l l_b c_e^l + \gamma^{tr} l_b c_{e,d_{e,j}}^{tr} - \sum_{i \in I} \delta_{b,d_{e,j},i} \Pi_{d_{e,j},i}^{dem} \\
& - \sum_{m=0}^{\bar{M}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,m}^{1F}}(b) \Pi_{e,j,t,m}^{for,\bar{W}} - \sum_{r=0}^{\bar{R}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,r}^{1B}}(b) \Pi_{e,j,t,r}^{bac,\bar{W}} \\
& - s_b \delta_{j-1,j} \Pi_{e,j-1,j}^{sep} + (s_b + l_b) \delta_{j,j+1} \Pi_{e,j,j+1}^{sep} - l_b \Pi_e^l + l_b T_{min} \Pi_e^{tr} - \Pi_{e,j}^{jour}
\end{aligned} \tag{7.9}$$

, avec :

- $\gamma^l l_b c_e^l$: le coût des périodes travaillées couvertes par le bloc b .
- $\gamma^{tr} l_b c_{e,d_{e,j}}^{tr}$: le coût des périodes transférées réalisées durant le bloc b dans le département receveur $d_{e,j}$.
- $-\sum_{i \in I} \delta_{b,d_{e,j},i} \Pi_{d_{e,j},i}^{dem}$: le coût réduit correspondant à la couverture des courbes de demande.
- $-\sum_{m=0}^{\bar{M}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,m}^{1F}}(b) \Pi_{e,j,t,m}^{for,\bar{W}}$: le coût réduit correspondant aux contraintes forward pour la liaison des blocs de début externes aux blocs de fin internes.
- $-\sum_{r=0}^{\bar{R}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,r}^{1B}}(b) \Pi_{e,j,t,r}^{bac,\bar{W}}$: le coût réduit correspondant aux contraintes backward pour la liaison des blocs de début externes aux blocs de fin internes.
- $-s_b \delta_{j-1,j} \Pi_{e,j-1,j}^{sep} + (s_b + l_b) \delta_{j,j+1} \Pi_{e,j,j+1}^{sep}$: le coût réduit correspondant aux 2 contraintes de séparation entre les jours consécutifs $j-1, j$ et $j, j+1$.
- $-l_b \Pi_e^l$: le coût réduit correspondant à la contrainte de la charge de travail maximale.
- $l_b T_{min} \Pi_e^{tr}$: le coût réduit correspondant à la contrainte du pourcentage de travail minimal sans transfert.
- $-\Pi_{e,j}^{jour}$: le coût réduit correspondant à la contrainte du jour travaillé j selon le cycle reçu par l'employé.

Passons maintenant à calculer le coût réduit d'un bloc de fin interne de l'employé e correspondant au jour $j \in J_e$ et admissible selon le type de quart t . Ce bloc interne est noté (f, d_e) avec $f \in \bar{K}_{e,j,t}^2$.

$$\begin{aligned}
c_{(f,d_e)} = & -\gamma^{tr} l_f c_{e,d_e,j}^{tr} - \sum_{i \in I} \delta_{f,d_e,i} \Pi_{d_e,i}^{dem} + \sum_{i \in I} \delta_{f,d_e,j,i} \Pi_{d_e,j,i}^{dem} \\
& + \sum_{m=0}^{\bar{M}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,m}^{2F}}(f) \Pi_{e,j,t,m}^{for,\bar{W}} + \sum_{r=0}^{\bar{R}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,r}^{2B}}(f) \Pi_{e,j,t,r}^{bac,\bar{W}} \\
& - l_f \Pi_e^{tr}
\end{aligned} \tag{7.10}$$

, avec :

- $-\gamma^{tr} l_f c_{e,d_e,j}^{tr}$: l'économie de rediriger l'employé de son département receveur $d_{e,j}$ à son département d'origine d_e durant la durée l_f du bloc de fin f .
- $-\sum_{i \in I} \delta_{f,d_e,i} \Pi_{d_e,i}^{dem}$: le coût réduit correspondant à la couverture des courbes de demande du département d'origine d_e dans lequel se déroule le bloc f .
- $\sum_{i \in I} \delta_{f,d_e,j,i} \Pi_{d_e,j,i}^{dem}$: le coût réduit correspondant à la redirection de l'employé de son département receveur $d_{e,j}$ vers son département d'origine d_e ce qui réduit la couverture des courbes de demande du département receveur $d_{e,j}$ durant les périodes de réalisation du bloc f .
- $\sum_{m=0}^{\bar{M}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,m}^{2F}}(f) \Pi_{e,j,t,m}^{for,\bar{W}}$: le coût réduit correspondant aux contraintes forward pour la liaison des blocs de début externes aux blocs de fin internes où $\mu_{\bar{K}_{e,j,t,m}^{2F}}(f) = 1$ si $f \in \bar{K}_{e,j,t,m}^{2F}$ et 0 sinon.
- $\sum_{r=0}^{\bar{R}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,r}^{2B}}(f) \Pi_{e,j,t,r}^{bac,\bar{W}}$: le coût réduit correspondant aux contraintes backward pour la liaison des blocs de début externes aux blocs de fin internes où $\mu_{\bar{K}_{e,j,t,r}^{2B}}(f) = 1$ si $f \in \bar{K}_{e,j,t,r}^{2B}$ et 0 sinon.
- $-l_f \Pi_e^{tr}$: le coût réduit correspondant à la contrainte du pourcentage de travail minimal sans transfert.

Calculons le coût réduit d'un bloc de fin externe de l'employé e correspondant au jour $j \in J_e$ et admissible selon le type de quart t . Ce bloc externe est noté $(f, d_{e,j})$ avec $f \in \bar{K}_{e,j,t}^2$ et $d_{e,j}$ le département receveur.

$$\begin{aligned}
c_{(f,d_{e,j})} = & \gamma^{tr} l_f c_{e,d_{e,j}}^{tr} - \sum_{i \in I} \delta_{f,d_{e,j},i} \Pi_{d_{e,j},i}^{dem} + \sum_{i \in I} \delta_{f,d_e,i} \Pi_{d_e,i}^{dem} \\
& + \sum_{m=0}^{\bar{M}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,m}^{2F}}(f) \Pi_{e,j,t,m}^{for,\bar{V}} + \sum_{r=0}^{\bar{R}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,r}^{2B}}(f) \Pi_{e,j,t,r}^{bac,\bar{V}} \\
& + l_f \Pi_e^{tr}
\end{aligned} \tag{7.11}$$

, avec :

- $\gamma^{tr} l_f c_{e,d_{e,j}}^{tr}$: le coût des périodes travaillées à l'externe dans le département receveur $d_{e,j}$ durant le bloc de fin f .

- $-\sum_{i \in I} \delta_{f,d_{e,j},i} \Pi_{d_{e,j},i}^{dem}$: le coût réduit correspondant à la couverture des courbes de demande du département receveur $d_{e,j}$ dans lequel se déroule le bloc f .
- $\sum_{i \in I} \delta_{f,d_e,i} \Pi_{d_e,i}^{dem}$: le coût réduit correspondant à la redirection de l'employé de son département d'origine d_e vers son département receveur $d_{e,j}$ ce qui réduit la couverture des courbes de demande du département d'origine d_e durant les périodes de réalisation du bloc f .
- $\sum_{m=0}^{\bar{M}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,m}^{2F}}(f) \Pi_{e,j,t,m}^{for,\bar{V}}$: le coût réduit correspondant aux contraintes forward pour la liaison des blocs de début internes aux blocs de fin externes où $\mu_{\bar{K}_{e,j,t,m}^{2F}}(f) = 1$ si $f \in \bar{K}_{e,j,t,m}^{2F}$ et 0 sinon.
- $\sum_{r=0}^{\bar{R}_{e,j,t}-1} \mu_{\bar{K}_{e,j,t,r}^{2B}}(f) \Pi_{e,j,t,r}^{bac,\bar{V}}$: le coût réduit correspondant aux contraintes backward pour la liaison des blocs de début internes aux blocs de fin externes où $\mu_{\bar{K}_{e,j,t,r}^{2B}}(f) = 1$ si $f \in \bar{K}_{e,j,t,r}^{2B}$ et 0 sinon.
- $l_f \Pi_e^{tr}$: le coût réduit correspondant à la contrainte du pourcentage de travail minimal sans transfert lors de la redirection de l'employé durant le bloc de fin f vers son département receveur $d_{e,j}$.

Le sous-problème de pricing défini pour chaque employé e et chaque jour travaillé $j \in J_e$ calcule le coût réduit de tous les quarts complètement externes, tous les blocs de début et de fin qui ne sont pas encore considérés par le problème maître restreint. Un nombre maximal de 10 colonnes ayant les coûts réduits strictement négatifs les moins élevés est introduit au problème maître restreint à chaque itération. Ce processus itératif est répété jusqu'à obtenir un optimum positif pour tous les sous-problèmes de pricing de tous les employés et durant tous les jours travaillés. L'intégralité des variables de quart, des variables de blocs de début et de fin est rétablie. Le problème en nombre entiers obtenu est résolu par le Branch and Bound de CPLEX.

7.3 Analyse de performance de l'approche de résolution post-transfert

Dans cette partie, nous commençons par comparer la performance des trois méthodes de désagrégation pour l'affectation à postériori des blocs externes. Une fois la meilleure méthode déterminée, nous passons à comparer la performance des deux approches de pré-transfert et de post-transfert.

7.3.1 Évaluation des performances des méthodes de désagrégation de l'heuristique de post-transfert

Rappelons que l'étape de désagrégation correspondant à l'heuristique de post-transfert se fait selon trois méthodes distinctes pour la résolution du problème désagrégé P restreint à

\tilde{Q} : la méthode GenColBB-MCQ avec $\tilde{Q} = \tilde{Q}_{MCQ}$, la méthode BB-MCB avec $\tilde{Q} = \tilde{Q}_{MCB}$ et la méthode GenColBB-MCB avec $\tilde{Q} = \tilde{Q}_{MCB}$. L'ensemble des quarts voisins au calendrier agrégé interne \bar{C}^{int} est obtenu par une perturbation d'amplitude deux heures des heures de début et des durées de travail. Le placement de tous les départements d'accueil possibles (en plus des départements d'origine) permet d'obtenir l'ensemble \tilde{Q}_{MCQ} . La restriction sur les départements receveurs par la résolution de la relaxation linéaire du problème P restreint à \tilde{Q}_{MCQ} donne l'ensemble \tilde{Q}_{MCB} .

Le Tableau 7.1 donne la taille du problème désagrégé P restreint à \tilde{Q} pour les trois méthodes de désagrégation de l'heuristique post-transfert. Pour les deux méthodes GenColBB-MCQ et GenColBB-MCB, le problème P désagrégé restreint à \tilde{Q} correspond au problème en nombres entiers au niveau du nœud racine. Pour la méthode BB-MCB, il s'agit du problème en nombre entiers restreint à \tilde{Q} sans génération de colonnes. Les lignes du tableau correspondent aux classes d'instances du JEU1. Les colonnes du tableau donnent le nombre moyen de quarts désagrégés formant l'ensemble \tilde{Q} sans et avec la restriction quant à l'identité des départements receveurs définissant ainsi les ensembles \tilde{Q}_{MCQ} et \tilde{Q}_{MCB} , respectivement. Le tableau donne également le pourcentage moyen de diminution du nombre de quarts désagrégés formant l'ensemble \tilde{Q} suite à la décision relative à l'identité des départements receveurs. Cet écart moyen devient très élevé pour les très grandes instances (= 95.67%). Le tableau donne également le nombre moyen de variables de décision, de contraintes et de coefficients non nuls des trois problèmes désagrégés. Il est normal que la méthode BB-MCB définit les problèmes les plus larges du moment que la génération de colonnes n'est pas utilisée pour la sélection des colonnes potentiellement bonnes. Concentrons-nous sur les méthodes GenColBB-MCQ et GenColBB-MCB. La formulation basée sur les quarts induit toujours un nombre moins élevé de contraintes que la formulation implicite basée sur les blocs à cause des contraintes de liaison entre segments de type forward et backward. Dans 35.4% des cas, le nombre de variables de décision correspondant à la méthode GenColBB-MCQ est plus élevé que celui correspondant à la méthode GenColBB-MCB. Le pourcentage moyen d'écart dans ce cas est égal à 16.15%. Dans les 64.6% des cas restants, le nombre de variables de décision correspondant à la méthode GenColBB-MCB est plus élevé que celui correspondant à la méthode GenColBB-MCQ. Cet d'écart moyen est égal à 19.26%. Notons que la différence en nombre de variables de décision est influencée, non seulement par le type de formulation utilisé, mais aussi par les problèmes de pricing résolus lors du processus de la génération de colonnes. La différence en nombre de coefficients non nuls est encore plus significative que celle portant sur le nombre de variables de décision. Dans la plupart des cas (83.3%), la formulation basée sur les quarts nécessite en moyenne 34.18% plus de coefficients que la formulation basée sur les blocs. Ce pourcentage d'écart atteint une valeur maximale de 102.31% pour l'instance D10E200P1. Uniquement pour 16.7% des instances, la formulation basée sur les blocs nécessite en moyenne 9.84% plus de coefficients que la formulation basée sur les quarts.

Tableau 7.1 – Taille du problème désagrégé P restreint à \tilde{Q} pour les trois méthodes de désagrégation post-transfert

Classe	# moyen de quarts désagrégés			# moyen de variables		
	\tilde{Q}_{MCQ}	\tilde{Q}_{MCB}	Différence(%)	GencolBB -MCQ	BB -MCB	GencolBB -MCB
D2E20	127,151	116,859	11.33	13,356	28,776	16,279
D3E20	168,191	110,729	34.50	14,395	29,124	16,862
D5E20	349,975	136,093	60.95	19,794	35,869	20,498
D5E50	937,011	594,581	37.83	54,976	118,435	56,162
D5E70	1,253,265	934,002	26.51	65,030	179,216	80,560
D5E200	3,480,840	2,149,470	38.72	170,322	420,771	174,056
D10E200	12,002,667	2,905,111	76.18	229,484	546,924	205,238
D10E300	16,246,623	4,594,546	72.14	293,593	853,534	286,391
D10E400	23,975,391	4,530,565	81.25	327,475	881,545	324,886
D20E600	82,624,872	5,299,655	93.67	412,319	1,094,399	439,518
D20E800	106,997,886	6,642,356	93.88	479,619	1,382,052	562,090
D20E1000	133,582,632	5,903,996	95.67	575,308	1,338,588	623,979
Classe	# moyen de contraintes			# moyen de coefficients non nuls		
	GencolBB -MCQ	BB -MCB	GencolBB -MCB	GencolBB -MCQ	BB -MCB	GencolBB -MCB
D2E20	1,549	4,049	4,049	385,162	869,315	407,450
D3E20	2,221	4,609	4,609	374,136	828,527	381,086
D5E20	3,565	6,546	6,546	475,661	972,166	400,716
D5E50	3,872	17,239	17,239	1,723,323	3,626,486	1,330,530
D5E70	4,078	25,212	25,212	2,086,340	5,573,477	1,950,431
D5E200	5,412	53,616	53,616	5,835,139	13,456,350	4,387,935
D10E200	8,772	74,666	74,666	7,702,305	17,144,853	4,691,851
D10E300	9,798	114,441	114,441	10,003,910	26,983,612	6,277,679
D10E400	10,824	112,707	112,707	11,187,585	28,181,544	7,773,432
D20E600	19,596	137,617	137,617	13,840,424	34,912,816	10,857,222
D20E800	21,648	169,009	169,009	16,309,025	44,286,226	14,249,483
D20E1000	23,700	151,717	151,717	19,748,474	43,385,586	17,124,746

Le Tableau 7.2 donne le coût du meilleur calendrier trouvé par chaque méthode de désagrégation post-transfert pour chaque instance du JEU1. Soit Z_C le coût de ce calendrier. De plus, soit $\min Z_C$ le coût du meilleur calendrier identifié par toutes les méthodes de désagrégation. Ainsi, pour chaque instance et chaque méthode, nous calculons la déviation en pourcentage du coût identifié par la méthode par rapport au meilleur coût trouvé (Déviation = $\frac{Z_C - \min Z_C}{\min Z_C}$). Cette déviation donne une idée sur la qualité du calendrier trouvé

Tableau 7.2 – Coûts des calendriers désagrégés identifiés par les trois méthodes de désagrégation post-transfert

Instance	GenColBB -MCQ	BB -MCB	GenColBB -MCB	Instance	GenColBB -MCQ	BB -MCB	GenColBB -MCB
D2E20P1	339.4	339.4	339.4	D10E200P1	1,889.5	1,926.7	1,854.9
D2E20P2	705.5	705.5	708.0	D10E200P2	3,510.5	3,421.3	17,499.5
D2E20P3	570.7	552.1	552.1	D10E200P3	4,038.0	22,688.8	17,874.8
D2E20P4	416.7	416.0	413.7	D10E200P4	3,941.9	4,273.1	17,205.1
D3E20P1	414.0	411.7	411.7	D10E300P1	4,661.0	3,921.4	3,990.1
D3E20P2	803.0	802.0	804.3	D10E300P2	4,143.0	32,183.7	27,005.6
D3E20P3	845.7	838.6	838.6	D10E300P3	5,396.3	34,351.6	27,175.7
D3E20P4	730.8	733.0	732.9	D10E300P4	5,719.9	31,649.7	25,380.9
D5E20P1	477.2	475.1	474.8	D10E400P1	5,935.0	5,088.9	4,534.9
D5E20P2	627.1	622.9	622.5	D10E400P2	37,441.4	34,998.2	30,676.6
D5E20P3	739.6	734.0	736.1	D10E400P3	42,677.8	39,774.7	6,585.0
D5E20P4	800.7	802.9	801.7	D10E400P4	7,220.7	39,520.2	31,060.5
D5E50P1	812.3	804.1	806.6	D20E600P1	42,014.1	41,984.9	39,839.9
D5E50P2	1,329.1	1,260.3	1,265.4	D20E600P2	56,825.6	54,461.1	48,492.9
D5E50P3	1,218.6	1,216.4	1,220.7	D20E600P3	61,365.8	55,489.4	51,530.0
D5E50P4	1,275.9	1,262.1	1,265.8	D20E600P4	57,224.8	50,977.9	46,962.3
D5E70P1	872.4	878.1	877.1	D20E800P1	57,304.2	57,318.2	10,667.7
D5E70P2	1,772.4	1,718.1	1,716.7	D20E800P2	76,210.3	70,504.7	67,224.8
D5E70P3	1,666.0	1,607.6	1,577.1	D20E800P3	81,586.8	70,862.0	64,441.3
D5E70P4	1,242.0	1,236.0	1,227.4	D20E800P4	76,578.3	76,877.3	62,576.3
D5E200P1	2,537.9	2,341.4	2,381.7	D20E1000P1	67,264.3	66,491.6	63,316.9
D5E200P2	3,861.6	3,310.3	3,380.9	D20E1000P2	98,595.7	90,635.2	87,176.7
D5E200P3	3,549.4	3,503.4	3,708.3	D20E1000P3	99,786.0	83,840.5	82,078.3
D5E200P4	3,445.1	3,721.2	3,539.2	D20E1000P4	90,566.9	79,398.1	77,053.4

par une méthode donnée relativement au meilleur calendrier trouvé par toutes les méthodes. Comme précédemment, nous considérons qu'un calendrier est de qualité relative acceptable si cette déviation reste $\leq 5\%$. Les cellules en gris correspondent à des instances résolues avec une qualité relative acceptable. De plus, une cellule apparaît en gras lorsqu'elle correspond au meilleur calendrier trouvé ayant le coût $\min Z_C$.

Le Tableau 7.3 permet de comparer les trois méthodes de désagrégation en se basant sur le coût des calendriers obtenus. Pour chaque catégorie d'instances, pour chaque intervalle et pour chaque méthode de désagrégation, nous calculons le pourcentage d'instances résolues avec une qualité relative appartenant à l'intervalle considéré. Observons que la méthode BB-MCB donne les calendriers les moins coûteux dans la plupart des cas pour les petites et moyennes instances. Ce résultat montre que les décisions sur l'identité des départements receveurs ne détériorent pas forcément la qualité des solutions trouvées. Lorsque la taille des

Tableau 7.3 – Répartition des instances en % selon la qualité relative des solutions obtenues par les trois méthodes de désagrégation post-transfert

Classes	Intervalle (%)	GenColBB-MCQ	BB-MCB	GenColBB-MCB
Petites	[0,1]	91.67	100.00	100.00
]1,5]	8.33	0.00	0.00
]5,10]	0.00	0.00	0.00
]10,20]	0.00	0.00	0.00
	>20	0.00	0.00	0.00
Moyennes	[0,1]	25.00	83.33	66.67
]1,5]	41.67	8.33	25.00
]5,10]	25.00	8.33	8.33
]10,20]	8.33	0.00	0.00
	>20	0.00	0.00	0.00
Grandes	[0,1]	50.00	16.67	33.33
]1,5]	16.67	8.33	8.33
]5,10]	0.00	8.33	0.00
]10,20]	8.33	16.67	0.00
	>20	25.00	50.00	58.33
Très grandes	[0,1]	0.00	0.00	100.00
]1,5]	0.00	33.33	0.00
]5,10]	16.67	41.67	0.00
]10,20]	41.67	8.33	0.00
	>20	41.67	16.67	0.00

instances augmente, la qualité des calendriers fournis par la méthode BB-MCB se détériore en faveur des méthodes utilisant la procédure GenColBB (GenColBB-MCQ et GenColBB-MCB). Ce résultat s'explique par l'augmentation de la taille du problème P restreint à \tilde{Q}_{MCB} résolu par le Branch and Bound de CPLEX dans le cadre de la méthode BB-MCB. Pour les très grandes instances, la méthode GenColBB-MCB devient largement la plus performante de point de vue qualité des solutions obtenues fournissant les calendriers les moins coûteux.

La qualité des calendriers obtenus est influencée par plusieurs facteurs dont fait partie le type de formulation employé, la procédure de résolution du modèle et les limites de temps imposées aux différents éléments composant la méthode de désagrégation. Le Tableau 7.4 illustre le temps de calcul total requis pour compléter l'étape de désagrégation par instance et par méthode. Notons ce temps par CPU. Une cellule de ce tableau est en gris si elle correspond à un calendrier de qualité relative acceptable. De plus, elle est en gras si elle correspond au meilleur temps total min CPU parmi les temps correspondant à des solutions de qualité relative acceptable.

Pour la méthode GenColBB-MCQ, nous avons imposé une limite d'une heure sur toute la

Tableau 7.4 – Temps requis pour compléter l'étape de désagrégation par les trois méthodes de désagrégation post-transfert

Instance	GenColBB -MCQ	BB -MCB	GenColBB -MCB	Instance	GenColBB -MCQ	BB -MCB	GenColBB -MCB
D2E20P1	10.2	14.8	20.0	D10E200P1	5,410.4	5,461.1	5,458.5
D2E20P2	58.8	103.1	41.0	D10E200P2	5,410.4	5,462.7	5,462.5
D2E20P3	71.2	86.4	153.6	D10E200P3	5,410.5	5,542.4	5,536.7
D2E20P4	68.7	72.3	49.0	D10E200P4	5,411.7	5,504.1	5,499.5
D3E20P1	9.4	14.9	17.8	D10E300P1	5,412.5	5,549.4	5,544.9
D3E20P2	12.5	44.8	35.8	D10E300P2	5,413.5	5,643.4	5,632.4
D3E20P3	13.1	39.1	35.4	D10E300P3	5,415.5	5,572.7	5,562.6
D3E20P4	17.3	27.3	29.0	D10E300P4	5,414.2	5,524.4	5,511.9
D5E20P1	24.2	39.5	40.1	D10E400P1	5,416.4	5,622.4	5,616.8
D5E20P2	18.8	35.8	39.8	D10E400P2	5,420.0	5,576.9	5,571.5
D5E20P3	36.7	46.2	59.2	D10E400P3	5,420.4	5,704.1	5,695.0
D5E20P4	39.5	66.9	58.1	D10E400P4	5,419.5	5,566.4	5,555.2
D5E50P1	3,601.6	3,265.3	1,271.6	D20E600P1	5,459.9	6,105.2	6,097.4
D5E50P2	631.0	475.2	594.8	D20E600P2	5,456.7	6,682.3	6,673.2
D5E50P3	864.8	1,841.2	1,218.7	D20E600P3	5,467.2	5,764.1	5,754.6
D5E50P4	481.2	481.5	1,158.4	D20E600P4	5,465.0	6,451.0	6,431.4
D5E70P1	2,661.9	1,410.7	1,078.9	D20E800P1	5,467.0	6,778.3	6,768.3
D5E70P2	3,602.2	4,668.7	4,668.6	D20E800P2	5,476.6	7,917.6	7,909.4
D5E70P3	3,601.7	4,398.5	4,397.7	D20E800P3	5,488.8	7,502.0	7,467.2
D5E70P4	3,602.1	4,452.9	4,451.9	D20E800P4	5,736.6	6,623.0	6,608.3
D5E200P1	3,604.9	5,455.9	5,453.3	D20E1000P1	5,488.3	6,917.5	6,910.9
D5E200P2	3,609.9	5,265.4	5,492.5	D20E1000P2	5,721.0	6,125.1	6,116.7
D5E200P3	3,604.7	5,278.9	5,275.0	D20E1000P3	4,302.2	6,074.5	6,061.7
D5E200P4	3,604.6	5,320.9	5,317.3	D20E1000P4	3,868.7	7,903.5	7,880.0

procédure pour les petites et moyennes instances. Pour les grandes et très grandes instances, nous avons remarqué que la génération de colonnes peut prendre un temps considérable proche ou supérieur à 1 heure ce qui ne laisse pas assez de temps au Branch and Bound de CPLEX pour rétablir l'intégralité des variables de quarts. En conséquence, nous avons imposé une limite de temps sur la génération de colonnes. À chaque début d'itération de la génération de colonnes, si la durée de 40 minutes est dépassée, alors la résolution de la relaxation linéaire du problème maître restreint est interrompue. Dans le cas contraire, le processus itératif est poursuivi. Une limite de 90 minutes est imposée sur le temps total cumulé de la procédure GenColBB en excluant le temps d'énumération des quarts et de chargement du modèle maître restreint initial. C'est pour cette raison que la limite des 90 minutes est légèrement dépassée pour certaines instances.

Pour les méthodes BB-MCB et GenColBB-MCB, une étape intermédiaire s'ajoute qui cor-

respond à la détermination de l'identité des départements receveurs. Cette étape consiste à résoudre la relaxation linéaire du problème P restreint à \tilde{Q}_{MCQ} par la génération de colonnes. L'extrait de cette étape est le sous-ensemble de quarts désagrégés \tilde{Q}_{MCB} vérifiant les restrictions quant aux départements receveurs. Cet ensemble sera exploré par les méthodes BB-MCB et GenColBB-MCB. Pour cette étape intermédiaire, nous interrompons la génération de colonnes à la première itération qui se termine après 20 minutes.

Pour la méthode BB-MCB, nous avons imposé une limite de temps égale à 70 minutes sur le Branch and Bound de CPLEX. Rappelons que le temps requis pour compléter la méthode BB-MCB inclut le temps de l'étape intermédiaire pour la construction de l'ensemble \tilde{Q}_{MCB} et le temps de résolution du problème P restreint à \tilde{Q}_{MCB} . En conséquence, le temps requis pour compléter la méthode BB-MCB doit rester inférieur ou légèrement supérieur à 90 minutes à moins que l'étape intermédiaire dépasse de beaucoup les 20 minutes. C'est le cas des très grandes instances.

Pour la méthode GenColBB-MCB, la génération de colonnes est interrompue à la première itération qui dépasse les 30 minutes. Une limite de 70 minutes est imposée sur le temps total cumulé de la procédure GenColBB en excluant le temps d'énumération des quarts et de chargement du modèle maître restreint initial. Rappelons que le temps total pour compléter la méthode GenColBB-MCB inclut le temps de l'étape intermédiaire pour la construction de l'ensemble \tilde{Q}_{MCB} et la procédure GenColBB pour la résolution du problème P restreint à \tilde{Q}_{MCB} . En conséquence, le temps total maximal doit rester autour de 90 minutes à moins que l'étape intermédiaire dépasse beaucoup la durée des 20 minutes. C'est le cas des très grandes instances. À chaque fois que CPLEX est utilisé pour toutes les méthodes, nous imposons un gap limite de 1%.

Le Tableau 7.5 donne la répartition des temps de calcul pour les solutions avec une qualité relative acceptable ($\text{Gap} \leq 5\%$). Pour chaque instance résolue par une méthode de désagrégation avec une déviation acceptable, nous calculons le meilleur temps de calcul $\min \text{CPU}$. Ensuite, nous calculons l'écart de temps en secondes par rapport à cette valeur $\Delta \text{CPU} = \text{CPU} - \min \text{CPU}$. Finalement, nous calculons pour chaque méthode et chaque catégorie d'instances, la proportion d'instances résolues avec une qualité acceptable durant un temps dont l'écart ΔCPU soit dans un intervalle donné. Observons que pour les petites, moyennes et grandes instances, la méthode GenColBB-MCQ est la plus rapide à compléter l'étape de désagrégation.

Rappelons que la méthode GenColBB-MCQ utilise la procédure GenColBB pour résoudre la formulation basée sur les quarts du problème P restreint à \tilde{Q}_{MCQ} et que la méthode GenColBB-MCB utilise la procédure GenColBB pour résoudre la formulation implicite basée sur les blocs du problème P restreint à \tilde{Q}_{MCB} . Principalement, il existe deux différences entre les deux méthodes : le type de formulation du problème P et l'ensemble des quarts

désagrégés explorés \tilde{Q} . La différence de temps entre ces deux méthodes provient d'une part de la performance du type de formulation, et d'autre part de l'étape intermédiaire qui s'ajoute dans le cas de la méthode GenColBB-MCB pour la sélection du sous-ensemble $\tilde{Q}_{MCB} \subseteq \tilde{Q}_{MCQ}$ selon l'identité des départements receveurs.

Tableau 7.5 – Répartition des instances résolues avec une qualité acceptable en % selon le temps total requis pour compléter l'étape de désagrégation par les trois méthodes de l'heuristique post-transfert

Classes	Intervalle (sec.)	GenColBB-MCQ	BB-MCB	GenColBB-MCB
Petites	=0	20.83	0.00	4.17
]0,60]	4.17	22.92	18.75
]60,300]	0.00	2.08	2.08
]300,600]	0.00	0.00	0.00
]600,2400]	0.00	0.00	0.00
Moyennes	=0	12.50	4.17	8.33
]0,60]	0.00	6.25	0.00
]60,300]	0.00	0.00	4.17
]300,600]	0.00	2.08	2.08
]600,2400]	4.17	10.42	8.33
Grandes	=0	16.67	0.00	8.33
]0,60]	0.00	6.25	2.08
]60,300]	0.00	0.00	0.00
]300,600]	0.00	0.00	0.00
]600,2400]	0.00	0.00	0.00
Très grandes	=0	0.00	0.00	25.00
]0,60]	0.00	8.33	0.00
]60,300]	0.00	0.00	0.00
]300,600]	0.00	0.00	0.00
]600,2400]	0.00	0.00	0.00

En conclusion, à partir des deux tableaux 7.3 et 7.5, la GenColBB-MCB est la méthode la plus performante pour les petites instances, la BB-MCB pour les moyennes instances, la GenColBB-MCQ pour les grandes instances et la GenColBB-MCB pour les très grandes instances. Notons que ce choix accorde la priorité au coût des calendriers obtenus plus qu'au temps de calcul. Globalement, GenColBB-MCB est la méthode qui est la plus performante du moment qu'elle permet d'identifier des solutions avec une déviation $\leq 1\%$ dans 75% des cas pour l'ensemble des 48 instances du JEU1 suivie de la méthode BB-MCB dans 50% des cas et finalement la méthode GenColBB-MCQ dans 41.67% des cas. Ces résultats démontrent que les décisions sur les départements receveurs ne détériorent pas la qualité des solutions trouvées pour notre jeu de données. De plus, la relaxation linéaire donne une information de bonne qualité.

Dans ce qui suit, nous proposons de retenir la méthode GenColBB-MCB pour l'étape de désagrégation de l'heuristique post-transfert.

7.3.2 Comparaison des heuristiques de prétransfert et de post-transfert

Dans cette partie, nous comparons entre les deux heuristiques de prétransfert et de post-transfert. Nous commençons par analyser les résultats obtenus pour les étapes d'agrégation et de désagrégation puis nous passons à une analyse plus globale des performances.

Problème agrégé

Rappelons que les deux heuristiques de pré- et de post-transfert commencent par une étape d'agrégation en définissant et résolvant un problème agrégé. Dans les deux cas, nous considérons l'agrégation des courbes de demande basée sur l'écart absolu et l'agrégation des réglementations en gardant les données réelles les plus proches des multiples de deux heures (c.-à-d. la méthode d'agrégation (NEAR2H,ABS)). De plus, le problème agrégé est tout le temps modélisé à l'aide de la formulation basée sur les quarts MCQ.

Pour faciliter l'analyse, notons par $\bar{P}_{Pré}$ et \bar{P}_{Post} les deux problèmes agrégés correspondant aux deux heuristiques de pré- et de post-transfert respectivement. La différence entre ces deux problèmes se trouve dans la définition des quarts agrégés considérés. Dans le premier cas, l'heuristique de prétransfert utilise tous les quarts internes et externes agrégés admissibles. Cet ensemble de quarts permet de prendre une décision intégrant les deux niveaux décisionnels de construction des heures de travail et des durées et du placement des transferts à l'intérieur des plages horaires travaillées. Dans le deuxième cas, les quarts agrégés considérés se limitent exclusivement aux quarts internes. En effet, l'heuristique de post-transfert commence par le niveau décisionnel quant aux heures de début et durées de travail indépendamment des décisions sur les transferts. Rappelons que le problème agrégé $\bar{P}_{Pré}$ est résolu par la procédure GenColBB et le problème agrégé \bar{P}_{Post} est résolu par la procédure BB. Dans le premier cas, CPLEX est utilisé pour résoudre le problème en nombres entiers restreint aux colonnes considérées par la génération de colonnes au niveau du nœud racine. Le temps limite de la procédure GenColBB est égal à 1 heure. Le gap de CPLEX limite est égal à 1%. Pour l'heuristique de post-transfert, le problème \bar{P}_{Post} est séparable par département. En conséquence, nous résolvons un ensemble de problèmes agrégés $\bar{P}_d, d \in D$ afin de réduire la complexité des calculs. CPLEX est utilisé avec un gap limite de 1% et un temps limite de 10 minutes pour chaque problème \bar{P}_d .

Le Tableau 7.6 donne les informations portant sur la taille du problème agrégé résolu lors de l'étape d'agrégation. Les lignes du tableau correspondent aux classes d'instances du JEU1. Commençons par les colonnes correspondant à l'heuristique de prétransfert. Pour chaque instance, nous calculons le nombre total de quarts agrégés admissibles et le nombre total de variables, de contraintes et de coefficients non nuls du problème agrégé de prétransfert $\bar{P}_{Pré}$.

Tableau 7.6 – Taille du problème agrégé pour les heuristiques de prétransfert et de post-transfert

Classe	# Quarts agrégés		# Moy. Var.		# Moy. Cont.		# Moy. Coeff. non nuls	
	\bar{P}_{Post} ($\sum_{d \in D} / \bar{P}_d$)	$\bar{P}_{Pré}$	\bar{P}_{Post} ($\mu_{d \in D} / \bar{P}_d$)	$\bar{P}_{Pré}$	\bar{P}_{Post} ($\mu_{d \in D} / \bar{P}_d$)	$\bar{P}_{Pré}$	\bar{P}_{Post} ($\mu_{d \in D} / \bar{P}_d$)	$\bar{P}_{Pré}$
D2E20	2,588	9,198	1,463	3,479	177	373	9,957	27,335
D3E20	2,588	11,915	1,032	3,722	146	457	6,694	28,109
D5E20	2,588	22,564	687	4,404	121	625	4,084	31,520
D5E50	6,470	58,320	1,463	10,581	176	932	9,950	84,443
D5E70	9,065	78,831	1,982	14,057	214	1,138	13,889	114,479
D5E200	25,894	221,023	5,348	38,803	454	2,472	39,369	326,981
D10E200	25,894	701,534	2,758	52,172	269	2,892	19,769	436,253
D10E300	38,841	970,203	4,053	79,037	362	3,918	29,569	669,271
D10E400	51,788	1,402,111	5,348	96,471	454	4,944	39,369	815,918
D20E600	77,682	4,827,016	4,053	174,225	362	7,836	29,569	1,475,511
D20E800	103,576	6,268,887	5,348	222,925	454	9,888	39,369	1,901,669
D20E1000	129,470	7,917,509	6,643	275,246	547	11,940	49,170	2,347,864

Pour chaque classe d'instances, nous calculons le nombre moyen de variables, de contraintes et de coefficients non nuls sur l'ensemble des quatre instances issues de cette classe qui diffèrent uniquement par les profils de demande. Le nombre total de quarts est le même pour les quatre instances d'une classe donnée. Passons aux colonnes correspondant à l'heuristique de post-transfert. Pour chaque instance, nous calculons le nombre total de quarts internes agrégés admissibles qui sont considérés par le problème agrégé de post-transfert \bar{P}_{Post} et le nombre moyen de variables, de contraintes et de coefficients non nuls sur l'ensemble des problèmes $\bar{P}_d, d \in D$. Pour chaque classe d'instances, nous calculons le nombre moyen de variables, de contraintes et de coefficients non nuls sur l'ensemble des quatre instances issues de cette classe. Notons que le nombre total de quarts internes est le même pour les quatre instances d'une classe donnée. Il est clair que la complexité du problème agrégé de prétransfert $\bar{P}_{Pré}$ est beaucoup plus grande que celle du problème agrégé de post-transfert \bar{P}_{Post} . D'abord, les quarts agrégés admissibles considérés par le problème $\bar{P}_{Pré}$ incluent, en plus des quarts internes agrégés considérés par le problème \bar{P}_{Post} , tous les quarts externes agrégés admissibles. De plus, le problème \bar{P}_{Post} est séparable par département alors que le problème $\bar{P}_{Pré}$ intègre tous les départements.

La Figure 7.2 donne le temps de calcul, la valeur de la meilleure solution trouvée et le gap de CPLEX des deux problèmes agrégés de prétransfert ($\bar{P}_{Pré}$) et de post-transfert (\bar{P}_{Post}) appliqués aux 48 instances du JEU1. L'axe des abscisses représente les instances triées selon un ordre croissant du nombre de départements puis du nombre d'employés puis de l'indice

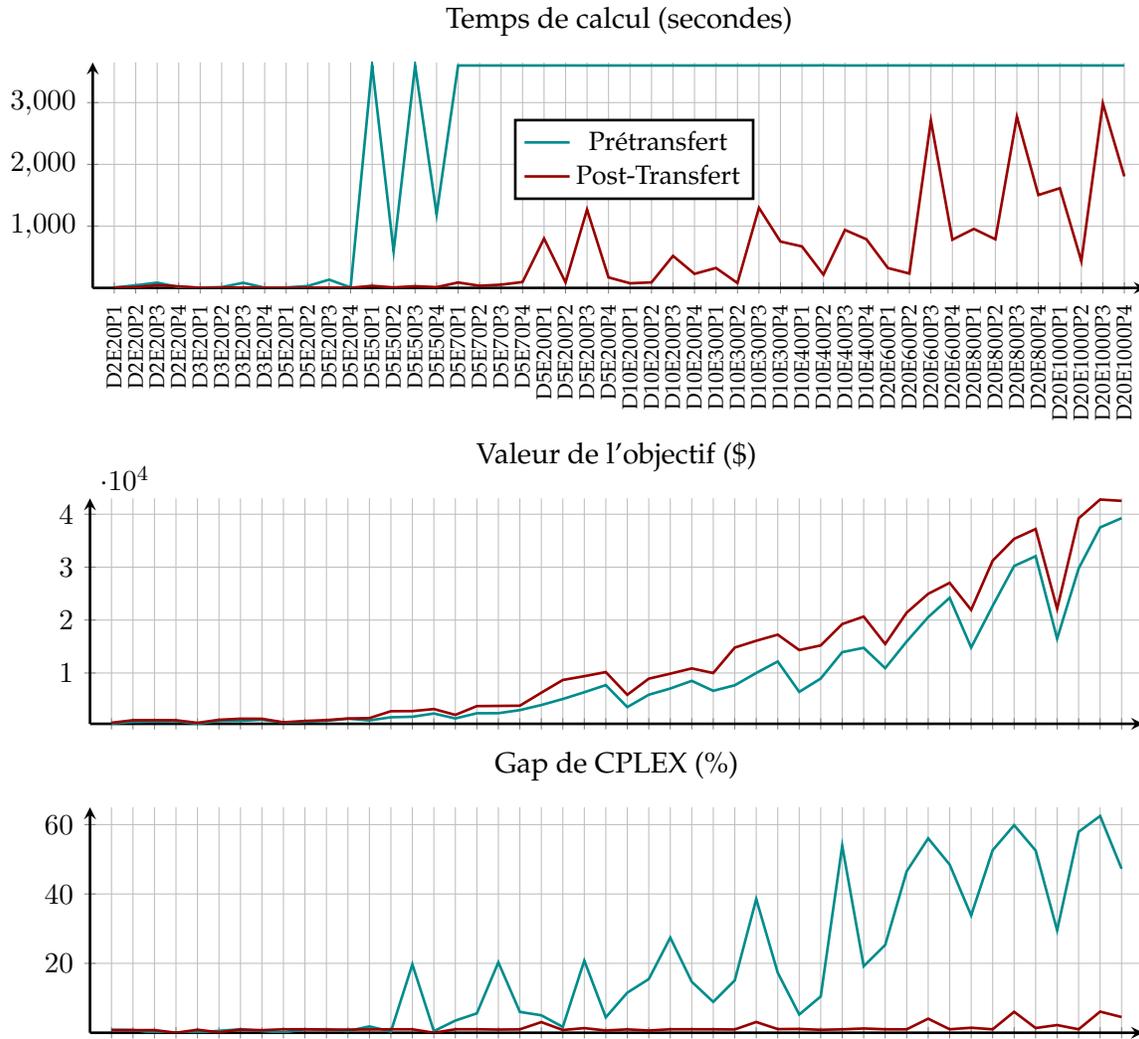


Figure 7.2 – Résultats du problème agrégé pour les heuristiques de prétransfert et de post-transfert

du profil de demande. Le temps de calcul correspond au temps total requis pour compléter l'étape d'agrégation. La valeur de la meilleure solution trouvée correspond à une borne supérieure pour le problème global P du moment que la méthode d'agrégation (NEAR2H,ABS) retient les réglementations initiales les plus proches des multiples de deux heures. Ainsi, les calendriers agrégés obtenus \bar{C} et \bar{C}^{int} correspondant aux heuristiques de prétransfert et de post-transfert, respectivement, sont admissibles pour le problème global P . Pour chaque instance, la figure montre le gap de CPLEX obtenu lors de la résolution du problème agrégé de prétransfert $\bar{P}_{Pré}$ et le gap maximal de CPLEX obtenu lors de la résolution des problèmes $\bar{P}_d, d \in D$ constituant le problème agrégé de post-transfert \bar{P}_{Post} .

Concentrons-nous sur le problème agrégé de prétransfert. Observons qu'à partir des moyennes instances, CPLEX atteint la limite de temps d'une heure lors de la résolution du problème

en nombres entiers au niveau du nœud racine. Le gap moyen de CPLEX est égal à 1% pour les petites instances, 7% pour les moyennes instances, 20% pour les grandes instances et 48% pour les très grandes instances. À l’opposé, CPLEX réussit à résoudre le problème agrégé de post-transfert \bar{P}_{Post} dans un temps maximal de 2,986 secondes pour l’instance D20E1000P3 avec un gap maximal de 6.09% au niveau de l’un des problèmes associé à un département de la même instance.

Alors que les gaps de CPLEX sont moins élevés pour le problème \bar{P}_{Post} que pour le problème $\bar{P}_{Pré}$, le coût du calendrier interne agrégé \bar{C}^{int} solution du problème \bar{P}_{Post} est plus élevé que le coût du calendrier agrégé \bar{C} solution du problème $\bar{P}_{Pré}$. En d’autres termes, même si la complexité numérique du problème $\bar{P}_{Pré}$ est très élevée comparée à celle du problème \bar{P}_{Post} , CPLEX reste capable de construire une solution de coût plus avantageux grâce à l’introduction des quarts agrégés externes permettant le transfert entre départements. D’après le tableau A.17 donné en Annexe A.3, les transferts des employés améliorent nettement le coût des calendriers obtenus.

Problème désagrégé

L’étape de désagrégation consiste à définir et résoudre un problème désagrégé P restreint à un ensemble de quarts \tilde{Q} . Rappelons que l’heuristique de prétransfert utilise deux méthodes de désagrégation selon la taille des instances à traiter. Pour les petites, moyennes et grandes instances, l’heuristique de prétransfert définit un problème désagrégé $P_{Pré}$ en utilisant la formulation basée sur les quarts et le résout par la procédure GenColBB. Pour les très grandes instances, cette procédure devient incapable de pallier à la complexité accrue des organisations visées. Une heuristique de décomposition espace-temps $DecompET(W7,V,P,E,B)$ est utilisée permettant ainsi de résoudre une séquence de sous-problèmes définis le long de l’horizon de planification ($W7$) et associés aux départements $P(I, d)$, $d \in D$ que l’on notera simplement par P_d , $d \in D$.

L’heuristique de post-transfert utilise une seule méthode de désagrégation pour l’ensemble des instances du JEU1. Elle définit un problème désagrégé P_{Post} en utilisant la formulation basée sur les blocs et le résout par la procédure GenColBB.

Le Tableau 7.7 fournit certaines informations sur la taille des problèmes désagrégés $P_{Pré}$ et P_{Post} . Les lignes du tableau correspondent aux 12 classes d’instances du JEU1. Commençons par les colonnes correspondant à l’heuristique de post-transfert. Pour chaque instance, nous calculons le nombre de quarts désagrégés \tilde{Q} ainsi que le nombre de variables de décision, de contraintes et de coefficients non nuls du problème désagrégé P_{Post} . Le tableau donne pour chaque classe d’instances, le nombre moyen de quarts désagrégés et le nombre moyen de variables, de contraintes et de coefficients non nuls calculés sur les quatre instances issues de cette classe. Passons aux colonnes correspondant à l’heuristique de post-transfert. Nous

Tableau 7.7 – Taille du problème désagrégé pour les heuristiques de prétransfert et de post-transfert

Classe	# Quarts désagrégés (\tilde{Q})		# Moy. Var.		# Moy. Cont.		# Moy. Coeff. non nuls	
	P_{Post}	$P_{Pré}$	P_{Post}	$P_{Pré}$	P_{Post}	$P_{Pré}$	P_{Post}	$P_{Pré}$
D2E20	116,859	9,102	16,279	9,271	4,049	1,549	407,450	238,018
D3E20	110,729	9,896	16,862	10,674	4,609	2,221	381,086	241,239
D5E20	136,093	9,746	20,498	13,575	6,546	3,565	400,716	252,116
D5E50	594,581	28,985	56,162	24,112	17,239	3,872	1,330,530	628,545
D5E70	934,002	41,717	80,560	31,644	25,212	4,078	1,950,431	897,482
D5E200	2,149,470	106,712	174,056	75,692	53,616	5,412	4,387,935	2,474,581
D10E200	2,905,111	104,160	205,238	83,041	74,666	8,772	4,691,851	2,501,247
D10E300	4,594,546	170,641	286,391	118,529	114,441	9,798	6,277,679	3,768,773
D10E400	4,530,565	199,757	324,886	150,710	112,707	10,824	7,773,432	4,926,072
	P_{Post}	$P_{Pré}$	P_{Post}	$P_{Pré}$	P_{Post}	$P_{Pré}$	P_{Post}	$P_{Pré}$
		$(\sum_{d \in D} / P_d)$		$(\mu_{d \in D} / P_d)$		$(\mu_{d \in D} / P_d)$		$(\mu_{d \in D} / P_d)$
D20E600	5,299,655	413,670	439,518	23,069	137,617	1,076	10,857,222	567,038
D20E800	6,642,356	553,264	562,090	31,470	169,009	1,235	14,249,483	762,192
D20E1000	5,903,996	678,303	623,979	37,400	151,717	1,342	17,124,746	930,391

distinguons deux cas selon que la désagrégation se fait par la GenColBB ou la DecompET. Pour la première procédure de résolution, les colonnes donnent des informations similaires à celles de l'heuristique post-transfert. Il s'agit de la première partie du tableau associée aux petites, moyennes et grandes instances. Pour la deuxième procédure de résolution du problème $P_{Pré}$, nous calculons pour chaque instance, le nombre total de quarts désagrégés formant l'ensemble \tilde{Q} et le nombre moyen de variables, de contraintes et de coefficients non nuls des problèmes $P_d, d \in D$. Les colonnes donnent pour chaque classe la moyenne de ces valeurs sur les quatre instances issues de cette classe.

Observons que la complexité des problèmes désagrégés de l'heuristique de prétransfert est beaucoup plus petite que celle des problèmes désagrégés de l'heuristique de post-transfert. Ce résultat peut être expliqué par deux facteurs. D'abord, le schéma d'optimisation distribué de l'heuristique de prétransfert utilise l'étape d'agrégation pour prendre certaines décisions quant aux transferts ce qui limite les options de placement des transferts lors de l'étape de désagrégation. À l'opposé, l'heuristique de post-transfert exclut les transferts de l'étape d'agrégation et passe à définir un ensemble plus large de quarts désagrégés afin de questionner les multiples façons de placer les transferts compte tenu de la décision agrégée sur les plages travaillées. Le deuxième facteur correspond à l'heuristique de décomposition espace-temps qui permet de diviser le problème désagrégé en un ensemble de sous-problèmes de complexité moindre. D'ailleurs, cette décomposition est possible grâce à l'information pro-

venant des transferts agrégés.

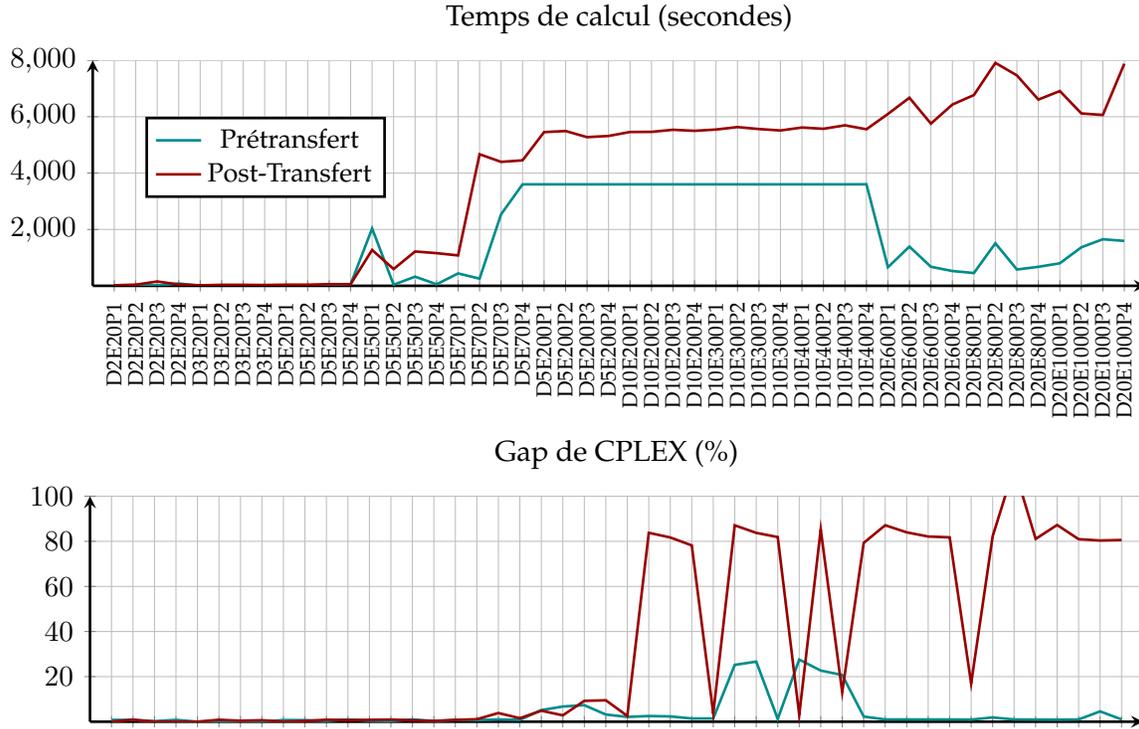


Figure 7.3 – Résultats du problème désagrégé pour les heuristiques de prétransfert et de post-transfert

La Figure 7.3 donne certaines informations portant sur le temps de calcul et les gaps de CPLEX obtenus lors de l'étape de désagrégation pour la résolution des problèmes désagrégés $P_{Pré}$ et P_{Post} . L'axe des abscisses représente les 48 instances du JEU1 triées selon un ordre croissant de leur complexité numérique. Le temps de calcul correspond au temps total requis pour compléter l'étape de désagrégation. La figure montre le gap de CPLEX obtenu lors de la résolution du problème désagrégé de post-transfert P_{Post} pour les 48 instances et du problème désagrégé de prétransfert $P_{Pré}$ pour les 36 petites, moyennes et grandes instances. Rappelons qu'il s'agit du problème désagrégé en nombres entiers obtenu au niveau du nœud racine par la procédure GenColBB. Pour les très grandes instances, l'heuristique de prétransfert utilise la méthode de décomposition DecomPET pour définir un ensemble de sous-problèmes $P_d, d \in D$ résolus par le Branch and Bound de CPLEX. Dans ce cas, le gap donné par la figure correspond au gap maximal de CPLEX obtenu lors de la résolution des problèmes $P_d, d \in D$ constituant le problème désagrégé de prétransfert $P_{Pré}$.

La figure montre clairement que la désagrégation prétransfert requiert un temps moins élevé que celui correspondant à la désagrégation post-transfert. Nous remarquons aussi une différence significative des gaps de CPLEX. Dans le cas de l'heuristique de post-transfert, ces gaps sont beaucoup plus élevés que ceux de l'heuristique de prétransfert. Notons que c'est

le résultat opposé de ce qui a été souligné lors de l'étape d'agrégation caractérisée par une complexité numérique plus élevée pour le problème $\bar{P}_{Pré}$ que pour le problème \bar{P}_{Post} .

Comparaison globale

Dans le cadre de cette thèse, nous avons proposé deux heuristiques : une heuristique de pré-transfert et une heuristique de post-transfert. Le Tableau 7.8 donne le temps total requis et la valeur du calendrier trouvé par chaque heuristique et pour chacune des 48 instances du JEU1. Notons par $CPU_{Pré}$ et CPU_{Post} les temps requis pour compléter les heuristiques de pré- et de post-transfert respectivement. De plus, soit $Z_{Pré}$ et Z_{Post} les coûts des meilleurs calendriers trouvés par les deux heuristiques de pré- et de post-transfert respectivement. Le tableau montre clairement que l'heuristique de prétransfert affiche une meilleure performance que l'heuristique de post-transfert.

Tableau 7.8 – Comparaison globale entre les deux heuristiques de pré- et de post-transfert

Instance	CPU		Valeur de l'obj.		Instance	CPU		Valeur de l'obj.	
	Pré.	Post.	Pré.	Post.		Pré.	Post.	Pré.	Post.
D2E20P1 (1)	8.6	25.4	290.0	339.4	D10E200P1 (1)	3,710.2	5,533.6	1,448.0	1,828.7
D2E20P2 (1)	46.1	51.8	501.1	708.0	D10E200P2 (1)	3,693.4	5,551.7	2,952.0	17,498.4
D2E20P3 (1)	87.4	198.5	483.3	552.1	D10E200P3 (1)	3,727.7	6,055.7	3,437.1	17,901.6
D2E20P4 (1)	15.9	74.0	405.9	413.7	D10E200P4 (1)	3,717.0	5,727.2	3,549.4	17,202.3
D3E20P1 (1)	6.8	19.1	292.5	411.7	D10E300P1 (1)	3,808.4	5,867.9	3,193.6	3,969.8
D3E20P2 (1)	14.8	39.8	713.7	804.3	D10E300P2 (1)	3,833.6	5,711.6	3,705.8	27,021.8
D3E20P3 (2)	85.2	41.5	641.4	838.6	D10E300P3 (1)	3,874.2	6,859.5	5,528.2	27,172.3
D3E20P4 (1)	5.2	30.0	709.3	732.9	D10E300P4 (1)	3,822.3	6,262.6	4,802.6	25,450.3
D5E20P1 (1)	9.2	41.0	418.6	474.1	D10E400P1 (1)	3,903.7	6,288.2	4,101.4	4,505.4
D5E20P2 (1)	32.0	42.1	595.2	622.4	D10E400P2 (1)	3,900.3	5,785.6	4,697.0	30,673.5
D5E20P3 (2)	136.6	61.3	654.0	735.7	D10E400P3 (1)	3,900.4	6,633.3	5,943.0	6,560.7
D5E20P4 (3)	9.5	59.2	813.4	801.5	D10E400P4 (1)	3,852.8	6,342.7	5,830.1	31,053.5
D5E50P1 (2)	3,613.2	1,305.2	556.8	806.6	D20E600P1 (1)	4,258.9	6,419.5	5,934.2	39,925.3
D5E50P2 (2)	602.3	602.2	968.0	1,265.2	D20E600P2 (1)	4,991.8	6,905.8	8,648.0	48,498.3
D5E50P3 (2)	3,610.8	1,246.6	916.9	1,220.9	D20E600P3 (1)	4,276.8	8,462.4	11,477.8	51,525.3
D5E50P4 (2)	1,204.8	1,170.4	1,030.8	1,265.4	D20E600P4 (1)	4,125.3	7,212.8	11,310.6	46,961.7
D5E70P1 (2)	3,626.7	1,166.9	661.7	876.3	D20E800P1 (1)	4,054.1	7,723.6	8,652.1	10,645.2
D5E70P2 (1)	3,620.0	4,703.1	1,340.1	1,714.7	D20E800P2 (1)	5,110.6	8,696.4	11,631.4	67,224.6
D5E70P3 (1)	3,630.6	4,450.2	1,223.0	1,576.7	D20E800P3 (1)	4,179.6	10,237.9	16,752.0	64,510.1
D5E70P4 (1)	3,626.3	4,547.1	1,035.7	1,226.3	D20E800P4 (1)	4,275.2	8,111.9	14,946.4	62,596.8
D5E200P1 (1)	3,718.9	6,253.0	1,538.8	2,380.2	D20E1000P1 (1)	4,400.7	8,525.2	9,582.0	63,349.2
D5E200P2 (1)	3,698.2	5,580.9	2,538.2	3,381.8	D20E1000P2 (1)	4,967.6	6,549.4	16,146.3	87,188.6
D5E200P3 (1)	3,728.8	6,543.1	3,015.4	3,706.9	D20E1000P3 (1)	5,253.1	9,047.9	21,110.1	82,073.8
D5E200P4 (1)	3,690.3	5,487.2	3,056.5	3,535.8	D20E1000P4 (1)	5,194.0	9,685.6	16,813.2	77,052.8

Soit ΔCPU l'écart du temps de calcul de l'heuristique de post-transfert par rapport à l'heuristique de prétransfert ($\Delta CPU = CPU_{Post} - CPU_{Pré}$). Soit de plus Gap l'écart en pourcentage du coût du calendrier identifié par l'heuristique de post-transfert par rapport au coût

du calendrier identifié par l'heuristique de prétransfert $(\frac{Z_{Post} - Z_{Pré}}{\min\{Z_{Post}, Z_{Pré}\}})$. Nous pouvons distinguer les trois situations différentes :

- Dans 83.33% des instances (marquées par (1)), l'heuristique de prétransfert affiche des calendriers moins coûteux dans des temps de calcul plus courts que ceux de l'heuristique de post-transfert. Dans ce cas, le gain moyen en temps de calcul est de l'ordre de 35 minutes ($\Delta CPU = 2,037.8$ secondes en moyenne). L'écart moyen en pourcentage de coût est très élevé ($Gap = 215\%$ en moyenne).
- Dans 14.58% des instances (marquées par (2)), l'heuristique de prétransfert affiche des calendriers moins coûteux dans des temps de calcul plus longs que ceux de l'heuristique de post-transfert. Dans ce cas, l'heuristique de prétransfert converge après un temps supérieur en moyenne de 1,040.8 secondes au temps de calcul de l'heuristique de post-transfert. Ce temps additionnel est justifié du moment que l'on puisse identifier des solutions avec un gain moyen de coût égal à 29.59%.
- Pour une seule instance (marquées par (3)), l'heuristique de prétransfert affiche un calendrier légèrement plus coûteux ($Gap = -1.48\%$) dans un temps relativement plus court que l'heuristique de post-transfert ($\Delta CPU = 49.7$ secondes).

En conclusion, les contributions principales de ce chapitre peuvent être listées comme suit :

- Nous proposons une approche post-transfert pour le partage a posteriori des effectifs entre les départements d'une organisation donnée. Cette approche utilise une formulation implicite basée sur les blocs que nous avons proposés dans le Chapitre 4 de cette thèse. De plus, cette nouvelle formulation est résolue à l'aide de la procédure de génération de colonnes suivie par le Branch and Bound de CPLEX que nous proposons. Quoique cette heuristique affiche une mauvaise performance par rapport à l'heuristique de résolution prétransfert, elle demeure pertinente pour d'autres contextes. En effet, dans bon nombre de cas, les organisations se trouvent dans l'obligation de changer les calendriers de travail déjà convenus avec leurs employés. L'heuristique de post-transfert permet de perturber légèrement les heures de début et les durées de travail des employés évitant ainsi des changements de grande ampleur sur l'horaire des employés. Par contre, la décision portant sur le contenu des quarts en termes de départements est une décision qui peut être changée sans avoir un grand impact sur la vie privée de l'employé. En conséquence, nous pouvons envisager de placer les départements à l'intérieur des quarts perturbés des employés en respectant les règles ponctuelles de travail et de transfert. L'heuristique de prétransfert peut être améliorée surtout pour le cas des grandes organisations en développant un Branch and Price pour une résolution plus efficace et plus efficiente du problème.
- La contribution principale de cette thèse est de démontrer empiriquement que la décision portant sur le partage des ressources en effectifs entre les différents départements doit être intégrée dans un même niveau décisionnel que la construction des quarts.

Pour simplifier le problème obtenu, la technique d'agrégation permet de définir un problème approximatif permettant d'identifier des transferts à raffiner dans l'étape de désagrégation. À l'opposé, la séparation décisionnelle entre la construction des quarts et le placement à posteriori des départements, même en présence d'une procédure de révision des décisions déjà établies sur les plages travaillées, détériore largement la qualité des calendriers obtenus. Nous avons montré qu'un calendrier agrégé externe avec un gap de CPLEX qui n'est pas forcément bon vaut mieux qu'un calendrier agrégé interne optimal suivi d'un placement à posteriori des transferts.

- Cette thèse contient un ensemble de techniques d'optimisation distribuée permettant à l'industrie des logiciels de gestion des horaires de proposer des solutions aux grandes organisations ayant un nombre d'employés qui peut atteindre jusqu'à 1000. Soulignons que l'heuristique de prétransfert supporte un degré de parallélisme lors de la résolution des sous-problèmes associés aux départements de l'organisation. Ceci permet donc de concevoir des logiciels efficaces fournissant des solutions dans des temps de calcul raisonnables.
- En se basant sur les travaux de cette thèse, une perspective future serait d'appliquer le concept d'agrégation pour la résolution des problèmes de construction simultanée de quarts et de cycles. D'ailleurs, la technique d'agrégation ouvre de nouvelles perspectives pour l'optimisation des horaires dans les grandes organisations.

Conclusion

Les problèmes de construction d'horaires de travail ont été dûment étudiés par la communauté scientifique durant ces dernières décennies. Les premiers travaux dans ce domaine proposaient des approches de modélisation et de résolution qui reposent principalement sur la séparation entre les niveaux décisionnels du processus de construction d'horaires. Avec le progrès scientifique, les travaux évoluaient selon les axes suivants : l'intégration des niveaux décisionnels, l'augmentation des formes de flexibilité et la multiplicité des réglementations de travail.

Dans ce cadre, cette thèse vise principalement à développer des approches de modélisation et de résolution permettant de résoudre un problème générique de construction d'horaires. De par sa définition, ce problème implique un haut niveau de flexibilité et d'intégration lui permettant d'être applicable à une large panoplie d'organisations. Notre problème s'insère dans le cadre des travaux de construction d'horaires à moyen et court terme. La taille de la main d'œuvre est connue. L'objectif est de construire des quarts personnalisés de travail sur un horizon continu et multi-jour pour une main d'œuvre hétérogène et possédant des qualifications multiples.

Cette thèse est dédiée au contexte multidépartement permettant le partage de ressources entre les départements d'une organisation. Les pratiques courantes dans certaines organisations consistent à résoudre le problème de construction d'horaires pour chaque département séparément. Une fois les employés inactifs dans leurs départements d'origine identifiés, ils sont redirigés temporairement aux départements goulots. Le placement a posteriori des transferts est justifié par la complexité du problème dans le contexte multidépartement permettant le partage de ressources. Cette thèse permet d'améliorer les pratiques des organisations en offrant des approches de modélisation et de résolution du problème intégré de pilotage centralisé des horaires.

Le premier axe de notre thèse porte sur l'aspect de modélisation du problème. Nous avons proposé principalement quatre formulations : la formulation basée sur les horaires, la formulation basée sur les quarts et deux formulations basées sur les blocs. Le problème est initialement résolu avec la procédure de séparation et d'évaluation progressive. Nous obtenons les conclusions suivantes. La formulation classique basée sur les horaires n'est pas

une option viable pour traiter le problème. La formulation basée sur les quarts est une alternative permettant de modéliser toute sorte de règles ponctuelles, c'est à dire, portant sur la définition des quarts. Cette formulation permet de résoudre le problème pour des instances de petite taille dans des temps relativement courts.

Ces dernières années, plusieurs travaux ont généralisé le concept classique de contraintes forward et backward pour modéliser implicitement certaines formes de flexibilité. À notre connaissance, nous sommes les premiers à généraliser ce concept au contexte multidépartement. C'est dans ce cadre que s'insèrent les nouvelles formulations basées sur les blocs qui reposent sur la définition de problèmes de transport spécifiques. La structure particulière de ces problèmes se prête bien à modéliser la flexibilité portant sur les transferts. En effet, au lieu de supposer que le transfert est permis par quart au complet tel est le cas de la plupart des travaux rencontrés, notre travail permet un transfert par bloc de travail.

La première formulation basée sur les blocs divise le quart en deux parties : avant et après le déplacement entre les départements de début et de fin de travail. Les contraintes forward et backward sont utilisées pour lier entre les segments de début et de fin pour assurer la réalisabilité des quarts de point de vue la composante temporelle, c'est à dire les heures de début et les durées de travail. D'autres contraintes additionnelles permettent la liaison entre les blocs de début et de fin de point de vue la composition en départements. Une autre manière de lier les blocs selon la composition en départements est de considérer deux graphes de transport distincts. Les résultats montrent que la première formulation basée sur les blocs devient plus prometteuse que celle basée sur les quarts lorsque le nombre de qualifications externes augmente. Une perspective future serait de questionner la performance de cette formulation par rapport à celle basée sur les quarts en développant une procédure de Branch-and-Price.

Pour améliorer la première formulation basée sur les blocs, nous avons introduit un nouveau modèle qui diminue le nombre de blocs de début énumérés. L'idée est de modéliser le quart en plaçant un bloc de fin à l'intérieur d'un quart monodépartement. Cette formulation peut être appliquée lorsque les départements receveurs sont connus a priori. Notre formulation peut être donc utilisée dans ce genre de situations ou lorsqu'elle est introduite dans une approche de résolution plus globale.

Notons que les modèles implicites que nous proposons peuvent être généralisés et appliqués à plusieurs contextes par une modification légère dans la définition des problèmes de transport. La première généralisation consiste à traiter le contexte multi-site permettant de tenir compte des coûts et des temps liés aux déplacements entre les sites d'une organisation. Les problèmes de transport sont légèrement modifiés pour tenir compte des temps de transfert. Une autre généralisation serait de tenir compte du placement des pauses en permettant le changement du département de travail uniquement entre les pauses et les jours de repos. Une troisième généralisation serait d'appliquer le problème au contexte monodépartement

et multiactivité. Dans ce cas, les segments de travail avant et après la pause peuvent être remplis par des séquences d'activités explicitement énumérées.

Le deuxième axe de notre thèse porte sur l'aspect de résolution du problème. En effet, lorsque la taille des organisations est grande, la résolution directe des formulations proposées est exclue. Cette thèse propose une procédure basée sur la génération de colonnes suivie par un Branch-and-Bound. Les tests expérimentaux montrent que cette procédure permet d'obtenir des solutions proches de l'optimum avec des gains considérables en temps de calcul. Un axe de recherche serait d'étendre notre procédure basée sur la génération de colonnes pour concevoir un algorithme plus efficace de Branch-and-Price. Ceci permettra, non seulement de résoudre des problèmes de plus grande taille, mais aussi de comparer entre les formulations proposées lorsque la taille des organisations augmente.

La contribution principale de notre thèse est de concevoir un outil très performant pour la construction d'horaires avec partage de ressources. Nous avons proposé deux approches : une approche de résolution prétransfert qui intègre la construction des quarts et le placement de blocs et une approche de résolution post-transfert qui commence par la construction de quarts et place a posteriori les transferts. Un constat s'impose : la performance de l'approche de résolution prétransfert dépasse largement la performance de l'approche de résolution post-transfert. Notons que l'approche prétransfert a réussi à identifier de meilleures solutions pour des instances impliquant 20 départements et 1000 employés en moins d'une heure et demi. L'intégration de la construction des quarts et le placement des transferts permet d'obtenir des calendriers qui coûtent en moyenne 215% moins chers que les calendriers identifiés par l'approche de résolution post-transfert dans 88.33% des cas avec un gain moyen en temps de calcul égal à 35 minutes.

L'approche de résolution prétransfert peut être améliorée selon plusieurs axes. Le premier axe de recherche serait de résoudre les sous-problèmes associés aux fenêtres de temps et aux départements d'une manière plus efficace en utilisant une procédure de Branch-and-Price. Ceci permettrait d'augmenter la qualité et le temps de calcul des calendriers finaux obtenus. Le deuxième axe de recherche que nous proposons est de généraliser la procédure de décomposition espace-temps au cas général de présence de transferts cycliques entre les départements. Au lieu de sacrifier certains transferts afin de supprimer les circuits dans les graphes de transfert, le défi serait de partitionner l'ensemble des départements en des sous-ensembles mutuellement disjoints de sorte que les transferts cycliques n'existent pas entre les partitions de départements. Il faut donc concevoir un algorithme qui permet un tel partitionnement et une procédure qui permet de résoudre un sous-problème restreint à une fenêtre de temps et une partition de départements.

Une autre contribution de notre travail est l'introduction de la technique d'agrégation. À notre connaissance, nous sommes les premiers à appliquer cette technique aux problèmes

de construction d'horaires. Nous avons proposé plusieurs méthodes afin de faire ressortir le meilleur compromis entre la précision des données et la complexité du problème. Cette méthode est prometteuse. En effet, elle offre la possibilité de traiter des problèmes de construction d'horaires hautement intégrés selon une stratégie en deux étapes. La première étape intègre plusieurs niveaux décisionnels et simplifie le problème en considérant une période de discrétisation grossière. La deuxième étape se base sur l'information recueillie du niveau agrégé et passe à raffiner la solution en remettant en question certaines décisions tout en rétablissant la précision de la période de discrétisation. Une perspective future serait de résoudre un problème de construction simultanée de cycles et de quarts de travail dans un contexte multidépartement en se basant sur cette technique.

Nous avons démontré que l'approche de résolution post-transfert n'est pas performante pour la résolution du problème étudié. Ceci dit, cette approche permettrait de résoudre un problème qui suscite un intérêt grandissant des chercheurs. Ce problème de reconstruction d'horaires de travail offre un outil d'aide à la décision pour l'ajustement des horaires lorsque les intrants du problème, supposés déterministes, changent. Cette situation peut se présenter lorsqu'un employé s'absente ou lorsque la demande s'écarte des prévisions initiales. Supposons que le planificateur dispose d'un horaire construit précédemment sur un horizon de planification de plusieurs semaines. Rappelons que nous traitons un problème de construction d'horaires sur un horizon de quelques jours. Ainsi, le planificateur peut utiliser notre approche pour construire un nouvel horaire révisé. Au début de chaque semaine, les intrants du problème sont connus d'une manière plus précise (réservations des clients, départs en congé de maladie, etc). Le planificateur peut donc utiliser notre heuristique en considérant les nouveaux intrants plus précis. Nous avons proposé trois méthodes différentes pour la résolution du problème et nous avons montré que la formulation améliorée basée sur les blocs donne les meilleures performances lorsqu'elle est résolue à l'aide de notre procédure basée sur la génération de colonnes. L'information portant sur les départements receveurs est déterminée par la formulation basée sur les quarts. Le problème traité dans cette thèse est déterministe. La procédure de résolution post-transfert permet donc de questionner la robustesse des calendriers obtenus face à des imprévus.

Quoique l'hypothèse qui suppose que chaque département soit monoactivité semble être restrictive, ce travail peut être appliqué au cas multiactivité par département. Une approche séquentielle en deux phases sépare entre les niveaux décisionnels : construction de quarts avec transferts entre les départements et affectation des activités aux périodes travaillées. Dans un tel contexte, la demande en effectif est spécifiée pour chaque période de l'horizon de planification et pour chaque activité exécutée au sein de chaque département. Dans un premier niveau, on calcule le nombre total d'employés requis pour chaque période toute activité confondue exécutée dans le même département. Cette agrégation permet d'obtenir une seule courbe de demande par département. On passe à utiliser les algorithmes proposés

dans cette thèse afin de construire des quarts avec des transferts entre les départements sans spécifier les activités exécutées par les employés. Dans un deuxième niveau, un ensemble de sous-problèmes séparables est défini par département. Chaque sous-problème consiste à affecter les activités aux périodes travaillées déduites des quarts multidépartements construits précédemment.

Finalement, une autre perspective serait de généraliser l'ensemble de nos approches de modélisation et de résolution au cas stochastique pour proposer un outil de construction d'horaires robustes.

Annexe A

Annexes

A.1 Choix de la méthode d'agrégation

Les Tableaux de l'Annexe A.1 donnent certaines informations supplémentaires sur les problèmes résolus \bar{P} et P restreint à $V(\bar{C})$ lors de l'évaluation des méthodes d'agrégation. Pour chaque méthode d'agrégation et pour chaque instance, la cellule correspondante apparaît en gris si la méthode d'agrégation réussit à identifier une solution de qualité relative acceptable. Rappelons qu'un calendrier C , solution du problème P restreint à $V(\bar{C})$, est de qualité acceptable si la déviation en pourcentage entre son coût Z_C par rapport au meilleur coût trouvé par les six méthodes d'agrégation $\min Z_C$ est inférieure ou égale à 5%.

A.1.1 Problème agrégé

Rappelons que le problème agrégé P est résolu par la procédure GenColBB. Rappelons aussi que cette procédure est basée sur la génération de colonnes pour sélectionner un ensemble restreint de quarts. Ensuite, les contraintes d'intégralité du problème maître restreint au niveau du nœud racine sont rétablies et le problème obtenu est résolu par le Branch and Bound de CPLEX. Le Tableau A.1 donne les gaps de CPLEX qui évaluent la qualité du calendrier agrégé \bar{C} obtenu par la résolution du problème agrégé \bar{P} restreint aux quarts sélectionnés par la génération de colonnes. Dans le tableau, une cellule apparaît en gras si elle correspond au calendrier agrégé \bar{C} menant au meilleur voisinage $V(\bar{C})$. On entend dire par le meilleur voisinage celui qui a permis de calculer le meilleur calendrier désagrégé C parmi l'ensemble des calendriers obtenus par les six méthodes d'agrégation. Rappelons que le calendrier C est obtenu par la résolution du problème désagrégé P restreint à ce voisinage $V(\bar{C})$.

Tableau A.1 – Gaps de CPLEX du problème agrégé \bar{P} restreint aux quarts sélectionnés par la génération de colonnes

Instance	(AV,2H)	(ABS,2H)	(MED,2H)	(AV,NEAR2H)	(ABS,NEAR2H)	(MED,NEAR2H)
D2E20P1	0	0	0	0.74	0.84	0.87
D2E20P2	0.64	0.04	0.64	0.23	0.81	0.23
D2E20P3	0	0.93	1	0.99	0	1
D2E20P4	1	0.25	0	0	0.03	0.99
D3E20P1	0	0	0	0.99	0.03	1
D3E20P2	0	0	0	0.74	0.56	0.89
D3E20P3	0	0	0	1	1	1
D3E20P4	0.76	0.38	0.01	0.27	0.49	1
D5E20P1	0	0	0	0	0.08	0.53
D5E20P2	0	0	0.99	1	1	0.96
D5E20P3	0	0.72	0	0.57	1	0.83
D5E20P4	0.98	0.95	0	1	0.48	1
D5E50P1	0.49	0	0	0.01	1.8	0.99
D5E50P2	1	0.41	0.06	0.92	0.35	1
D5E50P3	0.92	0.87	0.61	19.62	19.7	23.35
D5E50P4	0	0	0.49	1.29	0.45	4.59
D5E70P1	0	0	0	7.36	3.51	5.66
D5E70P2	0.02	0.48	0.55	3.11	5.27	2.97
D5E70P3	1.16	0	12.16	14.11	20.3	19.76
D5E70P4	0	0	0.53	3.44	6.66	16.27
D5E200P1	0	0	0	6	3.41	6.35
D5E200P2	0.56	0	0.01	1.82	1.58	6.79
D5E200P3	7.59	0.94	4.32	20.15	19.91	22.62
D5E200P4	0.99	0.55	7.93	11.59	4.81	16.06

A.1.2 Problème désagrégé restreint au voisinage du calendrier agrégé

Le Tableau A.2 donne les temps de calcul requis pour définir et résoudre le problème désagrégé P restreint au voisinage $V(\bar{C})$ en utilisant la procédure GenColBB. Ces données sont représentées selon le même format du tableau 6.5. Dans ce tableau, minCPU (meilleur temps écoulé pour résoudre le problème désagrégé P restreint à $V(\bar{C})$ en identifiant une solution de qualité relative acceptable) est affiché en gras.

Tableau A.2 – Temps de calcul requis du problème désagrégé P restreint au voisinage $V(\bar{C})$ pour les six méthodes d'agrégation testées

Instance	(AV,2H) (ABS,2H) (MED,2H) (AV,NEAR2H) (ABS,NEAR2H) (MED,NEAR2H)					
	CPU du problème désagrégé P restreint au voisinage $V(\bar{C})$ (secondes)					
D2E20P1	3	3	3	12	12	5
D2E20P2	12	15	11	125	27	125
D2E20P3	9	29	8	59	11	19
D2E20P4	39	22	56	50	80	45
D3E20P1	19	41	10	6	9	6
D3E20P2	8	26	5	24	9	9
D3E20P3	8	11	7	13	15	6
D3E20P4	5	4	8	11	10	7
D5E20P1	5	7	5	6	13	5
D5E20P2	5	5	7	8	9	15
D5E20P3	62	12	6	9	19	13
D5E20P4	23	24	34	50	53	33
D5E50P1	38	163	50	95	469	121
D5E50P2	49	24	20	30	38	45
D5E50P3	223	113	236	598	203	66
D5E50P4	96	105	35	53	52	63
D5E70P1	336	129	1,194	1,331	439	103
D5E70P2	461	221	266	188	427	185
D5E70P3	1,068	451	270	1,823	2,524	592
D5E70P4	501	2,827	2,002	376	529	189
D5E200P1	3,319	3,138	3,310	3,601	3,601	3,470
D5E200P2	2,559	3,600	3,457	3,600	3,601	3,601
D5E200P3	3,600	3,600	3,600	3,601	3,600	3,601
D5E200P4	3,600	3,600	3,600	3,601	3,601	3,601
Intervalle	%Ins. : $\frac{Z_C - \min Z_C}{\min Z_C} \leq 5\%$ et $\Delta \text{CPU} = \text{CPU} - \min \text{CPU} \in \text{Intervalle}$					
[0,0]	8.3	4.2	16.7	29.2	20.8	37.5
]0,60]	8.3	8.3	0.0	37.5	54.2	41.7
]60,120]	0.0	0.0	0.0	4.2	0.0	4.2
]120,360]	0.0	0.0	0.0	0.0	16.7	0.0
]480,1980]	0.0	0.0	0.0	12.5	4.2	0.0

Le Tableau A.3 donne pour chaque instance considérée et chaque méthode d'agrégation le gap de CPLEX correspondant au problème en nombres entiers P restreint aux quarts voisins dans $V(\bar{C})$ sélectionnés par la génération de colonnes. Ces données sont représentées selon le même format du tableau A.1. Ce tableau renseigne sur la qualité de la borne supérieure du problème original P . Bien entendu, un gap de CPLEX $\leq 1\%$ ne peut pas garantir que la solution soit de bonne qualité mais une valeur élevée peut amener à critiquer la qualité de

cette borne supérieure.

Tableau A.3 – Gaps de CPLEX du problème désagrégé P restreint aux quarts voisins sélectionnés par la génération de colonnes

Instance	(AV,2H)	(ABS,2H)	(MED,2H)	(AV,NEAR2H)	(ABS,NEAR2H)	(MED,NEAR2H)
D2E20P1	0.01	0.01	0	0.02	0.82	0.8
D2E20P2	0.98	0.89	0.98	0.91	0.79	0.91
D2E20P3	0.43	0.87	0	0.94	0.25	0.96
D2E20P4	0.47	0.73	0.99	0.94	0.82	0
D3E20P1	0.71	0.92	0.67	0	0.01	0
D3E20P2	0.36	0.92	0.89	0.77	0.32	0.94
D3E20P3	0.01	0.27	0.15	0.72	0	0.73
D3E20P4	0.41	0.09	0.25	0.69	0.01	0.84
D5E20P1	0.77	1	0.84	0.81	0.74	0
D5E20P2	0.33	0.33	0.22	0.7	0.7	0.97
D5E20P3	1	0.68	0.11	0.21	0.31	0.12
D5E20P4	0.97	1	0.99	0.95	1	1
D5E50P1	0.77	0.9	0.74	0.02	0.8	1
D5E50P2	0.72	0.49	0.51	0.9	0.25	0.66
D5E50P3	0.96	1	0.2	0.92	1	0.27
D5E50P4	0.82	0.9	0.19	0.91	0.32	0.96
D5E70P1	0.94	0.9	0.81	0.7	0.72	0.89
D5E70P2	0.74	0.38	0.76	0.59	0.99	0.95
D5E70P3	0.81	0.98	0.59	0.96	1	0.76
D5E70P4	0.79	0.81	0.6	1	0.96	0.78
D5E200P1	0.79	0.13	0.93	12.74	7.44	0.92
D5E200P2	0.92	1.88	0.51	6.61	5.79	1.15
D5E200P3	4.11	2.71	2.76	5.65	5.51	8.25
D5E200P4	3.37	1.34	0.6	5.56	2.88	4.61

A.2 Choix de la méthode de désagrégation

Les Tableaux de l'Annexe A.2 donnent certaines informations supplémentaires sur les sous-problèmes $P(w, d)$ définis par la procédure de décomposition espace-temps pour la résolution du problème désagrégé P restreint à \tilde{Q} . Pour chaque méthode de désagrégation et pour chaque instance, la cellule correspondante apparaît en gris si la méthode de désagrégation réussit à identifier une solution de qualité relative acceptable. Rappelons qu'un calendrier C , solution du problème désagrégé P restreint à \tilde{Q} , est de qualité acceptable si la déviation en pourcentage entre son coût Z_C par rapport au meilleur coût trouvé par les neuf méthodes de désagrégation $\min Z_C$ est inférieure ou égale à 5%.

Les tableaux A.4, A.5 et A.6 donnent le nombre moyen, minimal et maximal de variables de décision des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$. Rappelons que ces sous-problèmes

sont modélisés à l'aide de la formulation basée sur les quarts. Les variables de décision sont donc des variables de sous-couverture, de surcouverture et de quarts internes et externes désagrégés. Pour chaque instance (ligne), une cellule apparaît en gras lorsqu'elle correspond à la méthode de désagrégation (colonne) identifiant le calendrier de meilleur coût ($\min Z_C$) sur l'ensemble de toutes les méthodes de désagrégation.

Rappelons que 36 instances appartenant au JEU1 sont choisies afin de sélectionner la meilleure configuration de la procédure de décomposition espace-temps. Le Tableau A.7 donne pour chaque instance et pour chaque configuration le coût du meilleur calendrier désagrégé trouvé (Z_C). Les cellules en gras correspondent aux coûts minimaux. De même pour le Tableau A.11 qui donne le nombre total d'heures transférées dans les meilleurs calendriers identifiés par les méthodes de désagrégation.

Les Tableaux A.8, A.9 et A.10 donnent les gaps de CPLEX moyens, minimaux et maximaux correspondant aux sous-problèmes $P(w, d)$ restreints aux quarts désagrégés $\tilde{Q}(w, d)$. En effet, la résolution de chaque instance implique la définition d'un problème désagrégé P restreint à \tilde{Q} divisé à l'aide de la procédure de décomposition espace-temps en des sous-problèmes $P(w, d)$. La résolution de ces sous-problèmes se fait à l'aide du BB ou de la GenColBB. Dans les deux cas, un temps limite est imposé lors de l'utilisation de CPLEX égal à 10 minutes. Nous calculons la moyenne, le minimum et le maximum des gaps sur l'ensemble des sous-problèmes pour chaque instance et chaque méthode. Une cellule en gras apparaît lorsqu'elle correspond à la meilleure performance en termes de qualité de la solution.

Le Tableau A.12 donne le temps total requis pour compléter la procédure de décomposition espace-temps pour chaque instance et chaque méthode de désagrégation. Le temps de calcul est constitué par :

- le temps d'énumération de l'ensemble \tilde{Q} des quarts désagrégés considérés par la procédure d'exploration (voir Tableau A.13),
- le temps d'énumération des circuits dans le graphe de transfert et de résolution du modèle EC pour l'élimination de ces circuits (voir Tableau A.14)
- et le temps de résolution total de tous les sous-problèmes $P(w, d)$ en utilisant les procédures BB ou GenColBB (voir Tableau A.15).

Pour l'ensemble de ces quatre tableaux, les cellules en gras correspondent à la meilleure performance en termes de temps de calcul en se limitant aux solutions de qualité relative acceptable (c.-à-d. de déviation $\leq 5\%$).

Tableau A.4 – Nombre moyen de variables de décision des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	5,745	6,679	5,783	27,284	25,569	25,586	6,679	6,679	28,231
D2E20P2	6,201	7,544	6,232	26,122	14,631	14,387	3,836	4,819	18,742
D2E20P3	6,464	7,441	6,475	27,395	15,170	15,110	4,116	5,140	19,394
D2E20P4	6,781	7,975	6,790	25,942	19,815	19,819	6,608	6,608	22,873
D3E20P1	4,436	5,238	4,447	19,675	8,692	8,689	2,518	3,696	14,732
D3E20P2	4,553	5,682	4,803	17,495	9,862	9,968	2,410	3,290	12,908
D3E20P3	5,100	7,238	5,153	20,662	9,437	9,261	2,829	4,188	14,517
D3E20P4	5,188	5,993	5,198	19,588	9,124	9,022	2,898	4,257	14,744
D5E20P1	3,364	4,122	3,228	12,674	6,595	6,635	2,018	2,393	8,144
D5E20P2	3,190	3,606	3,204	11,686	7,248	7,256	2,404	2,620	9,308
D5E20P3	3,611	4,295	3,641	12,331	5,643	5,557	1,618	2,647	8,530
D5E20P4	3,748	4,380	3,761	12,115	5,288	5,386	1,794	2,575	7,975
D5E50P1	6,514	8,566	6,521	31,322	16,418	16,477	4,581	5,305	22,406
D5E50P2	6,746	9,728	6,792	32,571	14,939	14,969	4,906	6,337	23,603
D5E50P3	7,400	10,209	7,290	33,091	14,019	14,066	4,298	6,658	24,923
D5E50P4	7,538	10,114	7,568	31,941	14,209	13,788	4,591	6,557	23,631
D5E70P1	8,233	10,356	8,061	40,657	21,581	21,645	5,442	6,697	29,681
D5E70P2	8,804	12,990	8,845	42,700	19,265	19,190	5,247	7,735	30,671
D5E70P3	9,046	12,243	9,380	41,821	19,654	19,956	5,530	8,355	30,404
D5E70P4	9,448	11,274	9,739	42,136	19,007	19,175	5,414	8,735	30,167
D5E200P1	20,204	23,245	21,622	109,527	65,616	65,421	16,324	18,378	84,285
D5E200P2	22,256	28,599	22,369	112,238	57,464	57,077	16,131	21,032	91,751
D5E200P3	25,068	35,103	25,278	121,180	56,195	56,269	16,622	23,876	94,105
D5E200P4	24,427	29,712	24,550	116,043	53,832	53,552	15,333	22,753	92,959
D10E200P1	11,431	15,360	11,523	62,206	29,072	29,083	8,338	13,723	71,170
D10E200P2	13,415	17,705	13,387	62,418	26,397	26,327	9,192	14,934	67,848
D10E200P3	14,240	19,164	14,347	65,318	27,141	27,176	9,218	16,696	73,336
D10E200P4	14,952	19,813	14,944	66,513	27,356	27,195	9,090	16,971	77,405
D10E300P1	17,269	23,203	17,396	97,353	41,100	41,533	11,284	20,623	114,219
D10E300P2	19,815	29,831	19,926	103,420	37,959	38,434	15,191	27,479	131,835
D10E300P3	20,012	31,487	20,891	103,972	41,420	41,332	15,297	29,861	131,149
D10E300P4	20,559	29,015	20,761	99,618	42,315	42,344	14,618	26,955	126,402
D10E400P1	21,716	30,194	22,080	129,122	55,799	55,824	15,318	26,854	140,626
D10E400P2	23,301	30,274	23,451	115,882	47,668	47,695	16,481	28,521	138,361
D10E400P3	26,389	34,807	26,596	123,176	53,060	53,166	19,004	31,943	136,207
D10E400P4	25,336	33,389	25,721	119,404	52,941	53,267	18,757	30,788	150,013

A.3 Comparaison entre les heuristiques de prétransfert et de post-transfert

Les Tableaux de l'Annexe A.3 donnent certaines informations supplémentaires sur les problèmes définis lors des procédures d'optimisation distribuée de prétransfert et de post-transfert.

Les Tableaux A.16 et A.17 donnent pour chaque instance du JEU1 (chaque ligne) la taille et les résultats des problèmes agrégés de prétransfert et de post-transfert. Pour le problème

Tableau A.5 – Nombre minimal de variables de décision des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	4,513	6,381	4,589	22,233	18,799	18,852	6,381	6,381	24,127
D2E20P2	3,300	3,302	3,291	10,036	2,058	2,058	483	1,489	4,748
D2E20P3	5,208	7,158	5,253	23,359	3,574	3,624	1,250	1,626	4,740
D2E20P4	3,250	3,251	3,243	10,473	6,193	6,183	2,009	2,009	6,184
D3E20P1	2,223	2,220	2,221	5,489	669	669	363	907	1,970
D3E20P2	2,007	2,003	2,003	5,041	1,119	1,119	363	907	2,419
D3E20P3	2,151	2,146	2,146	4,135	773	727	391	831	1,965
D3E20P4	2,227	2,223	2,223	5,356	1,119	1,119	363	907	2,419
D5E20P1	1,750	1,750	1,750	3,502	656	656	278	822	1,956
D5E20P2	1,750	1,750	1,750	3,502	1,295	1,295	550	822	1,956
D5E20P3	1,606	2,704	2,309	2,620	247	240	205	677	1,061
D5E20P4	2,300	2,464	2,299	6,230	215	206	206	979	1,831
D5E50P1	2,695	2,695	2,695	9,524	2,565	2,565	1,187	1,405	5,186
D5E50P2	3,319	3,715	3,337	13,224	2,774	2,778	1,149	1,627	6,163
D5E50P3	5,090	3,035	3,455	21,923	2,072	1,610	377	1,987	6,133
D5E50P4	3,745	3,961	3,730	11,850	1,600	1,138	389	2,091	6,734
D5E70P1	1,684	1,684	1,684	3,502	656	656	278	822	1,956
D5E70P2	2,866	3,618	2,928	9,312	1,232	1,232	350	974	2,454
D5E70P3	2,938	3,690	3,000	9,312	1,184	1,154	350	974	2,484
D5E70P4	2,866	3,618	2,888	9,174	1,154	1,154	350	974	2,484
D5E200P1	6,158	6,665	6,148	27,780	11,803	11,362	2,189	2,893	13,482
D5E200P2	6,497	6,350	6,341	23,307	4,863	5,292	1,138	3,161	12,762
D5E200P3	6,640	6,541	6,464	23,123	4,404	3,925	851	3,243	11,222
D5E200P4	6,942	6,782	6,829	25,243	6,248	5,369	1,142	3,246	12,132
D10E200P1	4,822	4,822	4,817	20,521	2,547	2,547	770	2,490	11,209
D10E200P2	5,579	5,423	5,406	22,982	2,064	2,065	458	2,760	11,256
D10E200P3	6,226	6,011	6,087	21,313	1,592	1,592	602	2,980	11,269
D10E200P4	6,438	6,083	6,069	21,229	2,064	2,065	986	3,226	12,042
D10E300P1	7,064	7,830	7,138	34,913	2,508	2,562	666	3,172	14,896
D10E300P2	6,720	8,243	8,146	26,062	2,063	1,612	474	5,023	16,469
D10E300P3	8,562	11,575	8,591	37,603	2,054	2,508	618	6,204	21,469
D10E300P4	7,096	9,513	9,546	28,060	2,508	2,508	762	5,724	19,294
D10E400P1	4,086	4,070	4,069	17,789	3,434	3,434	801	2,305	10,241
D10E400P2	4,433	4,356	4,346	16,633	3,906	3,907	1,402	2,820	12,271
D10E400P3	5,590	7,230	5,620	24,404	2,997	3,897	726	2,918	16,627
D10E400P4	6,499	7,363	6,592	23,557	3,447	3,897	1,017	3,320	12,208

agrégé de prétransfert, le Tableau A.16 donne pour chaque instance le nombre de variables de décision du problème en nombre entiers $\bar{P}_{Pré}$ au niveau du nœud racine, le nombre de contraintes et le nombre de coefficients non nuls. Pour le problème agrégé de post-transfert, ce tableau donne pour chaque instance le nombre moyen de variables de décision des problèmes $\bar{P}_d, d \in D$ formant le problème agrégé de post-transfert \bar{P}_{Post} , le nombre moyen de contraintes et le nombre moyen de coefficients non nuls.

Pour le problème agrégé de prétransfert, le Tableau A.17 donne pour chaque instance le

Tableau A.6 – Nombre maximal de variables de décision des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	6,976	6,976	6,976	32,334	32,338	32,320	6,976	6,976	32,334
D2E20P2	9,102	11,786	9,172	42,207	31,656	31,644	6,736	7,136	32,036
D2E20P3	7,720	7,724	7,696	31,431	31,663	31,658	6,844	7,221	31,739
D2E20P4	10,312	12,698	10,336	41,410	31,605	31,594	8,250	8,250	31,605
D3E20P1	6,009	8,418	6,042	32,396	27,419	27,406	7,681	7,681	33,128
D3E20P2	6,493	9,428	6,811	26,487	26,595	26,647	6,821	7,221	32,025
D3E20P3	7,676	14,090	7,850	38,703	26,783	26,843	11,491	11,324	36,979
D3E20P4	6,993	9,416	7,043	32,276	27,431	27,436	9,476	9,476	33,415
D5E20P1	4,655	6,667	5,163	24,944	18,139	17,686	3,635	3,635	17,685
D5E20P2	5,152	6,112	5,188	21,609	20,457	20,484	5,353	5,353	21,243
D5E20P3	5,379	6,469	5,361	20,992	19,553	19,083	5,065	5,366	19,883
D5E20P4	5,740	6,891	5,780	20,747	20,232	20,244	5,961	5,961	21,228
D5E50P1	9,806	16,031	10,034	55,729	33,140	33,140	9,910	13,031	59,395
D5E50P2	9,923	14,954	10,070	49,245	33,460	33,006	12,283	10,366	49,370
D5E50P3	8,723	15,170	10,421	40,162	35,306	35,307	8,761	9,574	39,807
D5E50P4	10,113	14,132	10,236	46,228	35,306	35,308	9,583	15,053	52,209
D5E70P1	11,158	15,375	13,180	62,303	54,752	54,743	13,013	14,729	62,709
D5E70P2	12,540	21,706	11,836	67,374	54,752	54,780	14,403	13,379	56,041
D5E70P3	11,740	21,056	12,874	65,454	54,752	56,085	15,775	15,413	60,593
D5E70P4	13,014	17,040	14,183	66,213	61,263	63,506	20,097	16,089	59,763
D5E200P1	34,903	42,147	34,783	185,614	149,024	149,420	48,782	37,591	204,383
D5E200P2	38,666	48,400	39,237	193,771	147,357	147,317	36,682	43,458	169,390
D5E200P3	32,767	53,378	33,604	184,611	143,979	143,426	43,534	53,714	196,872
D5E200P4	33,269	50,189	35,360	163,084	145,143	144,163	39,301	49,660	204,372
D10E200P1	18,505	27,462	18,538	101,359	96,243	96,217	36,811	44,015	223,259
D10E200P2	23,984	28,366	23,745	114,425	96,216	96,166	57,482	54,310	258,765
D10E200P3	21,975	34,656	21,838	99,395	99,300	99,300	56,704	71,992	344,291
D10E200P4	25,802	44,933	27,038	135,421	100,322	100,902	83,799	90,006	391,716
D10E300P1	26,716	45,559	26,808	163,841	138,322	137,864	40,361	83,511	471,675
D10E300P2	34,090	82,319	39,917	229,655	145,065	145,215	93,884	141,481	484,544
D10E300P3	34,526	90,863	41,345	238,573	151,214	151,306	108,505	185,943	775,957
D10E300P4	36,442	76,813	37,705	219,054	149,170	149,395	101,866	161,478	695,541
D10E400P1	41,028	50,624	41,494	238,767	200,704	200,820	61,352	101,798	619,146
D10E400P2	47,906	51,277	47,652	205,289	207,025	206,699	78,706	78,706	395,526
D10E400P3	50,500	56,524	50,402	211,423	203,170	203,057	71,018	92,675	368,138
D10E400P4	56,090	64,264	54,451	250,449	201,470	201,456	72,477	96,833	530,596

temps de calcul total requis pour compléter l'étape d'agrégation, la valeur de la meilleure solution trouvée et le gap de CPLEX obtenu lors de la résolution du problème $\bar{P}_{Pré}$ au niveau du nœud racine. Pour le problème agrégé de post-transfert, ce tableau donne pour chaque instance le temps de calcul total requis pour compléter l'étape d'agrégation, la valeur de la meilleure solution trouvée et le gap maximal de CPLEX obtenu lors de la résolution des problèmes $\bar{P}_d, d \in D$ formant le problème agrégé de post-transfert \bar{P}_{Post} .

Tableau A.7 – Coûts des calendriers calculés par les différentes méthodes de désagrégation

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	304.1	288.9	290.0	287.8	301.9	286.6	288.9	288.9	287.8
D2E20P2	735.4	733.0	731.8	717.7	848.2	779.0	628.4	651.4	651.7
D2E20P3	781.7	751.0	758.0	634.4	589.8	596.1	583.7	646.9	586.2
D2E20P4	589.6	599.0	601.3	561.8	431.2	403.4	450.0	450.0	402.2
D3E20P1	386.5	391.1	388.9	383.1	390.2	358.6	359.6	318.0	314.7
D3E20P2	857.8	915.9	915.9	847.8	1,025.5	970.6	830.4	800.3	820.4
D3E20P3	905.4	933.2	937.9	914.0	841.2	871.0	741.0	821.7	756.8
D3E20P4	889.8	892.1	896.8	810.9	833.6	783.7	885.0	876.0	728.8
D5E20P1	541.9	495.1	541.9	487.1	491.0	467.7	478.0	459.3	459.3
D5E20P2	694.1	702.1	710.4	823.9	631.1	626.7	610.1	610.1	619.3
D5E20P3	831.0	795.9	798.1	803.3	822.7	808.0	743.2	787.4	741.0
D5E20P4	892.8	888.0	894.9	846.0	875.0	1,032.6	916.7	894.7	833.0
D5E50P1	907.2	910.9	913.1	839.0	1,069.9	969.4	743.0	1,023.6	915.6
D5E50P2	1,321.1	1,239.8	1,339.1	1,293.4	1,507.2	1,410.7	1,310.8	1,309.9	1,584.8
D5E50P3	1,379.7	1,352.6	1,303.8	1,216.6	1,609.8	1,690.7	1,473.2	1,380.7	1,479.4
D5E50P4	1,236.2	1,171.1	1,176.0	1,180.6	1,565.8	1,381.7	1,430.0	1,288.2	1,264.1
D5E70P1	1,443.1	1,361.0	1,511.0	1,364.1	1,228.1	1,272.7	1,078.4	1,278.2	1,435.8
D5E70P2	2,790.9	2,054.0	2,084.4	1,948.2	2,163.9	2,577.2	2,029.2	2,024.0	2,089.6
D5E70P3	2,219.9	2,396.7	2,029.3	1,798.8	2,192.8	2,190.6	2,016.9	1,954.4	2,304.0
D5E70P4	2,404.0	2,400.7	2,304.0	1,978.0	2,082.3	2,118.6	1,801.1	1,750.9	1,668.1
D5E200P1	3,771.0	4,030.4	3,071.7	5,299.6	2,932.4	2,678.9	2,122.8	3,746.1	3,775.9
D5E200P2	4,304.6	4,279.7	4,382.6	7,664.0	4,107.7	4,157.0	3,367.1	3,647.0	3,559.3
D5E200P3	4,387.0	4,777.6	4,520.6	6,375.3	5,059.4	4,829.9	4,445.4	5,721.0	5,533.1
D5E200P4	4,338.6	4,774.0	4,079.0	16,496.1	4,272.2	4,736.8	4,451.9	4,908.9	5,065.6
D10E200P1	2,362.4	2,102.0	2,118.4	2,253.0	2,994.2	3,039.9	2,406.7	2,073.8	2,253.0
D10E200P2	4,220.7	3,820.0	4,042.6	16,119.5	4,979.0	4,647.6	4,215.6	3,982.2	4,050.0
D10E200P3	4,510.7	4,057.3	4,407.7	5,782.0	5,413.0	5,703.7	5,026.8	4,390.4	4,789.6
D10E200P4	4,707.7	4,420.8	4,422.1	6,884.0	5,596.7	5,461.7	4,794.1	4,475.1	4,776.0
D10E300P1	4,526.1	4,263.8	4,286.0	5,351.6	5,712.0	5,507.4	4,394.8	4,275.0	4,140.7
D10E300P2	5,482.4	4,835.1	4,492.3	6,841.3	7,089.0	6,812.3	5,061.2	5,566.0	6,580.7
D10E300P3	6,985.7	5,751.0	6,028.0	7,113.4	8,215.9	7,333.6	6,259.8	5,970.7	10,058.5
D10E300P4	6,788.7	6,064.0	6,077.8	18,506.0	9,021.9	7,764.6	6,605.2	6,256.0	6,421.8
D10E400P1	4,296.8	3,902.8	4,066.3	65,731.3	6,921.0	6,522.0	4,773.1	4,101.2	4,994.4
D10E400P2	4,675.6	4,551.0	4,719.0	47,969.2	6,822.6	5,922.0	5,315.3	5,211.3	5,571.4
D10E400P3	7,475.4	7,207.4	7,275.8	76,710.6	8,352.0	7,580.0	7,822.2	7,779.6	7,525.4
D10E400P4	7,316.1	6,688.9	6,718.0	36,552.0	8,165.3	7,867.1	7,072.6	8,159.0	8,690.8

Tableau A.8 – Gaps moyens de CPLEX des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$ en pourcentage

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	0.0	0.0	0.3	0.5	0.0	0.0	0.0	0.0	0.5
D2E20P2	0.7	0.1	0.8	7.5	0.0	0.1	0.0	0.2	0.2
D2E20P3	0.4	0.2	0.2	5.5	0.0	0.0	0.0	0.0	0.0
D2E20P4	0.3	0.0	0.4	4.2	3.2	2.1	0.1	0.1	3.1
D3E20P1	0.0	0.0	0.1	0.3	0.0	0.0	0.0	0.1	0.2
D3E20P2	0.3	0.3	0.3	0.7	0.0	0.2	0.0	0.1	0.2
D3E20P3	0.1	0.1	0.6	0.5	0.1	0.2	0.0	0.2	0.3
D3E20P4	0.0	0.0	0.3	0.2	0.2	0.2	0.0	0.0	0.3
D5E20P1	0.0	0.0	0.0	0.2	0.1	0.1	0.1	0.1	0.1
D5E20P2	0.2	0.0	0.3	0.4	0.0	0.0	0.1	0.1	0.4
D5E20P3	0.0	0.1	0.2	0.3	0.0	0.0	0.0	0.0	0.2
D5E20P4	0.2	0.0	0.0	0.7	0.1	0.4	0.0	0.0	0.6
D5E50P1	0.0	0.2	0.1	0.9	0.0	0.0	0.0	0.0	0.0
D5E50P2	0.0	0.0	0.2	0.2	0.0	0.1	0.0	0.2	0.2
D5E50P3	0.2	0.3	0.3	2.0	0.1	0.0	0.0	0.1	0.9
D5E50P4	0.1	0.3	0.1	0.5	0.0	0.0	0.0	0.1	0.1
D5E70P1	0.2	0.2	0.2	0.5	0.0	0.0	0.0	0.0	0.4
D5E70P2	0.0	0.2	0.2	0.5	0.1	0.1	0.1	0.1	0.2
D5E70P3	0.1	0.1	0.6	3.3	0.2	0.1	0.0	0.2	1.0
D5E70P4	0.2	0.0	0.1	2.4	0.2	0.1	0.0	0.1	1.2
D5E200P1	0.1	0.5	0.3	19.3	0.0	0.2	0.1	0.1	3.7
D5E200P2	0.4	0.6	0.6	26.9	0.1	0.1	0.2	0.2	2.4
D5E200P3	0.7	0.8	1.1	18.5	0.2	0.2	0.0	0.2	1.8
D5E200P4	0.3	0.3	0.5	35.0	0.1	0.1	0.0	0.1	3.1
D10E200P1	0.1	0.4	0.1	1.0	0.4	0.4	0.0	0.1	0.7
D10E200P2	0.0	0.3	0.2	20.6	0.0	0.1	0.0	0.1	0.9
D10E200P3	0.4	0.4	0.3	13.4	0.0	0.0	0.0	0.2	1.6
D10E200P4	0.2	0.5	0.4	14.4	0.1	0.0	0.0	0.1	0.4
D10E300P1	0.2	0.4	0.4	11.4	0.4	0.3	0.0	0.1	0.2
D10E300P2	0.4	0.3	0.4	18.9	0.1	0.1	0.1	0.2	2.4
D10E300P3	0.4	0.5	0.4	14.8	0.1	0.1	0.0	0.2	4.2
D10E300P4	0.3	0.2	0.4	15.0	0.1	0.1	0.1	0.1	2.1
D10E400P1	0.4	0.4	0.6	26.5	0.1	0.1	0.1	0.2	1.7
D10E400P2	0.2	0.3	0.7	39.7	0.1	0.1	0.1	0.1	4.9
D10E400P3	0.4	0.4	0.7	41.3	0.1	0.1	0.0	0.2	2.7
D10E400P4	0.3	0.5	0.6	34.7	0.1	0.1	0.0	0.3	2.1

Tableau A.9 – Gaps minimaux de CPLEX des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$ en pourcentage

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D2E20P2	0.5	0.0	0.5	0.4	0.0	0.0	0.0	0.0	0.0
D2E20P3	0.0	0.0	0.0	0.6	0.0	0.0	0.0	0.0	0.0
D2E20P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D3E20P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D3E20P2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D3E20P3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D3E20P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E20P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E20P2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E20P3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E20P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E50P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E50P2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E50P3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E50P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E70P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E70P2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E70P3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E70P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E200P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D5E200P2	0.0	0.0	0.3	0.7	0.0	0.0	0.0	0.0	0.0
D5E200P3	0.0	0.0	0.0	0.8	0.0	0.0	0.0	0.0	0.0
D5E200P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E200P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E200P2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E200P3	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0
D10E200P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E300P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E300P2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E300P3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E300P4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E400P1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
D10E400P2	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0
D10E400P3	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0
D10E400P4	0.0	0.0	0.0	0.3	0.0	0.0	0.0	0.0	0.0

Tableau A.10 – Gaps maximaux de CPLEX des sous-problèmes $P(w, d)$ restreints à $\tilde{Q}(w, d)$ en pourcentage

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	0.0	0.0	0.7	0.9	0.0	0.0	0.0	0.0	0.9
D2E20P2	0.9	0.2	1.0	14.7	0.0	1.0	0.1	1.0	1.0
D2E20P3	0.8	0.4	0.4	10.4	0.0	0.0	0.0	0.1	0.0
D2E20P4	0.7	0.0	0.7	8.4	12.7	8.3	0.4	0.4	12.5
D3E20P1	0.0	0.0	0.3	0.8	0.0	0.0	0.0	0.8	1.0
D3E20P2	1.0	0.8	0.8	1.1	0.8	1.0	0.8	0.7	0.9
D3E20P3	0.4	0.4	0.9	1.0	1.0	2.7	0.1	1.0	2.3
D3E20P4	0.0	0.0	0.8	0.7	0.9	0.8	0.0	0.0	0.8
D5E20P1	0.0	0.0	0.0	0.9	1.0	1.0	1.0	1.0	1.0
D5E20P2	0.9	0.0	0.9	1.0	0.0	0.0	0.7	0.6	4.4
D5E20P3	0.0	0.3	0.8	0.9	0.9	0.3	0.0	0.1	1.0
D5E20P4	0.9	0.0	0.1	2.6	1.1	7.1	0.0	0.0	6.8
D5E50P1	0.0	0.8	0.5	4.5	0.3	0.0	0.0	0.0	0.2
D5E50P2	0.0	0.1	1.0	0.9	0.2	1.0	0.0	1.0	1.0
D5E50P3	0.6	0.8	0.9	5.8	1.0	1.0	0.7	1.0	11.3
D5E50P4	0.4	0.7	0.7	1.0	0.1	0.8	0.0	0.9	1.0
D5E70P1	1.0	0.8	0.7	1.5	0.9	0.7	1.0	0.3	3.2
D5E70P2	0.0	0.6	0.8	2.3	1.0	0.9	0.9	0.9	0.9
D5E70P3	0.4	0.3	0.9	11.7	4.7	1.0	0.8	0.8	5.0
D5E70P4	0.7	0.2	0.6	10.8	6.0	1.0	0.8	1.0	11.9
D5E200P1	0.7	0.9	0.6	65.9	0.7	2.4	0.9	0.7	35.9
D5E200P2	1.0	1.1	0.9	72.3	1.1	1.4	1.0	1.0	15.5
D5E200P3	1.2	1.5	1.9	43.0	5.7	3.7	0.4	0.8	7.8
D5E200P4	0.8	0.7	1.0	90.1	0.9	1.0	0.6	0.7	16.5
D10E200P1	0.9	1.0	0.7	4.8	34.1	28.4	0.8	1.0	17.6
D10E200P2	0.0	0.9	0.9	97.3	0.9	1.0	0.9	0.9	13.9
D10E200P3	1.0	1.0	1.0	49.8	1.0	1.0	0.9	0.9	29.9
D10E200P4	0.9	1.0	1.0	69.7	1.0	1.0	1.0	1.0	2.3
D10E300P1	0.9	1.0	0.8	45.5	26.2	11.1	1.0	0.9	1.0
D10E300P2	1.0	0.9	0.9	52.2	1.0	1.0	0.9	1.0	46.1
D10E300P3	1.0	1.0	0.9	43.8	2.1	3.5	0.9	0.9	81.2
D10E300P4	0.6	0.9	0.9	92.2	0.9	1.0	1.0	0.9	24.5
D10E400P1	0.9	1.0	1.0	97.9	1.0	1.0	1.0	0.9	37.6
D10E400P2	0.9	0.7	3.2	97.0	1.0	1.0	0.9	1.0	69.1
D10E400P3	2.4	1.0	1.6	96.6	0.9	1.0	0.7	0.9	48.5
D10E400P4	0.6	0.9	1.8	92.9	1.0	3.8	0.8	1.0	31.9

Tableau A.11 – Nombre total d’heures transférées des calendriers calculés par les différentes méthodes de désagrégation

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	14.0	15.0	14.8	15.0	14.0	15.0	15.0	15.0	14.8
D2E20P2	22.0	21.3	21.5	20.0	34.0	33.8	40.0	34.8	33.0
D2E20P3	24.0	24.0	24.0	24.0	36.0	33.8	48.0	34.8	31.8
D2E20P4	15.0	15.5	15.0	16.0	27.0	30.0	30.0	30.0	29.3
D3E20P1	15.0	16.0	16.0	16.0	25.0	26.8	26.8	26.8	26.8
D3E20P2	15.0	22.8	22.8	18.0	20.0	22.0	33.0	27.0	27.0
D3E20P3	29.0	35.3	35.3	38.3	31.0	35.0	54.5	41.5	33.8
D3E20P4	11.0	13.0	13.0	15.3	14.0	25.0	17.0	17.0	25.0
D5E20P1	18.0	23.8	18.8	8.8	15.0	22.0	36.8	36.8	36.8
D5E20P2	20.0	19.0	18.0	14.0	29.0	30.8	31.3	31.3	25.3
D5E20P3	26.0	25.8	25.0	28.3	23.0	35.0	40.5	34.5	32.5
D5E20P4	22.0	20.3	20.3	21.3	20.0	21.8	22.8	21.0	25.3
D5E50P1	76.0	89.0	89.0	84.5	40.0	69.5	107.3	82.3	69.5
D5E50P2	116.0	128.0	123.8	131.0	107.0	119.3	137.8	134.3	123.5
D5E50P3	124.0	141.5	148.3	127.0	93.0	85.0	144.8	136.5	112.0
D5E50P4	122.0	139.0	139.8	143.0	81.0	124.5	140.3	141.0	129.8
D5E70P1	75.0	100.3	76.0	78.8	95.0	140.0	157.0	105.3	81.3
D5E70P2	44.0	159.3	142.8	156.3	131.0	127.8	167.3	155.5	143.8
D5E70P3	123.0	126.0	169.3	173.0	154.0	185.3	207.5	189.8	146.3
D5E70P4	46.0	63.3	94.3	100.8	111.0	154.5	179.3	152.8	133.8
D5E200P1	203.0	159.0	259.8	216.3	284.0	356.0	426.3	235.8	244.0
D5E200P2	260.0	290.0	275.0	253.3	321.0	386.5	461.8	383.8	401.5
D5E200P3	305.0	332.0	367.0	309.3	313.0	430.0	477.5	281.5	306.0
D5E200P4	260.0	291.8	338.8	300.5	270.0	319.3	372.5	267.8	233.0
D10E200P1	348.0	397.5	392.3	397.0	302.0	352.5	451.3	434.5	417.5
D10E200P2	294.0	397.5	349.5	331.8	310.0	386.0	436.5	423.3	414.3
D10E200P3	352.0	440.0	401.8	416.3	308.0	339.8	453.5	437.3	408.0
D10E200P4	329.0	415.5	417.0	412.8	250.0	347.0	464.3	432.3	402.5
D10E300P1	474.0	547.5	544.8	490.0	359.0	465.5	636.3	571.3	528.0
D10E300P2	707.0	829.3	848.8	778.8	535.0	686.8	897.0	793.8	753.5
D10E300P3	601.0	807.3	800.3	785.5	499.0	714.8	892.0	818.0	571.3
D10E300P4	546.0	684.0	678.3	647.5	420.0	631.8	760.0	705.8	712.5
D10E400P1	942.0	1,031.8	1,017.3	839.0	694.0	774.8	1,052.5	1,020.5	998.5
D10E400P2	637.0	713.8	703.5	655.3	469.0	587.3	723.0	680.3	705.5
D10E400P3	708.0	802.8	797.0	717.8	508.0	676.0	845.8	782.5	736.3
D10E400P4	608.0	719.8	719.5	654.5	495.0	625.3	806.3	623.5	635.5

Tableau A.12 – Temps de calcul des méthodes de désagrégation en secondes

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	3.27	3.61	6.20	114.25	107.01	59.10	3.73	3.67	90.52
D2E20P2	8.21	40.81	38.23	606.89	214.58	110.21	6.96	8.68	435.98
D2E20P3	7.80	5.44	8.87	645.83	96.03	73.85	8.19	6.46	97.27
D2E20P4	3.93	5.07	7.26	606.27	615.89	624.31	6.80	6.79	622.49
D3E20P1	2.74	3.23	5.20	82.35	21.92	27.03	3.31	4.18	37.52
D3E20P2	3.43	3.76	5.94	620.75	69.62	107.87	6.52	6.67	258.82
D3E20P3	3.35	4.47	4.71	121.38	389.07	669.89	8.94	7.05	697.30
D3E20P4	3.86	4.84	5.14	180.72	74.04	87.84	4.86	5.16	455.40
D5E20P1	2.98	3.64	4.27	94.82	230.02	138.71	5.94	7.24	172.76
D5E20P2	2.64	3.26	5.25	154.22	16.76	22.77	3.65	4.02	632.89
D5E20P3	3.31	6.40	6.48	114.54	29.38	35.84	7.58	6.39	212.25
D5E20P4	4.54	5.50	6.15	649.49	613.44	620.17	5.19	6.86	669.86
D5E50P1	12.83	15.22	20.16	725.85	92.70	128.03	19.56	14.05	154.96
D5E50P2	8.00	13.52	14.56	187.28	80.26	124.25	16.94	18.96	172.98
D5E50P3	14.61	21.56	20.30	1,684.80	117.79	302.34	22.42	30.06	1,002.03
D5E50P4	9.56	14.52	17.97	231.57	108.54	127.74	21.88	19.42	196.75
D5E70P1	14.05	19.37	19.84	784.22	198.42	234.90	27.83	20.07	1,316.11
D5E70P2	10.48	23.40	45.37	695.01	213.06	265.32	26.41	24.11	492.63
D5E70P3	14.87	21.45	53.60	1,719.45	867.61	316.79	35.35	63.46	2,597.43
D5E70P4	11.72	14.60	26.24	1,807.56	769.78	265.65	35.16	64.08	1,517.51
D5E200P1	72.97	101.76	302.71	1,909.87	1,273.73	2,096.76	84.00	66.21	3,000.10
D5E200P2	619.05	730.56	632.34	2,813.17	1,366.61	1,535.49	119.04	331.29	3,574.82
D5E200P3	1,208.57	890.55	1,263.69	2,492.91	1,315.58	1,689.91	114.98	101.16	4,257.98
D5E200P4	328.93	573.86	739.22	1,945.80	732.23	1,328.49	101.94	117.83	3,696.73
D10E200P1	64.83	85.57	269.18	2,588.31	1,144.06	1,274.89	87.24	85.23	1,629.88
D10E200P2	120.36	303.67	486.21	3,611.11	554.58	1,316.43	112.98	116.13	4,775.86
D10E200P3	169.59	274.55	324.11	4,290.66	586.64	906.38	104.96	111.65	3,836.63
D10E200P4	61.30	260.89	207.09	3,102.51	633.10	915.47	113.06	154.80	3,360.39
D10E300P1	143.36	127.79	156.53	3,288.49	1,391.90	2,318.21	123.32	136.26	2,604.18
D10E300P2	131.87	592.27	381.43	4,536.44	823.04	1,389.68	176.89	224.42	5,140.00
D10E300P3	243.29	401.82	537.01	4,672.21	1,674.50	2,011.68	216.44	257.94	6,937.84
D10E300P4	405.94	495.76	749.87	4,628.76	1,071.43	1,914.76	174.53	188.59	6,710.47
D10E400P1	154.74	217.51	358.12	3,902.59	1,483.18	1,936.31	210.62	206.88	5,758.66
D10E400P2	536.73	724.88	1,191.10	4,437.85	1,771.02	2,490.95	178.00	314.40	6,774.61
D10E400P3	1,201.72	575.79	1,429.62	4,339.27	1,688.50	2,525.33	263.61	612.86	7,471.65
D10E400P4	507.18	421.64	1,165.74	4,550.87	1,841.92	2,676.53	251.48	355.54	7,522.32

Tableau A.13 – Temps d'énumération de l'ensemble \tilde{Q} par les méthodes de désagrégation en secondes

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	0.06	0.06	0.06	0.17	0.12	0.16	0.09	0.07	0.21
D2E20P2	0.05	0.08	0.08	0.26	0.10	0.23	0.13	0.12	0.41
D2E20P3	0.05	0.08	0.09	0.26	0.11	0.26	0.14	0.13	0.51
D2E20P4	0.05	0.08	0.08	0.19	0.10	0.19	0.10	0.10	0.31
D3E20P1	0.06	0.07	0.06	0.20	0.10	0.19	0.08	0.07	0.25
D3E20P2	0.05	0.07	0.07	0.22	0.10	0.22	0.09	0.10	0.32
D3E20P3	0.05	0.11	0.10	0.32	0.10	0.33	0.18	0.20	0.65
D3E20P4	0.05	0.07	0.07	0.19	0.10	0.19	0.12	0.10	0.28
D5E20P1	0.05	0.08	0.08	0.24	0.13	0.24	0.09	0.09	0.31
D5E20P2	0.05	0.06	0.07	0.20	0.10	0.19	0.08	0.09	0.24
D5E20P3	0.05	0.09	0.09	0.30	0.10	0.28	0.11	0.12	0.38
D5E20P4	0.05	0.08	0.09	0.21	0.10	0.21	0.08	0.09	0.25
D5E50P1	0.12	0.20	0.19	0.58	0.25	0.60	0.37	0.39	1.47
D5E50P2	0.13	0.23	0.24	0.77	0.24	0.76	0.48	0.57	1.90
D5E50P3	0.13	0.25	0.26	0.85	0.25	0.83	0.59	0.57	2.31
D5E50P4	0.12	0.23	0.24	0.73	0.23	0.77	0.50	0.51	1.88
D5E70P1	0.17	0.30	0.28	0.91	0.34	1.00	0.56	0.56	2.13
D5E70P2	0.17	0.33	0.33	1.08	0.34	1.07	0.70	0.72	2.61
D5E70P3	0.18	0.37	0.37	1.29	0.36	1.25	0.95	0.93	3.64
D5E70P4	0.18	0.37	0.38	1.20	0.33	1.19	0.91	0.89	3.25
D5E200P1	0.47	0.78	0.76	2.30	0.94	2.29	2.20	2.08	8.85
D5E200P2	0.50	0.90	0.91	2.63	0.98	2.63	2.25	2.24	8.96
D5E200P3	0.51	1.11	1.08	3.21	1.21	3.24	3.72	3.67	14.47
D5E200P4	0.48	0.98	0.88	2.46	1.07	2.55	2.71	2.70	10.50
D10E200P1	0.47	0.78	0.78	2.20	0.98	2.37	2.76	2.76	12.47
D10E200P2	0.49	0.88	0.83	2.55	0.95	2.55	3.32	20.07	12.89
D10E200P3	0.48	0.96	0.99	2.76	0.98	2.72	4.19	4.13	16.62
D10E200P4	0.48	0.94	0.96	2.73	1.00	2.74	4.34	4.28	16.71
D10E300P1	0.73	1.20	1.20	3.43	1.44	3.35	4.61	4.61	20.98
D10E300P2	0.76	1.62	1.60	5.09	1.54	4.84	7.16	7.16	30.22
D10E300P3	0.76	1.69	1.70	5.04	1.47	4.99	8.50	8.42	34.56
D10E300P4	0.78	1.69	1.50	4.41	1.44	4.37	7.28	6.96	29.21
D10E400P1	1.00	1.71	1.69	4.78	1.87	4.75	6.26	6.17	28.80
D10E400P2	1.01	1.56	1.66	4.32	1.92	4.28	6.95	6.91	28.85
D10E400P3	1.04	1.98	1.88	5.16	1.97	5.09	9.22	9.16	35.74
D10E400P4	1.00	1.75	1.78	4.95	1.86	4.87	9.26	77.52	37.45

Tableau A.14 – Temps de la procédure d'élimination des transferts cycliques par les méthodes de désagrégation en secondes

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	0.19	0.21	0.21	0.77	0.57	0.50	0.23	0.18	0.73
D2E20P2	0.26	0.32	0.26	0.88	1.72	1.41	0.82	0.63	1.69
D2E20P3	0.28	0.28	0.24	0.90	1.72	1.42	0.97	0.65	1.45
D2E20P4	0.29	0.33	0.23	0.93	1.01	0.81	0.34	0.30	1.12
D3E20P1	0.28	0.28	0.24	0.84	0.91	0.72	0.42	0.38	0.99
D3E20P2	0.29	0.32	0.25	1.28	1.69	1.42	1.06	0.79	1.61
D3E20P3	0.29	0.35	0.28	0.88	2.24	3.07	1.17	0.77	1.69
D3E20P4	0.29	0.32	0.24	0.89	0.96	0.92	0.57	0.42	1.07
D5E20P1	0.32	0.37	0.46	1.12	1.12	0.99	0.64	0.72	1.56
D5E20P2	0.32	0.34	0.30	0.89	0.78	0.61	0.38	0.39	1.34
D5E20P3	0.35	0.52	0.33	0.94	1.85	1.41	1.37	0.88	1.59
D5E20P4	0.34	0.40	0.31	1.01	1.56	1.38	0.76	0.96	1.59
D5E50P1	0.71	0.82	0.67	2.40	4.53	2.91	1.89	1.31	3.23
D5E50P2	0.53	0.69	0.51	2.17	3.01	2.38	1.49	1.42	3.64
D5E50P3	0.56	0.74	0.55	3.18	4.62	3.61	2.26	1.40	4.84
D5E50P4	0.59	0.74	0.54	2.11	4.91	3.80	2.26	1.34	3.50
D5E70P1	0.84	0.92	0.71	3.42	6.83	4.92	2.92	1.61	5.23
D5E70P2	0.88	1.03	0.83	3.00	6.70	4.75	2.89	1.97	5.03
D5E70P3	0.89	1.04	0.89	3.16	9.73	6.04	3.79	2.28	6.22
D5E70P4	0.90	1.00	0.83	3.32	8.74	6.54	4.41	2.22	5.77
D5E200P1	1.75	1.95	1.68	7.91	18.54	13.11	5.98	3.51	14.47
D5E200P2	1.85	2.42	1.72	8.03	18.22	13.52	6.33	4.07	14.28
D5E200P3	2.28	2.61	2.01	8.28	22.82	16.25	8.75	4.47	16.27
D5E200P4	2.00	2.38	1.82	8.04	19.50	13.98	6.83	4.32	14.59
D10E200P1	1.92	2.36	1.84	8.75	19.99	15.03	7.23	4.76	20.23
D10E200P2	2.39	2.92	2.43	9.12	18.78	14.39	8.47	5.40	22.81
D10E200P3	5.76	6.23	5.59	13.25	19.36	14.27	9.58	6.97	23.40
D10E200P4	4.55	5.14	4.37	11.43	20.77	15.37	9.84	6.23	22.96
D10E300P1	5.45	6.09	5.21	15.53	26.16	20.40	9.49	6.97	32.70
D10E300P2	14.84	16.53	15.05	25.41	24.05	19.25	13.34	11.14	42.71
D10E300P3	70.95	72.00	70.77	82.58	33.56	25.07	15.48	14.76	47.43
D10E300P4	309.78	310.78	309.71	322.07	28.77	21.34	13.57	11.82	41.07
D10E400P1	3.93	5.01	3.88	17.51	39.33	28.86	14.17	8.92	42.35
D10E400P2	3.67	4.46	3.51	16.21	24.77	19.02	11.14	9.61	41.40
D10E400P3	24.52	25.50	24.31	36.76	37.20	27.34	17.15	14.14	43.39
D10E400P4	14.43	15.47	14.19	27.14	37.94	27.61	16.85	11.81	43.84

Tableau A.15 – Temps total de résolution de tous les sous-problèmes $P(w, d)$ définis par les méthodes de désagrégation en secondes

Instance	(W7,V,F,E,B)	(W7,V,P,E,B)	(W7,V,P,E,G)	(W7,A,P,E,B)	(W1,A,F,E,B)	(W1,A,P,E,G)	(W1,V,P,D,B)	(W3,V,P,D,B)	(W3,A,P,D,B)
D2E20P1	3.02	3.34	5.93	113.30	106.32	58.44	3.41	3.42	89.57
D2E20P2	7.90	40.41	37.89	605.75	212.76	108.58	6.01	7.93	433.88
D2E20P3	7.47	5.07	8.54	644.67	94.20	72.17	7.08	5.67	95.31
D2E20P4	3.59	4.66	6.95	605.15	614.78	623.31	6.37	6.40	621.07
D3E20P1	2.41	2.88	4.90	81.32	20.91	26.12	2.81	3.73	36.28
D3E20P2	3.09	3.37	5.62	619.25	67.83	106.23	5.36	5.78	256.89
D3E20P3	3.01	4.01	4.33	120.18	386.73	666.49	7.59	6.09	694.96
D3E20P4	3.51	4.45	4.83	179.64	72.98	86.73	4.18	4.64	454.05
D5E20P1	2.61	3.19	3.74	93.46	228.77	137.48	5.21	6.43	170.89
D5E20P2	2.27	2.86	4.89	153.13	15.88	21.97	3.19	3.54	631.31
D5E20P3	2.90	5.79	6.06	113.30	27.44	34.15	6.10	5.38	210.28
D5E20P4	4.15	5.02	5.74	648.26	611.78	618.58	4.35	5.82	668.01
D5E50P1	12.00	14.21	19.30	722.87	87.92	124.52	17.30	12.35	150.26
D5E50P2	7.35	12.60	13.81	184.34	77.01	121.11	14.97	16.97	167.44
D5E50P3	13.92	20.57	19.49	1,680.77	112.92	297.89	19.57	28.08	994.88
D5E50P4	8.85	13.55	17.19	228.72	103.39	123.17	19.12	17.57	191.37
D5E70P1	13.04	18.15	18.84	779.89	191.25	228.97	24.35	17.91	1,308.75
D5E70P2	9.43	22.03	44.21	690.93	206.02	259.49	22.82	21.42	484.99
D5E70P3	13.81	20.04	52.34	1,715.00	857.53	309.50	30.61	60.25	2,587.57
D5E70P4	10.64	13.23	25.03	1,803.04	760.71	257.92	29.84	60.96	1,508.49
D5E200P1	70.75	99.03	300.26	1,899.66	1,254.25	2,081.35	75.82	60.62	2,976.78
D5E200P2	616.70	727.25	629.70	2,802.51	1,347.41	1,519.34	110.46	324.99	3,551.58
D5E200P3	1,205.78	886.83	1,260.59	2,481.42	1,291.55	1,670.42	102.51	93.02	4,227.24
D5E200P4	326.45	570.50	736.52	1,935.30	711.66	1,311.97	92.40	110.81	3,671.64
D10E200P1	62.44	82.42	266.55	2,577.35	1,123.09	1,257.49	77.25	77.71	1,597.19
D10E200P2	117.48	299.87	482.95	3,599.45	534.85	1,299.49	101.20	90.67	4,740.16
D10E200P3	163.35	267.35	317.53	4,274.65	566.30	889.38	91.19	100.55	3,796.61
D10E200P4	56.26	254.80	201.76	3,088.35	611.33	897.36	98.87	144.29	3,320.72
D10E300P1	137.17	120.51	150.11	3,269.53	1,364.30	2,294.46	109.23	124.67	2,550.50
D10E300P2	116.28	574.12	364.78	4,505.94	797.45	1,365.59	156.38	206.12	5,067.07
D10E300P3	171.58	328.13	464.54	4,584.59	1,639.47	1,981.63	192.46	234.76	6,855.85
D10E300P4	95.38	183.29	438.65	4,302.28	1,041.22	1,889.05	153.68	169.81	6,640.19
D10E400P1	149.81	210.78	352.55	3,880.30	1,441.98	1,902.70	190.19	191.79	5,687.51
D10E400P2	532.05	718.86	1,185.93	4,417.32	1,744.33	2,467.65	159.90	297.89	6,704.36
D10E400P3	1,176.16	548.31	1,403.43	4,297.36	1,649.33	2,492.90	237.24	589.55	7,392.52
D10E400P4	491.75	404.42	1,149.76	4,518.78	1,802.12	2,644.04	225.37	266.21	7,441.03

Tableau A.16 – Taille des problèmes agrégés de prétransfert $\bar{P}_{Pré}$ et de post-transfert \bar{P}_{Post}

Instance	# Quarts agrégés		# Variables		# Contraintes		# Coeff. non nuls	
	Post	Pré	Post	Pré	Post	Pré	Post	Pré
D2E20P1	2,588	9,198	1,463	3,557	177	373	9,957	28,013
D2E20P2	2,588	9,198	1,463	3,479	177	373	9,957	27,318
D2E20P3	2,588	9,198	1,463	3,447	177	373	9,957	27,091
D2E20P4	2,588	9,198	1,463	3,433	177	373	9,957	26,919
D3E20P1	2,588	11,915	1,032	3,780	146	457	6,694	28,530
D3E20P2	2,588	11,915	1,032	3,742	146	457	6,694	28,328
D3E20P3	2,588	11,915	1,032	3,736	146	457	6,694	28,256
D3E20P4	2,588	11,915	1,032	3,630	146	457	6,694	27,321
D5E20P1	2,588	22,564	687	4,801	121	625	4,084	34,833
D5E20P2	2,588	22,564	687	4,234	121	625	4,084	30,040
D5E20P3	2,588	22,564	687	4,441	121	625	4,084	31,937
D5E20P4	2,588	22,564	687	4,139	121	625	4,084	29,268
D5E50P1	6,470	58,320	1,463	10,973	176	932	9,950	88,051
D5E50P2	6,470	58,320	1,463	10,372	176	932	9,950	82,293
D5E50P3	6,470	58,320	1,463	10,694	176	932	9,950	85,393
D5E50P4	6,470	58,320	1,463	10,285	176	932	9,950	82,033
D5E70P1	9,065	78,831	1,982	14,711	214	1,138	13,889	119,924
D5E70P2	9,065	78,831	1,982	14,050	214	1,138	13,889	114,536
D5E70P3	9,065	78,831	1,982	14,126	214	1,138	13,889	115,178
D5E70P4	9,065	78,831	1,982	13,341	214	1,138	13,889	108,278
D5E200P1	25,894	221,023	5,348	39,939	454	2,472	39,369	336,890
D5E200P2	25,894	221,023	5,348	38,102	454	2,472	39,369	320,550
D5E200P3	25,894	221,023	5,348	38,738	454	2,472	39,369	326,270
D5E200P4	25,894	221,023	5,348	38,433	454	2,472	39,369	324,214
D10E200P1	25,894	701,534	2,758	58,451	269	2,892	19,769	491,396
D10E200P2	25,894	701,534	2,758	51,890	269	2,892	19,769	433,006
D10E200P3	25,894	701,534	2,758	49,828	269	2,892	19,769	415,161
D10E200P4	25,894	701,534	2,758	48,518	269	2,892	19,769	405,447
D10E300P1	38,841	970,203	4,053	84,245	362	3,918	29,569	715,965
D10E300P2	38,841	970,203	4,053	83,907	362	3,918	29,569	709,135
D10E300P3	38,841	970,203	4,053	74,579	362	3,918	29,569	630,070
D10E300P4	38,841	970,203	4,053	73,415	362	3,918	29,569	621,915
D10E400P1	51,788	1,402,111	5,348	104,875	454	4,944	39,369	890,043
D10E400P2	51,788	1,402,111	5,348	104,039	454	4,944	39,369	880,794
D10E400P3	51,788	1,402,111	5,348	92,222	454	4,944	39,369	779,130
D10E400P4	51,788	1,402,111	5,348	84,746	454	4,944	39,369	713,706
D20E600P1	77,682	4,827,016	4,053	177,702	362	7,836	29,569	1,509,955
D20E600P2	77,682	4,827,016	4,053	190,978	362	7,836	29,569	1,612,089
D20E600P3	77,682	4,827,016	4,053	164,305	362	7,836	29,569	1,392,853
D20E600P4	77,682	4,827,016	4,053	163,916	362	7,836	29,569	1,387,146
D20E800P1	103,576	6,268,887	5,348	233,410	454	9,888	39,369	1,995,843
D20E800P2	103,576	6,268,887	5,348	241,470	454	9,888	39,369	2,057,001
D20E800P3	103,576	6,268,887	5,348	205,225	454	9,888	39,369	1,746,816
D20E800P4	103,576	6,268,887	5,348	211,595	454	9,888	39,369	1,807,017
D20E1000P1	129,470	7,917,509	6,643	291,594	547	11,940	49,170	2,483,888
D20E1000P2	129,470	7,917,509	6,643	299,451	547	11,940	49,170	2,558,753
D20E1000P3	129,470	7,917,509	6,643	258,366	547	11,940	49,170	2,201,107
D20E1000P4	129,470	7,917,509	6,643	251,573	547	11,940	49,170	2,147,706

Tableau A.17 – Résultats obtenus par la résolution des problèmes agrégés de prétransfert $\bar{P}_{Pré}$ et de post-transfert \bar{P}_{Post}

Instance	Temps de calcul (sec.)		Valeur de l'objectif		Gap de CPLEX (%)	
	Post	Pré	Post	Pré	Post	Pré
D2E20P1	5.4	6.5	596.6	451.5	0.76	0.84
D2E20P2	10.8	42.3	1,077.3	750.0	0.74	0.81
D2E20P3	44.9	84.8	1,091.1	826.0	0.77	0.00
D2E20P4	25.0	13.6	1,071.9	899.9	0.00	0.04
D3E20P1	1.3	4.0	585.7	411.4	0.87	0.04
D3E20P2	4.0	12.8	1,158.4	923.9	0.13	0.56
D3E20P3	6.0	82.7	1,363.7	957.9	0.90	1.00
D3E20P4	1.0	3.7	1,351.1	1,171.4	0.74	0.50
D5E20P1	0.9	6.8	686.0	630.0	0.99	0.09
D5E20P2	2.3	30.3	936.7	772.4	1.00	0.99
D5E20P3	2.1	133.6	1,099.3	961.8	0.91	0.99
D5E20P4	1.2	7.6	1,400.1	1,366.9	0.91	0.49
D5E50P1	33.7	3,600.1	1,478.2	1,042.8	0.95	1.83
D5E50P2	7.4	594.0	2,780.2	1,636.8	1.00	0.35
D5E50P3	27.9	3,600.1	2,801.6	1,733.3	0.99	19.70
D5E50P4	12.1	1,193.0	3,186.9	2,340.8	0.00	0.46
D5E70P1	88.0	3,600.6	2,087.0	1,412.6	1.00	3.51
D5E70P2	34.5	3,600.2	3,746.2	2,393.1	1.00	5.58
D5E70P3	52.5	3,600.3	3,781.0	2,403.9	0.91	20.31
D5E70P4	95.2	3,600.2	3,819.8	2,986.4	0.97	6.01
D5E200P1	799.7	3,600.3	6,275.5	3,950.3	3.07	5.03
D5E200P2	88.5	3,600.9	8,678.7	5,086.0	0.79	1.68
D5E200P3	1,268.1	3,600.2	9,395.5	6,365.0	1.33	20.79
D5E200P4	169.9	3,600.5	10,174.5	7,717.0	0.65	4.44
D10E200P1	75.1	3,600.3	5,885.7	3,555.8	0.97	11.55
D10E200P2	89.3	3,600.3	8,929.0	5,892.4	0.64	15.47
D10E200P3	519.0	3,600.3	9,872.6	7,067.6	1.00	27.42
D10E200P4	227.7	3,600.3	10,865.5	8,514.5	1.00	14.69
D10E300P1	323.0	3,600.4	9,988.5	6,632.3	0.99	8.89
D10E300P2	79.3	3,600.5	14,799.4	7,677.2	0.95	15.09
D10E300P3	1,296.8	3,600.4	16,088.9	10,002.4	3.10	38.53
D10E300P4	750.6	3,600.4	17,240.6	12,175.1	1.05	17.30
D10E400P1	671.5	3,600.6	14,333.2	6,432.9	1.09	5.25
D10E400P2	214.0	3,602.2	15,212.9	8,951.4	0.85	10.46
D10E400P3	938.3	3,600.4	19,245.7	13,932.6	0.98	53.89
D10E400P4	787.5	3,600.4	20,663.1	14,755.7	1.23	19.18
D20E600P1	322.1	3,600.9	15,473.8	10,906.5	1.00	25.31
D20E600P2	232.6	3,600.6	21,401.7	15,952.3	0.98	46.53
D20E600P3	2,707.8	3,600.6	24,956.3	20,546.9	4.05	56.03
D20E600P4	781.5	3,600.4	27,032.7	24,210.4	1.00	48.47
D20E800P1	955.4	3,600.8	21,925.8	14,802.7	1.44	33.83
D20E800P2	787.0	3,600.7	31,213.4	22,675.0	0.99	52.62
D20E800P3	2,770.7	3,600.9	35,341.9	30,214.2	6.04	59.83
D20E800P4	1,503.6	3,600.9	37,201.1	32,069.6	1.35	52.53
D20E1000P1	1,614.3	3,601.1	22,132.5	16,476.1	2.21	29.52
D20E1000P2	432.7	3,600.7	39,209.5	29,734.7	0.99	57.91
D20E1000P3	2,986.2	3,600.6	42,772.1	37,496.3	6.09	62.47
D20E1000P4	1,805.6	3,600.8	42,520.9	39,260.9	4.52	47.27

Bibliographie

- Addou, I. et F. Soumis. 2007, «Bechtold-jacobs generalized model for shift scheduling with extraordinary overlap», *Annals of Operations Research*, vol. 155, n° 1, p. 177–205.
- Al-Yakoob, S. et H. Sherali. 2008, «A column generation approach for an employee scheduling problem with multiple shifts and work locations», *Journal of the Operational Research Society*, p. 34–43.
- Al-Yakoob, S. M. et H. D. Sherali. 2007, «Mixed-integer programming models for an employee scheduling problem with multiple shifts and work locations», *Annals of Operations Research*, vol. 155, n° 1, p. 119–142.
- Alfares, H. K. 1998, «An efficient two-phase algorithm for cyclic days-off scheduling», *Computers & Operations Research*, vol. 25, n° 11, p. 913–923.
- Alfares, H. K. 2003, «Flexible 4-day workweek scheduling with weekend work frequency constraints», *Computers & industrial engineering*, vol. 44, n° 3, p. 325–338.
- Baker, K. R. 1976, «Workforce allocation in cyclical scheduling problems : A survey», *Operational Research Quarterly*, p. 155–167.
- Baker, K. R. et M. J. Magazine. 1977, «Workforce scheduling with cyclic demands and day-off constraints», *Management Science*, vol. 24, n° 2, p. 161–167.
- Baptiste, P., V. Giard, A. Haït et F. Soumis. 2005, *Gestion de production et ressources humaines : méthodes de planification dans les systèmes productifs*, Presses inter Polytechnique.
- Bard, J. F., C. Binici et A. H. deSilva. 2003, «Staff scheduling at the united states postal service», *Computers and Operations Research*, vol. 30, n° 5, p. 745–771.
- Bard, J. F. et L. Wan. 2006, «The task assignment problem for unrestricted movement between workstation groups», *Journal of Scheduling*, vol. 9, n° 4, p. 315–341.
- Bard, J. F. et L. Wan. 2008, «Workforce design with movement restrictions between workstation groups», *Manufacturing & Service Operations Management*, vol. 10, n° 1, p. 24–42.

- Bechtold, S. E. 1988, «Implicit optimal and heuristic labor staffing in a multiobjective, multi-location environment», *Decision Sciences*, vol. 19, n° 2, p. 353–372.
- Bechtold, S. E. et L. W. Jacobs. 1990, «Implicit modeling of flexible break assignments in optimal shift scheduling», *Management Science*, vol. 36, n° 11, p. 1339–1351.
- Bechtold, S. E. et L. W. Jacobs. 1996, «The equivalence of general set-covering and implicit integer programming formulations for shift scheduling», *Naval Research Logistics (NRL)*, vol. 43, n° 2, p. 233–249.
- Van den Bergh, J., J. Beliën, P. De Bruecker, E. Demeulemeester et L. De Boeck. 2013, «Personnel scheduling : A literature review», *European Journal of Operational Research*, vol. 226, n° 3, p. 367–385.
- Billionnet, A. 1999, «Integer programming to schedule a hierarchical workforce with variable demands», *European Journal of Operational Research*, vol. 114, n° 1, p. 105–114.
- Bouchard, M. 2005, *Optimisation des pauses dans le problème de fabrication d'horaires avec quarts de travail*.
- Brucker, P., R. Qu et E. Burke. 2011, «Personnel scheduling : Models and complexity», *European Journal of Operational Research*, vol. 210, n° 3, p. 467–473.
- Brusco, M. et T. Johns. 2011, «An integrated approach to shift-starting time selection and tour-schedule construction», *Journal of the Operational Research Society*, vol. 62, n° 7, p. 1357–1364.
- Brusco, M. J. et L. W. Jacobs. 1998, «Personnel tour scheduling when starting-time restrictions are present», *Management Science*, vol. 44, n° 4, p. 534–547.
- Burke, E. K., J. Li et R. Qu. 2010, «A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems», *European Journal of Operational Research*, vol. 203, n° 2, p. 484–493.
- Burns, R. N. et M. Carter. 1985, «Work force size and single shift schedules with variable demands», *Management Science*, vol. 31, n° 5, p. 599–607.
- Costa, M.-C., F. Jarray et C. Picouleau. 2006, «An acyclic days-off scheduling problem», *4OR*, vol. 4, n° 1, p. 73–85.
- Côté, M.-C., B. Gendron, C.-G. Quimper et L.-M. Rousseau. 2011a, «Formal languages for integer programming modeling of shift scheduling problems», *Constraints*, vol. 16, n° 1, p. 54–76.

- Côté, M.-C., B. Gendron et L.-M. Rousseau. 2007, «Modeling the regular constraint with integer programming», dans *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Springer, p. 29–43.
- Côté, M.-C., B. Gendron et L.-M. Rousseau. 2011b, «Grammar-based integer programming models for multiactivity shift scheduling», *Management Science*, vol. 57, n° 1, p. 151–163.
- Côté, M.-C., B. Gendron et L.-M. Rousseau. 2013, «Grammar-based column generation for personalized multi-activity shift scheduling», *INFORMS Journal on Computing*, vol. 25, n° 3, p. 461–474.
- Dahmen, S. et M. Rekik. 2015, «Solving multi-activity multi-day shift scheduling problems with a hybrid heuristic», *Journal of Scheduling*, vol. 18, n° 2, p. 207–223.
- Dantzig, G. B. 1954, «Letter to the editor-a comment on edie's "traffic delays at toll booths"», *Journal of the Operations Research Society of America*, vol. 2, n° 3, p. 339–341.
- Demasse, S., G. Pesant et L.-M. Rousseau. 2005, «Constraint programming based column generation for employee timetabling», dans *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Springer, p. 140–154.
- Desrosiers, J. et M. E. Lübbecke. 2005, «A primer in column generation», dans *Column generation*, Springer, p. 1–32.
- Eitzen, G., D. Panton et G. Mills. 2004, «Multi-skilled workforce optimisation», *Annals of Operations Research*, vol. 127, n° 1-4, p. 359–372.
- Elshafei, M. et H. K. Alfares. 2008, «A dynamic programming algorithm for days-off scheduling with sequence dependent labor costs», *Journal of Scheduling*, vol. 11, n° 2, p. 85–93.
- Ernst, A. T., H. Jiang, M. Krishnamoorthy, B. Owens et D. Sier. 2004a, «An annotated bibliography of personnel scheduling and rostering», *Annals of Operations Research*, vol. 127, n° 1-4, p. 21–144.
- Ernst, A. T., H. Jiang, M. Krishnamoorthy et D. Sier. 2004b, «Staff scheduling and rostering : A review of applications, methods and models», *European journal of operational research*, vol. 153, n° 1, p. 3–27.
- Ertogral, K. et B. Bamuqabel. 2008, «Developing staff schedules for a bilingual telecommunication call center with flexible workers», *Computers & Industrial Engineering*, vol. 54, n° 1, p. 118–127.
- Fischetti, M. et L. Liberti. 2012, «Orbital shrinking», dans *International Symposium on Combinatorial Optimization*, Springer, p. 48–58.

- Fischetti, M. et A. Lodi. 2003, «Local branching», *Mathematical programming*, vol. 98, n° 1-3, p. 23–47.
- Hung, R. 2006, «Using compressed workweeks to save labour cost», *European journal of operational research*, vol. 170, n° 1, p. 319–322.
- Jin, J. 2009, *Pré-affectation des tâches aux employés effectuant des tâches non-interruptibles et des activités interruptibles*, thèse de doctorat, École Polytechnique de Montréal.
- Lawler, E. L. et D. E. Wood. 1966, «Branch-and-bound methods : A survey», *Operations research*, vol. 14, n° 4, p. 699–719.
- Lequy, Q., M. Bouchard, G. Desaulniers, F. Soumis et B. Tachefine. 2012a, «Assigning multiple activities to work shifts», *Journal of Scheduling*, vol. 15, n° 2, p. 239–251.
- Lequy, Q., G. Desaulniers et M. M. Solomon. 2012b, «A two-stage heuristic for multi-activity and task assignment to work shifts», *Computers & Industrial Engineering*, vol. 63, n° 4, p. 831–841.
- Mabert, V. A. et A. R. Raedels. 1977, «The detail scheduling of a part-time work force : a case study of teller staffing», *Decision Sciences*, vol. 8, n° 1, p. 109–120.
- Musliu, N., J. Gärtner et W. Slany. 2002, «Efficient generation of rotating workforce schedules», *Discrete Applied Mathematics*, vol. 118, n° 1, p. 85–98.
- Narasimhan, R. 1997, «An algorithm for single shift scheduling of hierarchical workforce», *European Journal of Operational Research*, vol. 96, n° 1, p. 113–121.
- Neapolitan, R. E. et K. Naimipour. 2010, *Foundations of algorithms*, Jones & Bartlett Learning.
- Ni, H. et H. Abeledo. 2007, «A branch-and-price approach for large-scale employee tour scheduling problems», *Annals of Operations Research*, vol. 155, n° 1, p. 167–176.
- Purnomo, H. W. et J. F. Bard. 2007, «Cyclic preference scheduling for nurses using branch and price», *Naval Research Logistics*, vol. 54, n° 2, p. 200–220.
- Quimper, C.-G. et L.-M. Rousseau. 2010, «A large neighbourhood search approach to the multi-activity shift scheduling problem», *Journal of Heuristics*, vol. 16, n° 3, p. 373–392.
- Rekik, M., J.-F. Cordeau et F. Soumis. 2004, «Using Benders decomposition to implicitly model tour scheduling», *Annals of Operations Research*, vol. 128, n° 1-4, p. 111–133.
- Rekik, M., J.-F. Cordeau et F. Soumis. 2008, «Solution approaches to large shift scheduling problems», *RAIRO-Operations Research-Recherche Opérationnelle*, vol. 42, n° 2, p. 229–258.
- Rekik, M., J.-F. Cordeau et F. Soumis. 2010, «Implicit shift scheduling with multiple breaks and work stretch duration restrictions», *Journal of scheduling*, vol. 13, n° 1, p. 49–75.

- Restrepo, M. I., B. Gendron et L.-M. Rousseau. 2015a, «Combining Benders decomposition and column generation for multi-activity tour scheduling», cahier de recherche, Technical report, CIRRELT.
- Restrepo, M. I., B. Gendron et L.-M. Rousseau. 2015b, «A two-stage stochastic programming approach for multi-activity tour scheduling», .
- Restrepo, M. I., B. Gendron et L.-M. Rousseau. 2016, «Branch-and-price for personalized multiactivity tour scheduling», *INFORMS Journal on Computing*, vol. 28, n° 2, p. 334–350.
- Restrepo, M. I., L. Lozano et A. L. Medaglia. 2012, «Constrained network-based column generation for the multi-activity shift scheduling problem», *International Journal of Production Economics*, vol. 140, n° 1, p. 466–472.
- Salvagnin, D. et T. Walsh. 2012, «A hybrid mip/cp approach for multi-activity shift scheduling», dans *Principles and practice of constraint programming*, Springer, p. 633–646.
- Seçkiner, S. U., H. Gökçen et M. Kurt. 2007, «An integer programming model for hierarchical workforce scheduling problem», *European Journal of Operational Research*, vol. 183, n° 2, p. 694–699.
- Topaloglu, S. et I. Ozkarahan. 2003, «Implicit optimal tour scheduling with flexible break assignments», *Computers & industrial engineering*, vol. 44, n° 1, p. 75–89.