



HAL
open science

Planification robuste des roulements d'engins dans le domaine ferroviaire

Sabine Tréfond

► **To cite this version:**

Sabine Tréfond. Planification robuste des roulements d'engins dans le domaine ferroviaire. Recherche opérationnelle [math.OC]. Conservatoire national des arts et metiers - CNAM, 2014. Français. NNT : 2014CNAM0921 . tel-02088275

HAL Id: tel-02088275

<https://theses.hal.science/tel-02088275>

Submitted on 2 Apr 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



École Doctorale Informatique, Télécommunication et Électronique de Paris

Conservatoire National des Arts et métiers
Laboratoire CEDRIC

Société Nationale des Chemins de fer Français
Direction Innovation et Recherche

Thèse de Doctorat en Informatique présentée par

Sabine TRÉFOND

soutenue le 17 mars 2014

pour obtenir le grade de Docteur du Conservatoire National des Arts et Métiers

PLANIFICATION ROBUSTE DES ROULEMENTS D'ENGINS DANS LE DOMAINE FERROVIAIRE

Alain BILLIONNET
Sourour ELLOUMI

Directeur de thèse
Directrice de thèse

Professeur, ENSIIE/CEDRIC-CNAM, Paris
Maître de Conférences-HDR, ENSIIE/CEDRIC-CNAM, Paris

Housni DJELLAB
Olivier GUYON

Encadrant industriel
Encadrant industriel

Docteur, SNCF-VOYAGES, Paris
Docteur, SNCF-Direction Innovation et Recherche, Paris

Stéphane DAUZÈRE-PÉRÈS
Anass NAGIH

Rapporteur
Rapporteur

Professeur, CMP-GC, ENSM-SE, Gardanne
Professeur, Université de Lorraine / LCOMS, Metz

Rémy CHEVRIER
Farah ZEGHAL MANSOUR

Examineur
Examinatrice

Chargé de Recherche, IFSTTAR-ESTAS, Lille
Maître de Conférences-HDR, ENIT, Tunis

*À mes deux grands pères,
André Tréfond et Yvon Roquet.*

Remerciements

Je tiens avant tout à remercier mes directeurs de thèse et mes encadrants. Je remercie Alain Billionnet pour son extrême gentillesse, cela a été un privilège de pouvoir travailler avec un si bon professeur. Je remercie Sourour Elloumi, pour les séances de travail enrichissantes, mais aussi pour son écoute et ses conseils précieux. Je remercie Housni Djellab pour sa confiance, pour ses encouragements et pour les opportunités qu'il m'a offertes. Je remercie Olivier Guyon pour son investissement dès la première seconde, pour le travail en équipe et pour ses bons conseils. J'ai beaucoup appris pendant ces trois années, mais au-delà de l'encadrement scientifique d'excellente qualité, je retiendrai de chacun d'eux le soutien dont j'avais besoin et qu'ils ont su m'apporter.

Je souhaite remercier Anass Nagih et Stéphane Dauzère-Pérès d'avoir accepté d'être rapporteurs de cette thèse et d'avoir consacré du temps à sa relecture. Je remercie également Rémy Chevrier et Farah Zeghal Mansour, qui ont accepté d'être membres du jury.

Je remercie tous ceux qui ont contribué au bon déroulement de mes travaux. Nicolas Marcos et Alexandre Quivet ont su faire preuve de réactivité en tant qu'experts PRESTO et ont été d'une aide précieuse au cours du projet. Merci aussi à Enzo et à Florent pour leur travail de qualité et leur bonne humeur pendant leurs quelques mois à la Direction. Je suis également reconnaissante envers les interlocuteurs métier et les personnes ayant soutenu le projet, en particulier David De Almeida et Philippe Berthier.

Merci à tous les membres de l'équipe MOD : François, Mathilde, Caroline, Jean et Adrien, pour leur accompagnement dans la vie cheminote, mais aussi pour les parties de baby et les "petits déj" en tout genre. Un mot particulier à Francis Sourd, dont l'oreille attentive et les conseils ont été d'un grand secours à plusieurs reprises. J'en profite pour faire un clin d'œil à tous les stagiaires qui ont apporté de la bonne humeur et souvent leur petit grain de folie dans l'équipe.

Je remercie l'équipe OC pour les réunions d'échange dans la convivialité. Un petit mot à Christophe et à Cédric pour leur aide à plusieurs reprises, encore quelques jours avant la soutenance.

Je souhaite adresser un petit mot à ceux qui ont partagé mon bureau. Merci à Faten pour son soutien et les fous rires, elle est rapidement devenue plus qu'une "colloc de bureau". Merci à Pierre-Louis et à Grégoire, avec qui les moments de doute étaient vite effacés à coup de madeleines à la fleur d'oranger -avec ou sans lait- et de séances de piano impressionnantes... Un petit mot pour Cédric et Thibaut, dont les visites moins fréquentes étaient toujours un plaisir, et pour Aurélien, dont j'ai beaucoup apprécié la compagnie pendant quelques mois.

Quelques pas en arrière pour évoquer mes débuts dans la RO... Je remercie M. Boulonier, qui m'a le premier intéressée à la théorie des graphes, avant même que je sache ce que c'était. Je souhaite aussi remercier Marie-Christine Costa, qui m'a ouvert les portes du CEDRIC. Elle m'a

fait confiance pour mon premier stage, trois mois passionnants aux côtés de Nicolas et Hélène, et m'a souvent soutenue par la suite. Enfin je pense à mes professeurs à Klagenfurt, en particulier à Angelika Wiecele qui m'a guidée pendant une année et à Franz Rendl avec qui j'ai eu la chance de faire mon mémoire de fin d'études.

Un grand merci à mes proches : famille et amis ont joué un rôle important au cours de ces trois années. Ils m'ont fait rire, ils m'ont épaulée. Ils ont parfois fait preuve de compréhension et d'indulgence. Un merci particulier à Sophie, Sybille et Antoine.

Un remerciement spécial à Gaylord. Il a cru en moi et a été présent au jour le jour pendant ces trois ans, toujours avec le sourire.

Je remercie également tous ceux qui ne sont pas cités ici, mais qui m'ont fait avancer jusqu'à la thèse et pendant ma thèse.

Enfin, je tiens à remercier mes parents. Ils m'ont fait confiance et m'ont soutenue dans mes choix au fil des années. Toujours présents, ils ont su être à l'écoute et de bon conseil. Sans eux, je n'en serais pas là aujourd'hui.

Résumé

Cette thèse est consacrée à la planification robuste du matériel roulant ferroviaire dans un contexte de transport de voyageurs régional. Il s'agit de déterminer pour chaque engin une suite de trajets à réaliser sur une période donnée de façon à résister au mieux aux aléas pouvant apparaître en opérationnel. Dans ce but, nous proposons une définition et une caractérisation de la robustesse par des indicateurs propres au contexte d'étude. Nous abordons le problème par une approche structurelle différente des approches robustes classiques qui permet d'agir sur la structure d'une solution en fonction des indicateurs définis. Trois méthodes sont alors mises en œuvre dans le cadre de ce travail. Une résolution approchée par une heuristique gloutonne et une recherche locale permet d'obtenir des solutions rapidement. Une méthode de résolution approchée par génération de colonnes a été développée afin de prendre en compte plus de contraintes de façon intégrée. Une méthode basée sur un programme linéaire en nombres entiers résolu exactement traite un problème plus général. Afin de limiter le coût de la robustesse, ces méthodes sont basées sur un existant à SNCF répondant au problème de planification des ressources matérielles à coûts de production minimaux. Un outil de simulation du comportement des solutions en situation perturbée permet de comparer les solutions entre elles par évaluation des indicateurs de robustesse. Des expérimentations sur des instances réelles ont prouvé la pertinence des approches et ont mené à l'industrialisation d'un prototype.

Mots-clés : Transport ferroviaire, planification, roulements engins, robustesse, optimisation.

Abstract

Robust railway rolling-stock planning

This thesis deals with robust rolling-stock planning problems for passenger regional trains. It consists in building robust rolling-stock schedules to operate trains under technical constraints while anticipating operational disturbances that can occur. First of all, we define indicators to characterize robustness in context. We use these indicators to have an effect on solutions that we build. This structural approach is unusual compared to classical robust optimization approaches. We have implemented three methods to solve the problem. A sequential heuristic method enhanced by a local search gives solutions quickly. A column-generation method calculates approximate solutions. An integer linear program is solved exactly to obtain solutions to a global problem. These methods are based on an existing tool at SNCF that optimizes the rolling-stock planning problem to assure optimal production costs. A simulation tool evaluates robustness indicators to compare solutions. Tests on real instances have proved the relevance of the approaches and have lead to the use of a prototype in production.

Keywords : railways, rolling-stock planning, robustness, optimization.

Table des matières

I	Problème	18
1	Problème de planification des roulements d’engins	20
1.1	Description du problème	20
1.1.1	Notions préalables	20
1.1.2	Planification des roulements d’engins : problème de base	24
1.1.3	Notions complémentaires au problème de base	24
1.1.4	Planification des roulements d’engins : problème détaillé	26
1.2	Solution existante à SNCF : logiciel PRESTO	28
1.2.1	Modélisation des données par un graphe spatio-temporel	28
1.2.2	PRESTO Phase 1 : calcul de la solution de flots	32
1.2.3	PRESTO Phase 2 : construction de lignes de roulement	36
1.2.4	PRESTO Phase 3 : enchaînement des lignes de roulement	36
1.2.5	PRESTO Phase 4 : ajout de passages en maintenance	37
1.2.6	Caractéristiques de la méthode de résolution de PRESTO	37
1.3	Rôle de PRESTO dans notre travail	38
1.3.1	Limites de PRESTO	38
1.3.2	Intérêt de PRESTO	39
1.3.3	Articulation de notre travail avec PRESTO	39
2	Existence d’un roulement droit par type d’engin	40
2.1	Définitions préliminaires	40
2.2	Preuve de l’existence d’un roulement droit	43
2.2.1	Cas du graphe Gr_{AR}^p non connexe	43
2.2.2	Cas du graphe Gr_{AR}^p connexe	43
2.2.3	Conclusion	45
3	Etat de l’art	46
3.1	Planification en milieu ferroviaire	46
3.1.1	Planification des lignes	46
3.1.2	Planification des horaires	47
3.1.3	Planification des engins	47
3.1.4	Planification des agents	48
3.2	La robustesse : modélisation et approches de résolution	48
3.2.1	Motivation	48
3.2.2	Approche stochastique	49
3.2.3	Approche par scénarios	49
3.2.4	Approches par intervalles d’incertitude	50

3.2.5	Variantes des approches classiques	52
3.2.6	Approche structurelle	55
3.3	Robustesse en milieu ferroviaire	56
3.3.1	Définitions de la robustesse	56
3.3.2	Caractérisation de la robustesse	57
4	La robustesse appliquée à notre problème	66
4.1	Notre définition de la robustesse	66
4.2	Approche structurelle	68
4.2.1	Critères de construction	69
4.2.2	Critères d'évaluation	72
4.3	Approche par scénarios	75
4.4	Discussion autour de quelques approches classiques non retenues	78
4.4.1	Approche par intervalles	78
4.4.2	Light robustness	79
4.4.3	Recoverable robustness	80
4.4.4	Synthèse	80
II	Approches de résolution	82
5	Vue d'ensemble	84
5.1	Données et notations principales	84
5.2	Division du problème en sous-problèmes indépendants	85
5.3	Prise en compte de la robustesse et de la maintenance dans chaque approche	86
5.4	Articulation de chaque approche avec PRESTO	87
5.5	Résultats	87
5.6	Instances cibles	88
6	Approche par heuristique gloutonne et recherche locale	90
6.1	Spécifications du problème considéré	91
6.2	Modélisation par un graphe issu du graphe de PRESTO	91
6.3	Méthode de résolution	91
6.3.1	Evaluation d'un roulement	92
6.3.2	Heuristique gloutonne de construction d'une solution initiale	94
6.3.3	Amélioration de la solution courante par recherche locale	95
6.4	Premiers résultats et conclusions	97
6.4.1	Tests sur les instances cibles	97
6.4.2	Premières conclusions	99
6.4.3	Industrialisation et perspectives	99
7	Approche utilisant la technique de génération de colonnes	101
7.1	Spécifications du problème considéré	101
7.2	Modélisation du problème	102
7.2.1	Modélisation par un graphe issu du graphe de PRESTO	102
7.2.2	Notations	102
7.2.3	Modélisation des critères de robustesse	103
7.2.4	Modélisation du problème par un PLNE	103

7.3	Méthode de résolution	104
7.3.1	Principe : étapes de résolution	104
7.3.2	Expression du sous-problème	107
7.3.3	Modélisation du sous-problème	108
7.3.4	Résolution du sous-problème	110
7.4	Premiers résultats et conclusions	114
7.4.1	Tests sur les instances cibles	114
7.4.2	Premières conclusions	116
7.4.3	Industrialisation et perspectives	116
8	Approche par programmation mathématique et résolution exacte	118
8.1	Spécifications du problème considéré	119
8.1.1	Données d'entrée	119
8.1.2	Sortie	119
8.1.3	Problème de base	119
8.1.4	Problème complet	119
8.2	Modélisation du problème	120
8.2.1	Modélisation des données sous forme de graphe	120
8.2.2	Notations	123
8.2.3	Modélisation du problème de base par un PLNE	125
8.2.4	Contrôle des coûts de production	127
8.2.5	Modélisation des critères de robustesse	127
8.2.6	Modélisation de la maintenance	128
8.2.7	Modélisation du problème complet par un PLNE	131
8.3	Méthode de résolution	133
8.4	Premiers résultats et conclusions	133
8.4.1	Choix des paramètres	133
8.4.2	Tests sur les instances cibles	134
8.4.3	Premières conclusions	136
8.4.4	Industrialisation et perspectives	137
III	Expérimentations	139
9	Evaluation de la robustesse de roulements d'engins par simulation	141
9.1	Principe	141
9.2	Génération de scénarios de retards	141
9.3	Simulation de la propagation des retards	144
9.4	Calcul des indicateurs	146
9.5	Fiabilité des résultats	146
9.6	Limites de la méthode	149
10	Analyse des résultats	150
10.1	Informations préliminaires	150
10.1.1	Environnement	150
10.1.2	Instances de référence	150
10.2	Comparaison des différentes approches sans maintenance selon le critère <i>invret</i>	151
10.3	Corrélation entre les critères optimisés : <i>invret</i> et <i>recomp</i>	153

10.4	Estimation des critères d'évaluation	154
10.4.1	Corrélation entre <i>invret</i> et la charge de travail	155
10.5	Analyse de la maintenance	157
10.6	Conclusion	160
A	Résultats	168
A.1	Fiabilité des résultats obtenus par simulation en fonction du nombre de scénarios testés	168
B	Glossaire	172

Introduction

La principale mission de SNCF est la production de trains, un train étant défini par une origine, une destination, des dessertes, des horaires et un régime, c'est-à-dire les jours de passage dans la semaine. Pour produire un train, il faut notamment lui affecter un ou plusieurs engins, un engin correspondant à un ensemble de rames indivisible, et des agents, c'est-à-dire du personnel. L'objectif est de répondre à une demande, par exemple le transport d'un nombre de passagers estimé.

La production de trains nécessite alors la conception puis la mise en œuvre d'un plan de transport. Il s'agit de données définissant l'ensemble des trains à produire et les emplois du temps des engins et des agents.

Un plan de transport est le résultat d'un processus de planification en plusieurs étapes, ayant lieu à différents niveaux décisionnels : le niveau stratégique à long terme, le niveau tactique à moyen terme, et le niveau opérationnel. En phase de conception, au niveau tactique, différentes étapes de planification ont lieu :

- planification des lignes ;
- planification des horaires ;
- planification du matériel roulant : engins ;
- planification du personnel : agents.

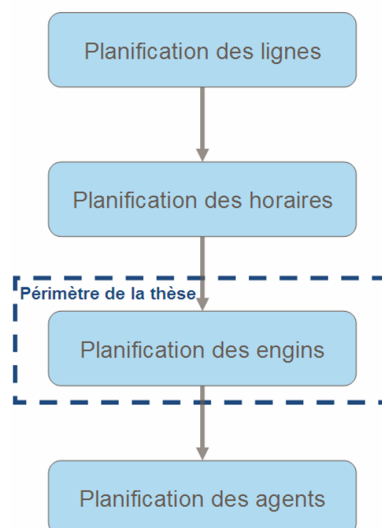


FIGURE 1 – Périmètre d'étude de la thèse.

Comme illustré par la Figure 1, la thèse s'inscrit dans le cadre de la planification des roulements d'engins (évoquée ci-après puis détaillée par la suite à la section 1.1.4). On s'intéresse en particulier à ce problème dans le cadre du transport express régional à SNCF (TER). Il s'agit alors de trains

commerciaux de transport de passagers.

La planification des roulements d'engins est la mise en relation d'un parc d'engins avec une charge de travail sur l'horizon temporel. Cette charge de travail est composée de tâches élémentaires définies sur le réseau de transport par des gares de départ et d'arrivée, des horaires de début et de fin, et une demande.

Pour répondre aux demandes de ces tâches, on dispose d'un ensemble d'engins répartis par types d'engin, aussi appelés séries d'engin. Chaque type d'engin regroupe des engins interchangeable car présentant des caractéristiques techniques communes, dont le nombre de places, c'est-à-dire le nombre de passagers pouvant être transportés.

Le problème de planification des roulements d'engins est l'affectation des engins aux différentes tâches à couvrir de façon à répondre à la demande et en respectant un certain nombre de contraintes structurelles, techniques et réglementaires. Parmi ces contraintes, on trouve :

- toutes les contraintes spatio-temporelles évidentes ;
- des contraintes de capacité : nombre d'engins disponibles, nombre d'engins maximum couvrant une même tâche ;
- des contraintes de compatibilité : incompatibilités entre deux types d'engin ou entre certains types d'engin et des tâches ;
- des contraintes liées à la sécurité : intervalle minimum réglementaire entre une arrivée et un départ, passages en maintenance à intervalles réguliers ;
- des contraintes de cyclicité : reproduction du même motif d'affectation d'une période sur l'autre.

Les données et les contraintes ne permettent pas toujours de couvrir la demande, par exemple si le nombre d'engins disponibles est insuffisant. Dans ce cas, on cherche à couvrir un nombre de tâches maximal et on énumère en sortie l'ensemble des tâches non couvertes.

Par ailleurs, la production de trains engendre des coûts :

- les coûts d'utilisation : coût d'engagement forfaitaire pour chaque engin selon son type ;
- les coûts de déplacement : coûts liés au nombre de kilomètres parcourus ou à des déplacements d'engins supplémentaires.

L'objectif principal de la planification des engins est la couverture maximale des tâches, mais il faut aussi minimiser ces coûts de production.

Le problème de planification des roulements d'engins permet d'obtenir un résultat sous forme d'un ensemble de roulements d'engins. Il s'agit de l'emploi du temps des engins sous forme de lignes : chaque ligne de roulement correspond à la succession de tâches effectuées par un engin sur une période (l'horizon temporel). Sur la période suivante, par exemple en fin de semaine, les mêmes roulements doivent être répétés pour produire à nouveau les mêmes trains. Il s'agit alors de roulements cycliques. Si les roulements ne sont pas cycliques, il n'y a pas forcément le nombre d'engins requis en certaines gares au début de la période suivante, on ne peut donc pas garantir la réalisabilité des mêmes roulements.

Le problème de planification des roulements d'engins consiste à construire des roulements d'engins cycliques et à coûts de production minimaux. Actuellement, la construction des roulements d'engins à SNCF est optimisée par le logiciel PRESTO.

Lors de la planification des roulements d'engins, les données fournies sont considérées comme des données déterministes. On suppose en effet que les horaires ou le nombre d'engins disponibles donnés sont immuables.

Or, en opérationnel, des aléas peuvent survenir : panne matérielle, attente ou absence de personnel, travaux. Les conséquences de ces aléas telles que ralentissements, arrêts, ou sur-stationnements, engendrent souvent des retards qui vont se propager dans le plan de transport. Par exemple, pour deux trajets successifs effectués par un même engin, un retard sur le premier provoquera probablement un retard sur le second. On constate alors qu'un retard de quelques minutes peut avoir un impact significatif sur le reste du trafic, engendrant une augmentation des coûts de production et une dégradation de la qualité de service initialement prévus. C'est pourquoi nous adoptons une démarche robuste.

Un des premiers objectifs de la thèse est de pouvoir définir et caractériser la robustesse dans le contexte et pour le problème étudiés : la planification des roulements d'engins en milieu ferroviaire pour le transport express régional. À partir d'une analyse bibliographique, de l'existant à SNCF, et de plusieurs rencontres avec des experts métier, nous proposons la définition suivante :

La robustesse d'un plan de transport est son aptitude à résister, à s'autoréguler en un temps limité, ou à être facilement réparable en présence d'une faible perturbation :

- résister : il n'y a aucun impact sur le plan de transport ;
- s'autoréguler : l'impact sur le plan de transport est limité, le retard est absorbé rapidement sans créer un effet "boule de neige" ;
- facilement réparable : le plan de transport doit être modifié mais il est facilement réparable par l'application de procédures préétablies lors de la conception.

Afin de construire des roulements d'engins robustes selon cette définition, nous cherchons à caractériser la robustesse d'un plan de transport et en particulier d'un roulement d'engins par des critères de robustesse adaptés au contexte d'étude. On distingue deux catégories de critères : les critères de construction et les critères d'évaluation. Les critères de construction agissent sur la structure de la solution lors de sa construction pour la rendre robuste. On peut par exemple augmenter ou diminuer le temps de stationnement des engins en gare. Les critères d'évaluation permettent d'évaluer la robustesse de roulements existants et parfois uniquement après leur réalisation. On peut par exemple s'intéresser au retard cumulé.

Le problème étudié dans la thèse est alors la planification robuste de roulements d'engins. Il s'agit de construire des roulements d'engins cycliques, à coûts de production optimaux, et robustes.

Dans ce but, nous adoptons une démarche qui diffère des méthodes robustes classiques telles que les approches par scénarios ou par intervalles. En effet, l'aléa n'est pas modélisé et pris en compte directement. L'intégration de la robustesse dans les approches de construction de roulements d'engins se fait principalement par l'optimisation de critères de robustesse identifiés et sélectionnés d'après le contexte d'étude. Cette méthode permet d'agir directement sur la structure des solutions afin de les rendre plus robustes.

Nous proposons trois approches permettant de construire des roulements robustes : une approche par une heuristique gloutonne et une recherche locale, une approche utilisant la génération de colonnes pour résoudre un programme linéaire en nombres entiers de façon heuristique, et une approche par programmation linéaire en nombres entiers avec résolution exacte. Ces approches résolvent des problèmes comparables mais distincts : les données d'entrée considérées, les critères à optimiser et les contraintes à respecter diffèrent d'une approche à une autre.

De plus, nous proposons un outil de simulation permettant l'évaluation de la robustesse d'un

plan de transport par le calcul des critères de robustesse et par l'étude du comportement du plan de transport en situation perturbée. L'étude du comportement d'un plan de transport en situation perturbée dépend de deux facteurs : la capacité à reproduire une situation perturbée et la capacité à simuler le comportement d'un plan de transport. Afin de reproduire une situation perturbée, nous étudions la modélisation de l'incertitude sur les données par des scénarios de retards. En simulant des retards sur certaines tâches et leur propagation aux tâches suivantes, on reproduit le comportement possible d'un plan de transport en situation perturbée et on peut alors calculer les critères de robustesse tels que le retard cumulé.

La Partie I de ce mémoire est consacrée à la définition du problème de planification robuste des roulements d'engins en milieu ferroviaire, en particulier pour le transport express régional.

Dans le Chapitre 1, le problème de planification des engins est défini précisément : les notions essentielles sont introduites et les objectifs et les contraintes du problème sont énumérés. PRESTO, le logiciel existant à SNCF, est décrit : chaque étape de la modélisation puis de la résolution du problème de planification des engins est détaillée. L'intérêt de ce logiciel pour nos travaux est explicité.

Dans le Chapitre 2 est présenté un résultat important pour les approches de résolution proposées. En effet, nous avons étudié les propriétés d'un graphe de modélisation du problème considéré issu de la modélisation de PRESTO. Nous avons constaté que la connexité de ce graphe garantissait l'existence d'un roulement droit, c'est-à-dire d'une solution regroupant tous les engins d'un même type en un même roulement. Les approches utilisées pour la planification de roulements d'engins robustes sont basées sur ce résultat.

Le Chapitre 3 est une synthèse de la revue de la littérature effectuée pendant la thèse, découpée en trois parties. La première partie concerne le processus de planification en milieu ferroviaire, dont la planification des roulements d'engins est une brique. La deuxième partie évoque certaines méthodes robustes classiques, telles que des méthodes par scénarios ou par intervalles. Enfin, la troisième partie cible les problématiques de robustesse en milieu ferroviaire.

Le Chapitre 4 concerne l'application de la robustesse au problème de planification des roulements d'engins. Après une définition précise de la robustesse telle qu'on la considère, une étude des différentes méthodes de prise en compte de la robustesse envisagées est présentée. Pour la construction de roulements d'engins robustes, on agit directement sur la structure de la solution par optimisation de critères de robustesse. La modélisation de l'incertitude sous forme de scénarios de retards est utilisée pour l'évaluation de roulements existants. D'autres approches, comme une approche par intervalles, nous ont semblé moins adaptées au contexte.

La Partie II est dédiée aux approches de résolution proposées pour répondre au problème de planification robuste des roulements d'engins. Trois approches réutilisant certaines étapes de PRESTO sont proposées, traitant trois problèmes comparables mais distincts. Chaque approche est adaptée à un besoin différent.

Le Chapitre 5 introduit les trois approches : données d'entrée, prise en compte de la robustesse, articulation par rapport à l'existant, résultats. Ce chapitre met en valeur les principaux points communs et différences entre les approches, donnant ainsi un premier aperçu des avantages et des inconvénients de chacune. Trois instances cibles correspondant à des instances réelles utilisées pour les premiers tests y sont définies.

Chaque approche est ensuite décrite par un chapitre propre construit selon un canevas commun : problème, modélisation, méthode de résolution, premiers résultats et conclusions. Les résultats sont

systématiquement issus de tests sur les instances cibles.

Le Chapitre 6 concerne une approche heuristique en deux phases intégrée à PRESTO : un algorithme glouton calcule une solution initiale robuste, puis une phase de recherche locale cherche à améliorer la solution obtenue. Cette approche permet une résolution rapide du problème en intégrant la robustesse par optimisation de plusieurs critères. Les solutions obtenues présentent un gain intéressant en termes de robustesse, c'est-à-dire d'après les critères optimisés. Pour sa rapidité de calcul et le gain en robustesse constaté, cette méthode est actuellement en cours d'industrialisation.

Le Chapitre 7 décrit une modélisation par programmation linéaire en nombres entiers contenant un nombre exponentiel de variables. Elle permet l'intégration de certains critères de robustesse. La méthode de résolution utilisée est une heuristique basée sur une technique de génération de colonnes combinée à une méthode par arrondi. Dans ce cadre, le sous-problème étudié est ramené à la recherche d'un chemin optimal particulier dans un graphe. En effet, les poids sur les arcs ne sont pas tous fixes comme lors de la recherche d'un plus court chemin classique. Un algorithme adapté a été mis en place pour résoudre le sous-problème. Lors des tests d'application de cette méthode, nous avons rencontré certaines difficultés : afin de ne pas saturer la mémoire du moteur de calcul, nous avons choisi d'optimiser un unique critère de robustesse. Malgré tout, le temps de calcul peut se compter en jours.

Le Chapitre 8 détaille une seconde approche par programmation linéaire en nombres entiers résolue exactement. La méthode employée permet de modéliser et de résoudre un problème moins contraint dans le sens où il donne plus de latitude pour s'écarter de la solution fournie par PRESTO à coûts de production constant ou peu dégradés. En plus de critères de robustesse, elle intègre des critères de coûts et de maintenance non pris en compte dans les deux premières méthodes. Elle remplace ainsi différentes étapes de l'existant et fournit des solutions plus robustes. Le temps de calcul est plus faible qu'en utilisant la méthode de résolution par génération de colonnes mais reste long comparé à celui de la première méthode heuristique.

La Partie III synthétise les expérimentations menées dans le cadre de l'évaluation des approches proposées. Testées une première fois indépendamment sur les instances cibles, les trois approches sont ici comparées entre elles et avec l'existant sur neuf instances de référence. L'objectif de cette dernière partie est d'évaluer l'intérêt et les inconvénients de chaque approche.

Dans ce but, le Chapitre 9 est consacré à la description d'un outil de simulation permettant de mesurer la robustesse d'une solution, c'est-à-dire d'un roulement d'engins. Une situation perturbée est modélisée par un scénario de retards : à chaque tâche est affecté un nombre de minutes de retard, décalant sa date d'arrivée. À partir d'une solution et d'un tel scénario, l'outil simule de façon simplifiée le comportement de la solution en présence des retards considérés et calcule tous les critères de robustesse envisagés. On l'utilise lors des expérimentations pour comparer les solutions de chaque approche sur un grand nombre de scénarios de retards.

Le Chapitre 10 présente les principaux résultats des tests effectués sur plusieurs instances réelles de référence. Il s'agit d'évaluer les approches proposées et de les comparer entre elles et à l'existant PRESTO. Dans ce chapitre, on constate en premier lieu un réel intérêt à l'utilisation d'une approche optimisant les critères de robustesse sélectionnés. De plus, la première méthode permet d'obtenir des solutions intéressantes rapidement. La méthode de résolution par génération de colonnes fournit des solutions équivalentes mais en un temps conséquent. Enfin, la dernière méthode donne les solutions les plus intéressantes en termes de robustesse et de maintenance.

Les dernières pages du mémoire sont consacrées à une conclusion rappelant les idées principales

de la mise en œuvre de la robustesse dans le cadre de la planification des roulements d'engins pour le transport de passagers :

- étude de la problématique, de l'existant et de l'état de l'art ;
- définition et caractérisation de la robustesse dans le contexte d'étude ;
- étude, conception et mise en œuvre de trois approches permettant la construction de roulements robustes ;
- expérimentation des approches sur des instances réelles ;
- évaluation des solutions obtenues par un module de simulation, comparaison des approches entre elles et avec l'existant ;
- industrialisation d'un premier prototype.

La conclusion présente également les perspectives d'évolution envisageables.

Première partie

Problème

Cette première partie est consacrée au problème de planification robuste des roulements d'engins. Un premier chapitre décrit précisément le problème classique de planification des roulements d'engins et la solution existante à SNCF. On s'intéresse ensuite à la littérature autour de ce sujet puis de la robustesse. Enfin, on présente une analyse de la robustesse dans le contexte d'étude afin de définir concrètement le problème de planification robuste des roulements d'engins considéré et d'envisager une ou plusieurs approches de résolution.

Chapitre 1

Problème de planification des roulements d'engins

On définit ici le problème de planification des roulements d'engins sans aborder la robustesse. À SNCF, un outil (PRESTO) répond déjà à ce problème en optimisant les coûts de production associés. Cependant, l'apparition d'aléas en opérationnel peut perturber significativement la solution optimale, motivant l'utilisation d'une méthode de planification de roulements d'engins robustes.

1.1 Description du problème

Le problème de planification des engins consiste à construire des roulements d'engins à coûts de production optimaux avec maintenance. Il faut déterminer un enchaînement de trajets et de créneaux de maintenance pour chaque engin sur une période donnée, de façon à couvrir au maximum la demande, en respectant des contraintes structurelles, techniques et réglementaires, et en minimisant les coûts de production. Ce problème est formalisé ci-après.

1.1.1 Notions préalables

Engin

Un *engin* est un ensemble de rames indivisible dont on connaît les caractéristiques techniques, comme la capacité de traction ou le nombre de places (nombre de passagers pouvant être transportés). Des engins présentant les mêmes caractéristiques techniques appartiennent à une même série de matériel (ou type d'engin) et sont par conséquent interchangeables.

Train

Un *train* est défini par une origine, une destination, des dessertes, des horaires et un régime (par exemple tous les jours du lundi au jeudi). Un train est composé de trajets unitaires appelés tâches.

Tâche

Nous définissons une *tâche* comme un parcours direct d'une gare appelée gare de départ vers une autre gare appelée gare d'arrivée, avec un horaire de début et un horaire de fin.

Deux tâches peuvent se suivre si la gare de départ de la seconde est la gare d'arrivée de la première, et si la date de début de la seconde est ultérieure à la date de fin de la première.

En entrée du problème de planification des engins, on donne l'ensemble des tâches à couvrir avec pour chacune la *demande*, exprimée comme le nombre de passagers à transporter, et le nombre d'engins maximum autorisé.

Une tâche peut être couverte par des engins de types différents, à condition de ne pas dépasser le nombre d'engins maximum autorisé et de satisfaire la demande.

Grille horaire

Une *grille horaire* est un graphique représentant l'ensemble des tâches à couvrir sur un horizon temporel (par exemple une semaine), avec pour chaque tâche la demande et le nombre d'engins maximum autorisé.

La Figure 1.1 est une grille horaire décrivant l'ensemble des tâches à couvrir sur un horizon temporel d'une semaine (du dimanche au samedi). Par exemple, la tâche T_1 part le mercredi de la gare A à 6 heures et arrive en gare B à 6 heures et 45 minutes. 400 passagers doivent être transportés par au plus 4 engins. L'exemple illustré par cette figure sera utilisé à plusieurs reprises par la suite.

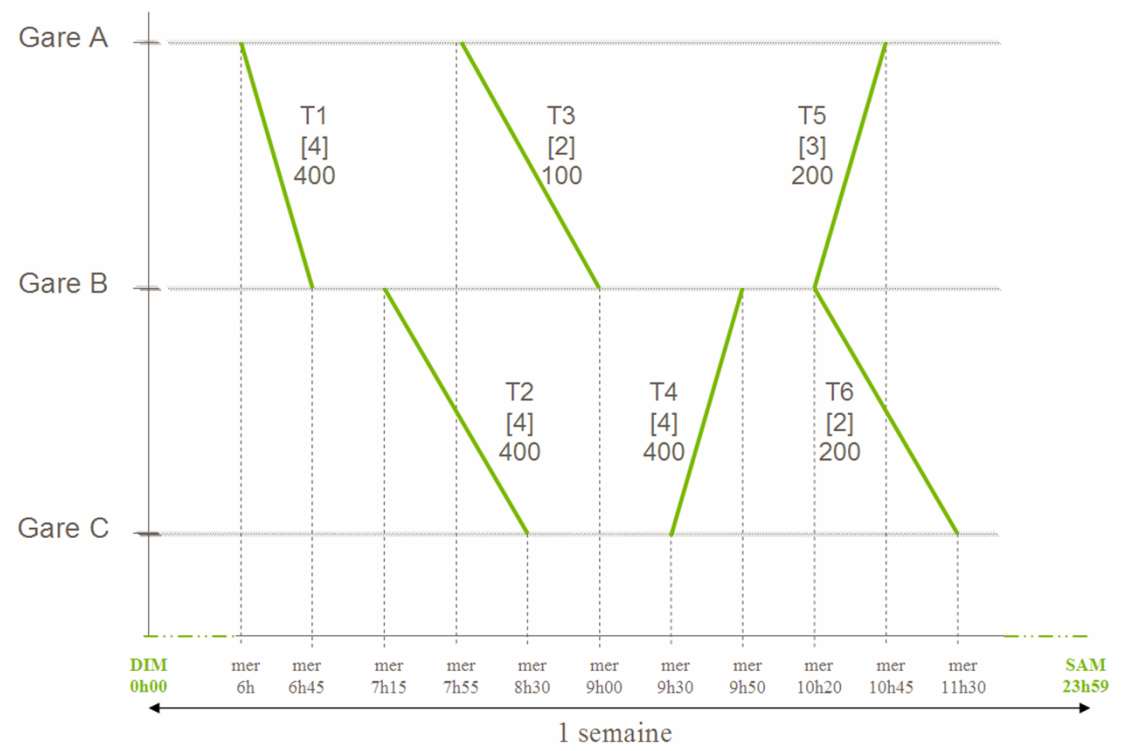


FIGURE 1.1 – Une grille horaire avec pour chaque tâche la demande et le nombre d'engins maximum (entre crochets), sur un horizon temporel d'une semaine (du dimanche au samedi).

Ligne de roulement

Une *ligne de roulement* est une succession admissible de tâches effectuées sur l’horizon temporel d’étude par un même engin. Sa gare de départ est la gare de départ de sa première tâche, et sa gare d’arrivée est la gare d’arrivée de sa dernière tâche. De même, sa date de début est celle de sa première tâche, et sa date de fin est celle de sa dernière tâche.

La Figure 1.2 est un exemple de ligne de roulement. L’engin associé effectue d’abord la tâche T_1 de la gare A à 6 heures vers la gare B à 6 heures et 45 minutes, puis la tâche T_2 , et enfin la tâche T_4 de la gare C à 9 heures et demi vers la gare B à 9 heures et 50 minutes, le mercredi. La ligne a donc pour date de début mercredi à 6 heures, pour date de fin mercredi à 9 heures et 50 minutes, pour gare de départ la gare A et pour gare d’arrivée la gare B .

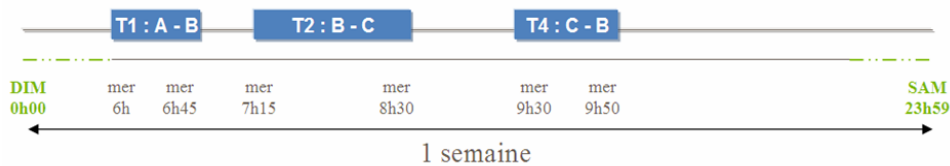


FIGURE 1.2 – Une ligne de roulement sur un horizon temporel d’une semaine.

Roulement d’engins cyclique

Un *roulement de N engins* est un enchaînement de N lignes de roulement sur l’horizon temporel.

Deux lignes ℓ_1 et ℓ_2 peuvent s’enchaîner si la première tâche de ℓ_2 peut suivre la dernière tâche de ℓ_1 , c’est-à-dire si :

- la gare de départ de ℓ_2 est la gare d’arrivée de ℓ_1 ;
- la date de début de ℓ_2 est ultérieure à la date de fin de ℓ_1 .

Un roulement de N engins est *cyclique* si on peut numéroter les lignes de 1 à N , de sorte que :

- pour tout i allant de 1 à $N - 1$, les lignes i et $i + 1$ s’enchaînent ;
- les lignes N et 1 s’enchaînent.

Le fonctionnement d’un tel roulement est le suivant. La première semaine, chaque engin i est affecté à la ligne i pour i allant de 1 à N et chaque engin effectue les tâches de sa ligne d’affectation. Les N lignes sont effectuées en parallèle. La semaine suivante est telle que :

- pour tout i allant de 1 à $N - 1$, l’engin i est affecté à la ligne $i + 1$,
- l’engin N est affecté à la ligne 1.

Et ainsi de suite pour les semaines suivantes.

La Figure 1.3 représente un roulement cyclique de cinq engins ($N = 5$) dont la capacité d’emport est de 200 passagers. La première semaine, les engins 001 et 004 sont couplés pour effectuer conjointement la tâche T_1 de 6 heures en gare A à 6 heures et 45 minutes en gare B . L’utilisation de deux engins ici permet de couvrir la demande de 400 passagers.

On remarque que dans un roulement cyclique, il y a le même nombre d’engins en une même gare au début et en fin de l’horizon temporel.

Dans l'exemple de la Figure 1.3, les engins 001, 004 et 005 commencent en gare A, et les engins 003, 004 et 005 finissent en gare A. On a bien trois engins en gare A en début et en fin d'horizon temporel. De même, l'engin 002 part de la gare B, tandis que l'engin 001 arrive en gare B, et l'engin 003 part de la gare C, tandis que l'engin 002 arrive en gare C.

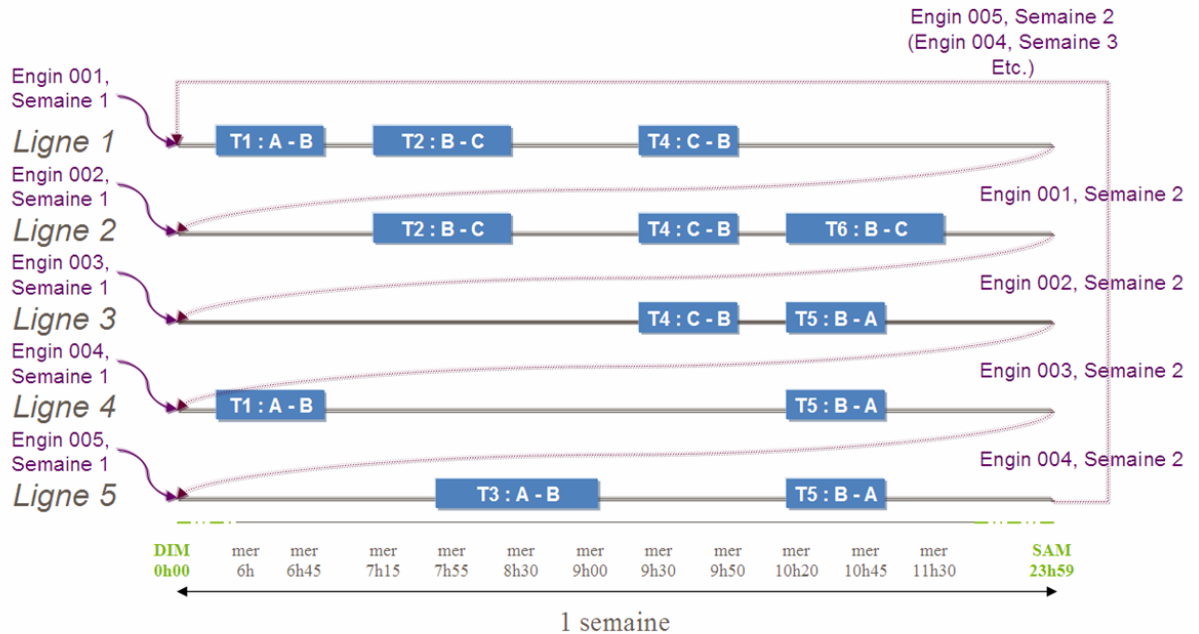


FIGURE 1.3 – Un roulement cyclique de cinq engins sur un horizon temporel d'une semaine couvrant la demande exprimée par la grille horaire de la Figure 1.1.

Comme il existe plusieurs types d'engin différents, un roulement d'engins concerne un seul type d'engin. En effet, deux engins de type différent ont des caractéristiques techniques différentes et ne sont par conséquent pas interchangeables.

Un *roulement droit* est un roulement cyclique formé par tous les engins utilisés d'un même type. Par exemple, le roulement de la Figure 1.3 est un roulement droit puisqu'il n'existe pas d'autre roulement d'engins de même type pour couvrir la demande.

Il est possible de ne pas obtenir un roulement droit pour un type d'engin. Dans ce cas, il existe plusieurs roulements cycliques pour ce type d'engin : un *sous-roulement* est un roulement formé d'une partie des engins d'un même type uniquement.

Considérons cinq engins de même type. Si les lignes 1 à 3 commencent et finissent toutes en gare A, tandis que les deux dernières commencent et finissent en gare B, il est impossible de construire un roulement droit. Il y aura alors au minimum deux sous-roulements, l'un de trois lignes et le second de deux lignes.

1.1.2 Planification des roulements d'engins : problème de base

En entrée du problème de base de planification des roulements d'engins, on a besoin des données suivantes :

- la grille horaire, c'est-à-dire l'ensemble des tâches à couvrir, incluant la demande et le nombre d'engins maximum sur chacune ;
- la pénalité de non couverture des tâches ;
- le parc des engins disponibles, divisé en plusieurs types d'engin, chaque type d'engin étant défini par des caractéristiques techniques, dont la capacité d'emport, les coûts de production, ainsi que par la taille de son parc, c'est-à-dire le nombre d'engins disponibles.

L'objectif premier non compensatoire du problème de planification des roulements d'engins est la couverture des demandes de toutes les tâches. La non couverture d'une tâche entraîne une pénalité très importante. Par ailleurs, on cherche à minimiser les coûts de production : l'utilisation d'un engin entraîne un coût d'engagement forfaitaire qui dépend du type d'engin.

Le problème est soumis à un certain nombre de contraintes structurelles, techniques et réglementaires listées ci-dessous :

- contraintes spatio-temporelles : il s'agit de toutes les contraintes évidentes liées à l'espace et au temps ; par exemple, un engin ne peut pas couvrir deux tâches différentes en même temps, ou partir de la gare B s'il vient d'arriver en gare A ;
- nombre d'engins disponibles : le parc des engins n'est pas infini, on ne peut donc pas utiliser plus d'engins que le nombre d'engins disponibles ;
- nombre d'engins maximum sur une tâche : on ne peut pas utiliser plus d'engins que le maximum autorisé pour couvrir une tâche ;
- contraintes de cyclicité : le ou les roulements obtenus en sortie doivent être cycliques.

Le problème de base de planification des roulements d'engins permet donc la construction d'un ou plusieurs roulements d'engins cycliques couvrant au mieux la demande en minimisant les coûts de production.

1.1.3 Notions complémentaires au problème de base

Le problème exposé en section 1.1.2 est la base du problème de planification des roulements d'engins. Dans cette section, on s'intéresse à de nouvelles notions correspondant à des données ou à des contraintes, et permettant une description plus détaillée du problème par la suite (cf. section 1.1.4).

Temps de crochet

Le *temps de crochet* est le temps minimum de stationnement imposé en gare entre l'arrivée d'un engin et un nouveau départ (en général quelques minutes). Il doit être impérativement respecté afin de permettre un certain nombre d'opérations de préparation des engins pour le prochain départ (descente et montée des passagers, ménage, relève de l'agent de conduite, etc.). Dans notre contexte d'étude, il dépend de la gare de stationnement.

Alors deux tâches peuvent se suivre si la gare de départ de la seconde est la gare d'arrivée de la première, et si l'intervalle entre la date de fin de la première et la date de début de la seconde

est supérieur au temps de crochet de la gare d'arrivée de la première tâche.

De plus, un intervalle de temps minimum doit également être respecté entre la date de fin d'une ligne et la date de début de la suivante.

On rappelle que les lignes de roulement sont toutes effectuées simultanément. La date de début d'une ligne est donc souvent antérieure à la date de fin de la précédente.

Dans le cas d'un horizon temporel d'une semaine, il faut comparer la date de fin d'une ligne lors de la première semaine (D_1 sur le schéma de la Figure 1.4) et la date de début de la ligne suivante lors de la deuxième semaine (D'_2 sur le schéma). Le schéma de la Figure 1.4 permet de visualiser l'intervalle entre deux lignes sur un exemple simple : D_1 est ultérieure à D_2 , mais antérieure à D'_2 . L'intervalle entre la fin de la ligne 1 et le début de la ligne 2 est donc $D'_2 - D_1$. Dans cet exemple, $D'_2 = (D_2 + \text{une semaine})$ est bien la date de début de la ligne 2 lors de la deuxième semaine.

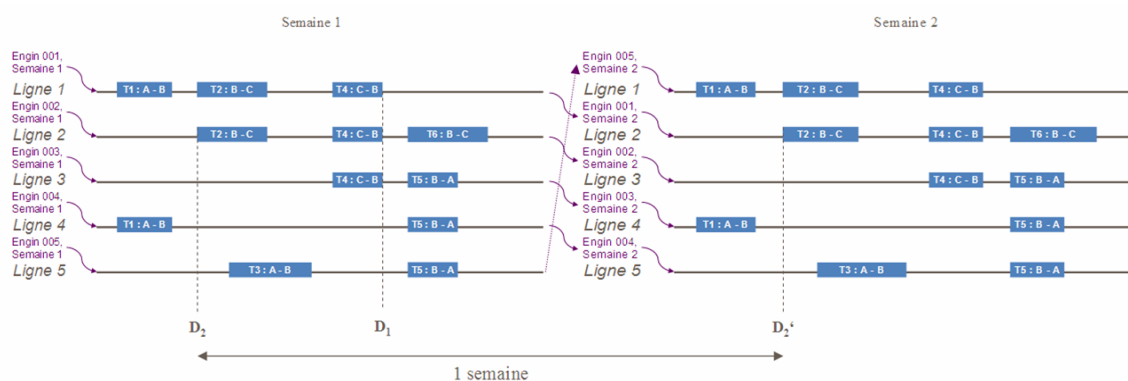


FIGURE 1.4 – Illustration de l'intervalle entre deux lignes qui se suivent (ici $D'_2 - D_1$ entre les lignes 1 et 2).

Incompatibilités matérielles

Pour des raisons techniques ou réglementaires, certains engins de type différent ne peuvent pas être couplés (incompatibilité engin-engin) et certaines tâches ne peuvent pas être couvertes par un type d'engin précis parce que la voie empruntée peut nécessiter l'utilisation d'engins spécifiques (incompatibilité engin-infrastructure). Par exemple, un engin à alimentation électrique ne peut pas circuler sur une voie non électrifiée.

Maintenance

Les engins doivent être contrôlés régulièrement dans le cadre de la maintenance préventive. Chaque engin se rend au site de maintenance imposé pour une durée fixée et à intervalles réguliers, en général tous les jours ou tous les trois jours.

Les paramètres liés à ces contraintes sont propres à un type d'engin et sont donnés en entrée :

- le technicentre (ou dépôt) est le site de maintenance auquel l'engin doit se rendre ;
- le pas d'entrée en maintenance pm_p est la fréquence à laquelle un engin doit se rendre en maintenance, en général tous les jours ou tous les trois jours ;
- la durée d'immobilisation dm_p est le temps minimum pendant lequel un engin doit rester au dépôt.

Déplacements actifs et passifs, sillons à vide

Un *déplacement actif* correspond à un engin circulant sur une tâche afin d'en couvrir la demande (ou une partie de la demande). C'est un déplacement nécessaire.

Un *déplacement passif* est un déplacement non nécessaire à la couverture de la demande : un engin supplémentaire circule sur une tâche dont la demande est déjà couverte. Il peut être ajouté pour disposer d'un engin supplémentaire dans une gare. Dans l'exemple de la Figure 1.1, la tâche T_5 peut être couverte par un seul engin transportant 200 passagers. Or en Figure 1.3, elle est couverte par trois engins, afin de repositionner deux engins supplémentaires en gare A et ainsi satisfaire les contraintes de cyclicité.

Un *sillon à vide* (ou W) est une autre solution pour repositionner un engin dans une gare différente : un nouveau trajet est créé afin de déplacer un ou plusieurs engins sans transport de passagers. Cette option est plus coûteuse que le déplacement en passif mais permet une plus grande marge de manœuvre. La création d'un sillon à vide nécessite des données supplémentaires : couples origine-destination (OD) autorisés (on ne peut pas créer de sillons de ou vers n'importe quelle gare), distances et durées des sillons à vide autorisés (les horaires ne sont pas fixés a priori).

Temps de retournement

Le *temps de retournement* est le temps de stationnement en gare entre l'arrivée d'un engin et un nouveau départ. Pour deux tâches qui se suivent sur une même ligne, le temps de retournement est l'intervalle entre la date de fin de la première tâche et la date de début de la seconde. Le temps de retournement en une gare doit être supérieur au temps de crochet de cette gare.

Creux de roulement

Un *creux de roulement* est un temps de retournement très long (plusieurs heures). Dans PRESTO, les creux de roulement facilitent l'ajout de créneaux de maintenance dans un roulement.

Unités multiples et recompositions

Une *unité simple*, notée US , est un engin couvrant une tâche seul. Or pour couvrir la demande, il est parfois nécessaire de coupler plusieurs engins¹. Une *unité multiple*, notée UM , est un ensemble de plusieurs engins couvrant une même tâche au même moment.

Dans le roulement de la Figure 1.3, l'engin 005 est en US sur la tâche T_3 car il la couvre seul, tandis que les engins 001 et 004 forment une UM pour couvrir la tâche T_1 .

Une *recomposition* est une opération permettant de coupler plusieurs engins entre eux afin d'obtenir une unité multiple ou de les découpler après utilisation.

1.1.4 Planification des roulements d'engins : problème détaillé

Le problème de planification des engins est détaillé plus finement ici : de nouvelles données et contraintes sont ajoutées au modèle de base présenté à la section 1.1.2.

Les données d'entrée du problème détaillé de planification des roulements d'engins sont :

1. Avant le départ d'une UM , une opération est requise pour accoupler les engins. Dans le cas d'engins moteurs, un seul conducteur est alors nécessaire.

- la grille horaire;
- la pénalité de non couverture des tâches;
- le parc des engins disponibles;
- les coûts de production (par type d’engin) : coût d’engagement (par engin), coûts en actif, en passif et à vide (par engin, par kilomètre);
- les incompatibilités engin-engin ou engin-infrastructure;
- les données pour la création de sillons à vide : paires origine-destination possibles, distances, durées;
- le temps de crochet (par gare);
- les données de maintenance : le dépôt, le pas d’entrée en maintenance et la durée minimum d’immobilisation (par type d’engin).

On rappelle que l’objectif premier de la planification des roulements d’engins est la couverture des tâches. Cependant, on cherche à couvrir toutes les demandes en minimisant les coûts de production : coûts d’engagement, coûts de déplacement en actif et en passif, coûts des sillons à vide.

En plus des contraintes de base détaillées en section 1.1.2, d’autres contraintes sont à considérer dans le problème détaillé. On énumère alors l’ensemble des contraintes :

- contraintes spatio-temporelles;
- respect du nombre d’engins disponibles;
- respect du nombre d’engins maximum sur une tâche;
- contraintes de cyclicité;
- respect des temps de crochet;
- incompatibilités engin-engin et engin-infrastructure;
- contraintes de maintenance : les engins doivent respecter la fréquence de passage en maintenance, en se rendant au site de maintenance (technicentre ou dépôt) imposé pour une durée fixée et à intervalles réguliers.

La planification des engins permet alors d’obtenir des roulements d’engins cycliques avec maintenance couvrant au mieux la demande en minimisant les coûts de production.

1.2 Solution existante à SNCF : logiciel PRESTO

À SNCF, Marcos (2006) s'est intéressé au problème de planification des engins. Le projet a abouti au développement du logiciel PRESTO (Planification des RESSources de Transport Optimisée) qui, à partir des données, propose des roulements d'engins optimisés, c'est-à-dire une solution couvrant au maximum les demandes des tâches tout en minimisant les coûts de production.

1.2.1 Modélisation des données par un graphe spatio-temporel

La grille horaire est d'abord modélisée sous forme d'un graphe spatio-temporel orienté sans circuit qui intègre les contraintes structurelles sur les circulations de ressources. Il s'agit en fait d'un réseau de transport. Ce graphe est caractérisé par une liste de sommets et d'arcs définis ci-dessous.

Caractéristiques d'une tâche

Nous rappelons qu'une tâche est définie comme un parcours direct d'une gare vers une autre avec un horaire de début, un horaire de fin, une demande et un nombre d'engins maximum. Par la suite, on notera $T_i = (G_i^{dep}, D_i^{deb}, G_i^{arr}, D_i^{fin})$ la tâche T_i de gare de départ G_i^{dep} , de gare d'arrivée G_i^{arr} , d'horaire de début D_i^{deb} , d'horaire de fin D_i^{fin} , de demande $dem(T_i)$ et pouvant être couverte par $cap(T_i)$ engins au maximum.

Alors deux tâches T_i et T_j peuvent se suivre si :

- T_j part de la gare d'arrivée de T_i ($G_i^{arr} = G_j^{dep}$);
- les tâches sont espacées d'un intervalle supérieur au temps de crochet $tc(G_i^{arr})$ de la gare d'arrivée de T_i : $D_j^{deb} \geq D_i^{fin} + tc(G_i^{arr})$.

Les sommets

Un sommet (G, D) correspond à un couple gare-date. Une date D est caractérisée par un jour de la semaine et un horaire.

On note D^S la date correspondant au début de l'horizon temporel et D^P la date correspondant à la fin de l'horizon temporel. Dans le cas d'un horizon temporel d'une semaine du dimanche au samedi, D^S représente le dimanche à minuit et D^P le samedi à 23 heures et 59 minutes.

Pour toute gare G , deux sommets (G, D^S) et (G, D^P) sont ajoutés.

Un nœud-source S^p par type d'engin p et un nœud-puits P , points de départ et d'arrivée fictifs des engins, sont également ajoutés.

Les arcs

Pour modéliser les données dans le graphe, différents types d'arcs sont utilisés.

Un *arc-tâche* est un arc représentant une tâche. Pour chaque tâche $T_i = (G_i^{dep}, D_i^{deb}, G_i^{arr}, D_i^{fin})$ est créé un arc-tâche dont le sommet d'origine correspond au couple (G_i^{dep}, D_i^{deb}) et le sommet de destination au couple (G_i^{arr}, D_i^{fin}) . Sur chaque arc-tâche sont indiqués le nombre d'engins maximum (entre crochets sur la Figure 1.5) et la demande à satisfaire pour couvrir cette tâche.

Un *arc-connexion* est situé juste avant ou juste après un arc-tâche et correspond à un demi temps de crochet. Afin de respecter le temps de crochet $tc(G)$ en chaque gare G , deux arcs-connexions sont créés juste avant et juste après chaque arc-tâche T_i . Le premier arc relie les sommets $(G_i^{dep}, D_i^{deb} - \frac{tc(G_i^{dep})}{2})$ et (G_i^{dep}, D_i^{deb}) , tandis que le second relie les sommets (G_i^{arr}, D_i^{fin}) et $(G_i^{arr}, D_i^{fin} + \frac{tc(G_i^{arr})}{2})$. Aucun autre arc n'est adjacent à un arc-tâche.

Dans l'exemple de la Figure 1.5, on considère des temps de crochet de 8 minutes. La tâche T_1 se termine à 6 heures et 45 minutes en gare B . L'arc-tâche correspondant relie les sommets 5 et 6. On ajoute alors un arc-connexion entre les sommets 6 et 7, correspondant à un stationnement en gare B de 6 heures et 45 minutes à 6 heures et 49 minutes. De même, un arc-connexion d'une durée de 4 minutes est ajouté entre les sommets 8 et 9, précédant ainsi l'arc-tâche entre les sommets 9 et 12.

Un *arc-attente* correspond à un temps de stationnement en gare. Pour chaque couple de sommets (G, D) et (G, D') n'ayant aucun arc-tâche incident tels que $D < D'$ et qu'il n'existe aucun sommet (G, D'') vérifiant $D < D'' < D'$, un arc-attente est créé de (G, D) vers (G, D') .

Dans l'exemple de la Figure 1.5, l'arc reliant les sommets 4 et 10 correspond à une attente de 1 heure et 55 minutes à partir de 5 heures et 56 minutes le mercredi en gare A .

Un *arc-à-vide* correspond à un sillon à vide possible. Les horaires n'étant pas fixés à l'avance, on peut créer des sillons à vide selon les besoins en respectant les données (OD autorisées, distances et durées).

Dans le graphe spatio-temporel, un ou plusieurs arcs-à-vide sont ajoutés avant une tâche de façon à pouvoir faire venir un engin de n'importe quelle autre gare si besoin. Afin de ne pas surcharger le graphe de la Figure 1.5, aucun arc-à-vide n'apparaît.

Soit une tâche T_i de A vers B . Pour toute origine G telle que le couple origine-destination (G, A) est autorisé, un arc-à-vide est ajouté de G vers A de sorte que tout engin empruntant cet arc puisse ensuite effectuer la tâche T_i . Comme illustré par la Figure 1.6, un arc-à-vide peut être considéré comme un arc-tâche de demande nulle.

Un *arc-source* est un arc partant d'une source. Pour toute gare G , un arc-source est créé de chaque source S^p à (G, D^s) , D^s correspondant à la date de début de l'horizon temporel.

Un *arc-puits* est un arc arrivant au puits. Pour toute gare G , un arc-puits est créé de (G, D^p) à P , D^p correspondant à la date de fin de l'horizon temporel.

Un exemple de graphe correspondant aux données de la grille horaire de la Figure 1.1 est donné en Figure 1.5, avec les demandes et les nombres d'engins maximum sur toutes les tâches. Les différents types d'arcs sont distingués (cf. légende de la figure) et les sommets du graphe hors sources et puits sont énumérés dans le tableau 1.1. On considère un temps de crochet de 8 minutes pour chaque gare.

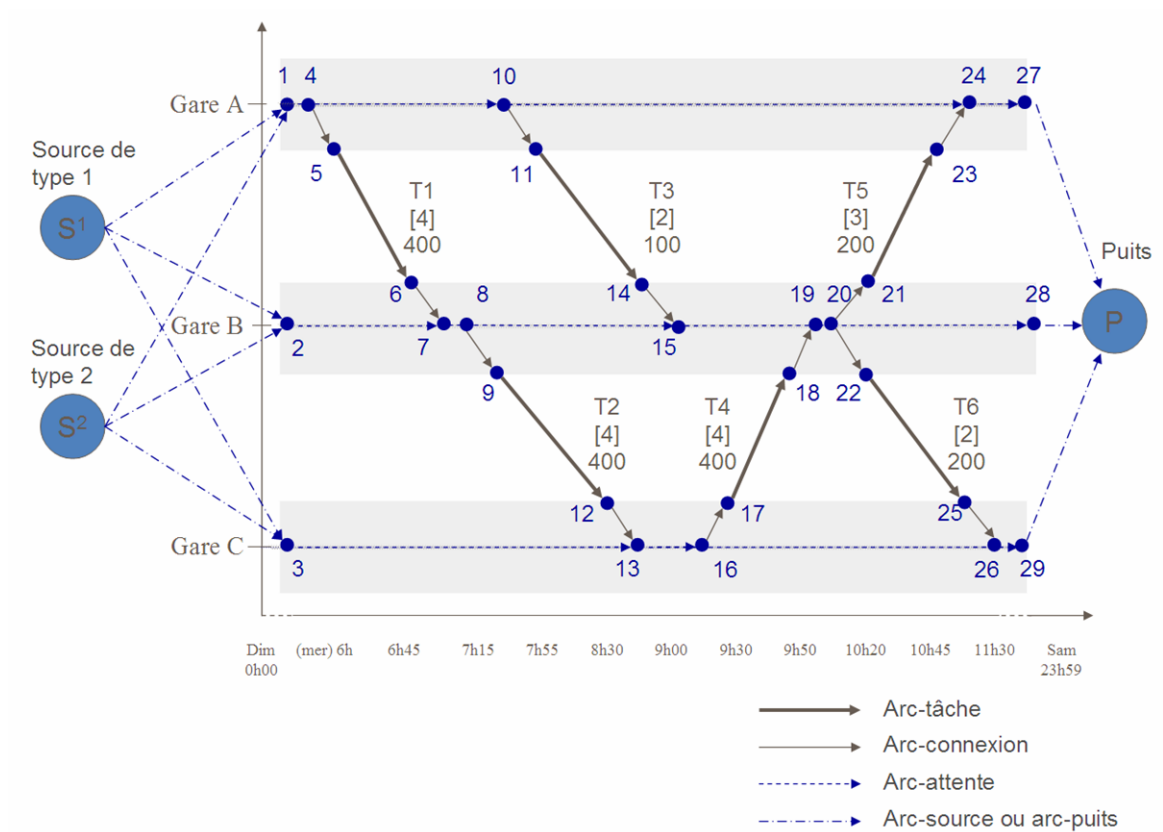


FIGURE 1.5 – Le graphe spatio-temporel correspondant à la grille horaire de la Figure 1.1, pour deux types d’engin, avec pour chaque arc-tâche la demande et le nombre d’engins maximum, sur un horizon temporel d’une semaine.

Numéro	Gare	Jour	Horaire
1	A	Dimanche	00h00
2	B	Dimanche	00h00
3	C	Dimanche	00h00
4	A	Mercredi	05h56
5	A	Mercredi	06h00
6	B	Mercredi	06h45
7	B	Mercredi	06h49
8	B	Mercredi	07h11
9	B	Mercredi	07h15
10	A	Mercredi	07h51
11	A	Mercredi	07h55
12	C	Mercredi	08h30
13	C	Mercredi	08h34
14	B	Mercredi	09h00
15	B	Mercredi	09h04
16	C	Mercredi	09h26
17	C	Mercredi	09h30
18	B	Mercredi	09h50
19	B	Mercredi	09h54
20	B	Mercredi	10h16
21	B	Mercredi	10h20
22	B	Mercredi	10h20
23	A	Mercredi	10h45
24	A	Mercredi	10h49
25	C	Mercredi	11h30
26	C	Mercredi	11h34
27	A	Samedi	23h59
28	B	Samedi	23h59
29	C	Samedi	23h59

TABLE 1.1 – Sommets du graphe de la Figure 1.5.

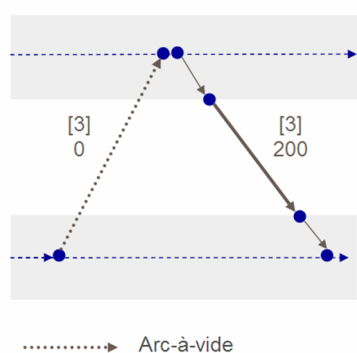


FIGURE 1.6 – Ajout d'un arc-à-vider représentant la possibilité d'emprunter un sillon à vide juste avant une tâche.

Dans le graphe spatio-temporel, un chemin de la source S^p au puits P correspond à une ligne de roulement possible pour un engin de type p . Par exemple, le chemin de la source S^1 au puits illustré par la Figure 1.7 correspond à la ligne de roulement proposée en Figure 1.2.

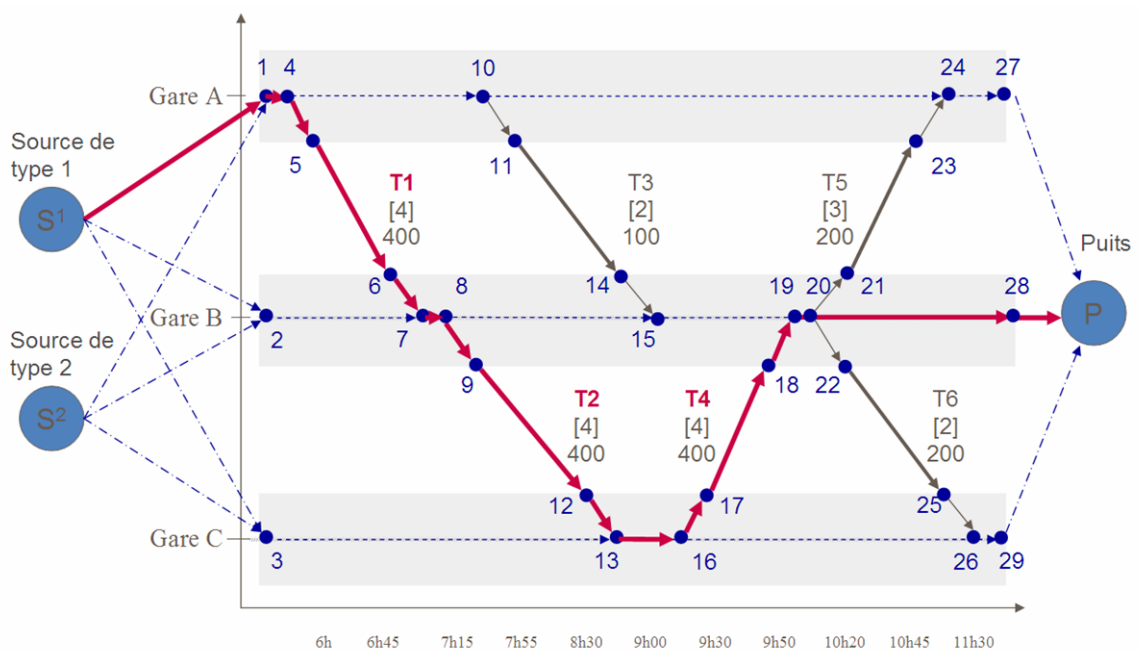


FIGURE 1.7 – Exemple de chemin dans le graphe de la Figure 1.5, correspondant à la ligne de roulement de la Figure 1.2.

L’objectif à partir de ce graphe est alors de trouver des chemins des sources au puits satisfaisant au maximum les demandes des tâches et minimisant les coûts de production, tout en respectant les contraintes exposées à la section 1.1.4 (nombre d’engins maximum par tâche, cyclicité, incompatibilités). Seule la maintenance n’est pas encore prise en compte à ce niveau et sera intégrée au roulement lors d’une phase de post-traitement.

Une fois les données modélisées sous forme de graphe spatio-temporel, le problème de planification des roulements d’engins est résolu en quatre phases par PRESTO. La Phase 1 permet le calcul du nombre d’engins de chaque type à utiliser pour assurer une couverture maximale des demandes en minimisant les coûts de production. À partir de cette solution appelée solution de flots, la Phase 2 consiste en la construction de lignes de roulement. La Phase 3 enchaîne ensuite ces lignes en respectant les contraintes de cyclicité. Enfin, la Phase 4 ajoute des créneaux de maintenance dans le roulement.

1.2.2 PRESTO Phase 1 : calcul de la solution de flots

Objectif

Il s’agit ici de trouver le nombre d’engins de chaque type à affecter à chaque tâche (flots) afin de maximiser la couverture de la demande en minimisant les coûts de production. Les coûts de production sont une agrégation de plusieurs critères à minimiser : les coûts d’engagement, les coûts de déplacement en actif et en passif, les coûts des sillons à vide.

Chaque engin utilisé entraîne un coût d’engagement forfaitaire qui peut différer d’un type d’engin à un autre. Ce critère est lié au nombre d’engins utilisés.

Les coûts de déplacement sont calculés en fonction du nombre de kilomètres parcourus. Le coût

d'un kilomètre diffère selon la nature du déplacement (actif, passif, sillon à vide) et selon le type d'engin.

Modélisation et résolution

Marcos (2006) a prouvé que la couverture maximale des tâches en minimisant les coûts de production était un problème NP-difficile de type multi-flot (avec des contraintes supplémentaires), qu'il a modélisé sous forme d'un programme linéaire en nombres entiers (PLNE). Dans la suite, on présente ce PLNE simplifié, noté P_{PRESTO} .

Notations Les notations suivantes sont propres à PRESTO (Marcos, 2006) :

Pen_a	Pénalité de non couverture de la tâche a .
CA^k	Coût d'un déplacement en actif d'un engin de type k .
CP^k	Coût d'un déplacement en passif d'un engin de type k .
E^k	Coût d'engagement d'un engin de type k .
F_a	Coût de l'arc-à- <i>vide</i> a .
K	Ensemble des types d'engin.
$GARES$	Ensemble des gares.
$Atache$	Ensemble des arcs-tâches.
A	Ensemble des arcs.
$ASource$	Ensemble des arcs-sources.
$Avid$	Ensemble des arcs-à- <i>vide</i> .
B^k	Nombre d'engins de type k disponibles.
FVa	Ensemble des types d'engin valides sur l'arc a .
$Incomp(k)$	Ensemble des types d'engin incompatibles avec le type k .
Cap^k	Capacité d'un engin de type k (par exemple en nombre de passagers).
$CapReq_a$	Capacité requise pour couvrir l'arc a .
Max	Nombre d'engins maximum sur un arc-tâche ou -à- <i>vide</i> .
N'	Ensemble des sommets du graphe.
$In(i)$	Ensemble des arcs entrants du sommet i .
$Out(i)$	Ensemble des arcs sortants du sommet i .
$as(g)$	Arc-source arrivant au premier sommet de la gare g .
$ap(g)$	Arc-puits partant du dernier sommet de la gare g .

Variables de décision Les variables de décision principales de ce programme sont entières et correspondent au nombre d'engins de chaque type circulant sur chaque arc, c'est-à-dire les flots sur chaque arc. Elles sont divisées en deux catégories :

- la variable x_a^k correspond au nombre d'engins de type k en actif sur l'arc-tâche a ;
- la variable y_a^k correspond au nombre d'engins de type k en passif sur l'arc a (définie pour tout arc du graphe).

Des variables intermédiaires sont aussi utilisées :

$$t_a^k \begin{cases} = 1 & \text{s'il existe au moins un engin de type } k \text{ circulant sur l'arc } a. \\ = 0 & \text{sinon.} \end{cases}$$

$$e_a \begin{cases} = 0 & \text{si l'arc-tâche } a \text{ est couvert.} \\ = 1 & \text{sinon.} \end{cases}$$

$$z_a \begin{cases} = 1 & \text{s'il existe au moins un engin circulant sur l'arc-à-vide } a. \\ = 0 & \text{sinon.} \end{cases}$$

PLNE Le programme linéaire en nombres entiers à résoudre est le suivant :

$$\begin{aligned}
PPRESTO : \min & \sum_{a \in \text{Atache}} Pen_a \cdot (1 - e_a) + \sum_{a \in \text{ASource}} \sum_{k \in K} E^k \cdot y_a^k \\
& + \sum_{a \in \text{Atache}} \sum_{k \in K} CA^k \cdot x_a^k + \sum_{a \in \text{Atache}} \sum_{k \in K} CP^k \cdot y_a^k + \sum_{a \in \text{Avid}} F_a \cdot z_a \\
& \sum_{k \in FVa} t_a^k + e_a \geq 1 \quad \forall a \in \text{Atache} \quad (1.1) \\
& t_a^k + t_a^{k'} \leq 1 \quad \forall A \in \text{Atache}, \forall k \in FVa, \\
& \quad \quad \quad \forall k' \in \text{Incomp}(k) \quad (1.2) \\
& \sum_{k \in FVa} x_a^k \cdot Cap^k \geq CapReq_a \cdot (1 - e_a) \quad \forall a \in \text{Atache} \quad (1.3) \\
& x_a^k \leq Max \cdot t_a^k \quad \forall a \in \text{Atache}, \forall k \in FVa \quad (1.4) \\
& \sum_{a \in \text{In}(i)} (x_a^k + y_a^k) = \sum_{a \in \text{Out}(i)} (x_a^k + y_a^k) \quad \forall i \in N', \forall k \in K \quad (1.5) \\
& \sum_{k \in K} y_a^k + \sum_{k \in FVa} x_a^k \leq Max \quad \forall a \in \text{Atache} \quad (1.6) \\
& \sum_{k \in K} y_a^k \leq Max \cdot z_a \quad \forall a \in \text{Avid} \quad (1.7) \\
& \sum_{a \in \text{ASource}} y_a^k \leq B^k \quad \forall k \in K \quad (1.8) \\
& y_{as(g)}^k = y_{ap(g)}^k \quad \forall k \in K, \forall g \text{ GARES} \quad (1.9) \\
& x_a^k \in \mathbb{N} \quad \forall a \in \text{Atache}, \forall k \in FVa \\
& y_a^k \in \mathbb{N} \quad \forall a \in A, \forall k \in FVa \\
& t_a^k \in \{0, 1\} \quad \forall a \in \text{Atache}, \forall k \in FVa \\
& e_a \in \{0, 1\} \quad \forall a \in \text{Atache} \\
& z_a \in \{0, 1\} \quad \forall a \in \text{Avid}
\end{aligned}$$

L'objectif est multicritère. Il consiste à maximiser la couverture des demandes en minimisant les coûts de production : coût d'engagement d'un engin, coûts en actif et en passif, coûts des sillons à vide.

Les contraintes (1.1) à (1.4) concernent la couverture des tâches. Les contraintes (1.1) permettent de définir les variables e_a : soit une tâche est couverte, soit elle ne l'est pas. Les contraintes (1.2)² assurent qu'une tâche ne peut pas être couverte par des engins incompatibles entre eux. Les contraintes (1.3) garantissent le respect de la demande (nombre de places suffisant par rapport

2. cette contrainte a été réécrite par rapport au PLNE d'origine apparaissant dans le manuscrit de thèse Marcos (2006)

au nombre de passagers à transporter). Enfin, les contraintes (1.4) définissent les variables t_a^k . Les contraintes (1.5) sont des contraintes de conservation de flots qui assurent la cohérence entre le nombre d'engins entrants et sortants de chaque type en chaque sommet. Les contraintes (1.6) et (1.7) garantissent que le nombre d'engins utilisés ne dépasse pas le nombre d'engins autorisé sur tout arc. Les contraintes (1.8) assurent que le nombre d'engins de type k utilisés ne dépasse pas le nombre d'engins de type k disponibles. Les contraintes (1.9) sont les contraintes de cyclicité : en chaque gare, il y a le même nombre d'engins en début et en fin d'horizon temporel.

On rappelle que la maintenance n'est pas prise en compte à cette étape.

Résultats En résolvant P_{PRESTO} , on connaît donc pour tout type d'engin k :

- $N^k = \sum_{a \in ASource} y_a^k$ le nombre d'engins de type k utilisés pour couvrir la demande;
- $flot_k(T) = x_a^k + y_a^k$ le nombre d'engins de type k affectés à la tâche T représentée par l'arc a , pour toute tâche T .

On appelle solution de flots la solution à ce problème.

Pour illustrer une solution de flots, on considère deux types d'engin 1 et 2 de capacités d'emport respectives 200 et 400. En Figure 1.8, on propose une solution de flots d'après le graphe de la Figure 1.5, utilisant uniquement des engins de type 1, tandis que la Figure 1.9 illustre une solution de flots possible en utilisant les deux types d'engin.

Les flots sont indiqués par des couples d'entiers correspondant au nombre d'engins de chaque type et par des entiers sur les arcs-sources (un arc-source correspond à un type d'engin uniquement). Par exemple, sur la Figure 1.9, la tâche T_1 est couverte par un engin de type 2 uniquement, tandis que la tâche T_5 est couverte par un engin de chaque type.

La construction d'un roulement en respectant les flots ainsi calculés garantit l'optimalité en termes de coûts de production (Marcos, 2006).

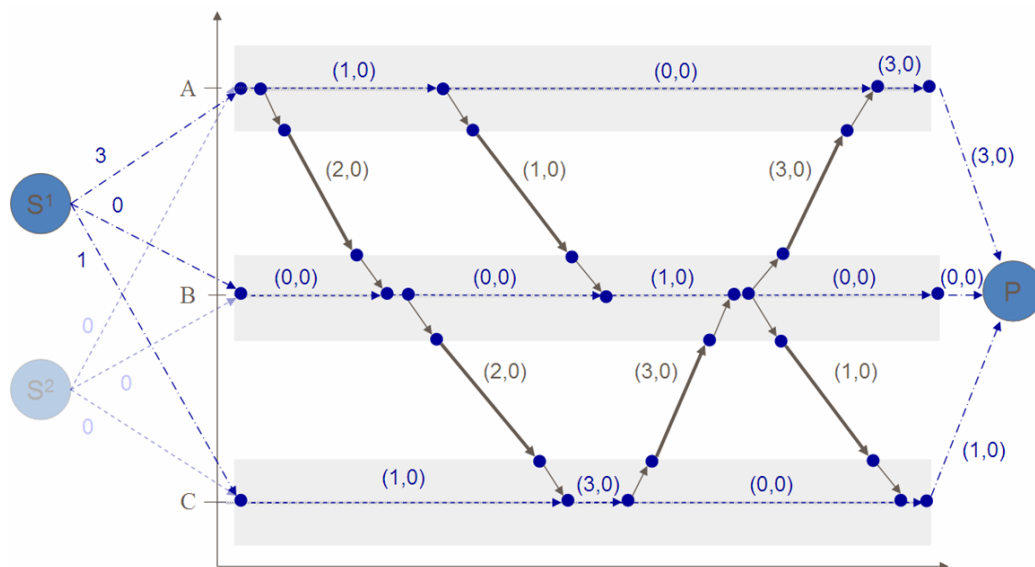


FIGURE 1.8 – Exemple de solution de flots avec un seul type d'engin correspondant à l'exemple de la Figure 1.5.

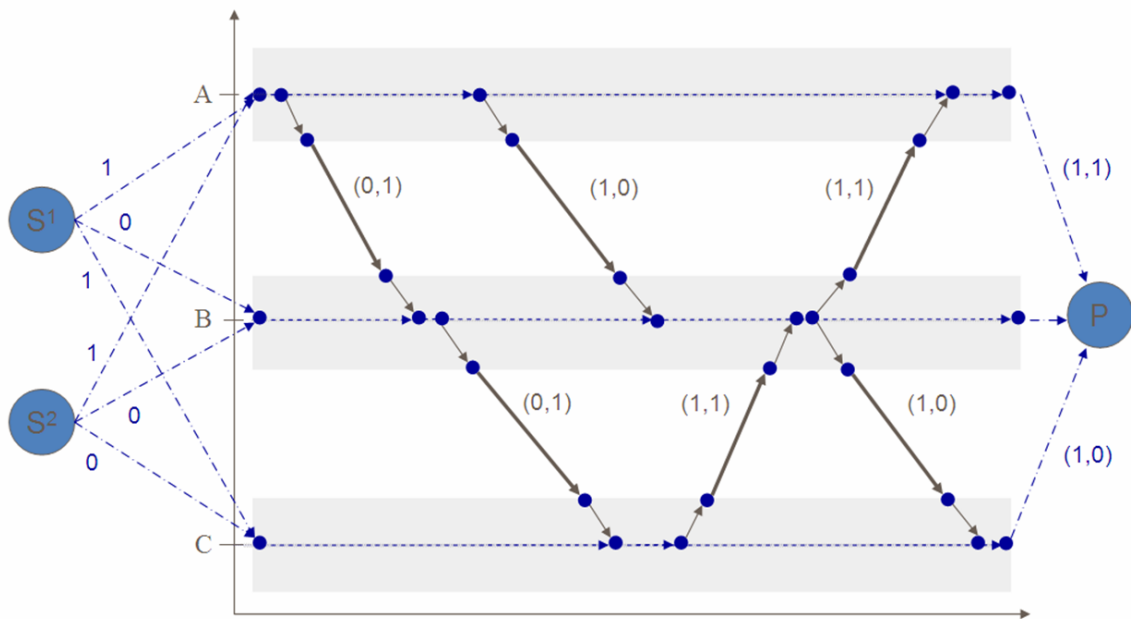


FIGURE 1.9 – Exemple de solution de flots avec deux types d’engin correspondant à l’exemple de la Figure 1.5.

1.2.3 PRESTO Phase 2 : construction de lignes de roulement

La Phase 2 de PRESTO consiste à construire des chemins des différentes sources au puits dans le graphe spatio-temporel en respectant les flots indiqués. Chaque engin effectue donc une suite de tâches (le chemin ainsi construit), qui représente une ligne de roulement. Comme le respect des flots calculés garantit l’optimalité, les chemins sont construits aléatoirement ou arbitrairement : si un sommet a plusieurs arcs sortants, on choisit l’arc suivant aléatoirement ou arbitrairement selon une règle de priorité sur les arcs (arcs-tâches en premier, arcs-connexions en second, arcs-attentes en dernier). Dans PRESTO, les arcs sont choisis arbitrairement afin de créer des creux de roulement, facilitant ainsi l’ajout de créneaux de maintenance en Phase 4 (cf. section 1.2.5).

1.2.4 PRESTO Phase 3 : enchaînement des lignes de roulement

La Phase 3 permet d’enchaîner les lignes de roulement créées à la Phase 2 afin de constituer des roulements d’engins cycliques. Lors de cette étape, il s’agit de garantir des enchaînements admissibles entre les lignes qui se suivent. Soient T_i la dernière tâche (ou le dernier sillon à vide) d’une ligne ℓ_k et T_j la première tâche (ou le premier sillon à vide) de la ligne suivante ℓ_{k+1} . Alors les tâches T_i et T_j doivent pouvoir se suivre : l’intervalle de temps entre les deux tâches doit être supérieur au temps de crochet et la gare de départ de T_j doit correspondre à la gare d’arrivée de T_i .

La structure de la solution de flots et en particulier les contraintes de cyclicité garantissent l’obtention d’une solution admissible par cette Phase 3 (Marcos, 2006), c’est-à-dire une solution sous forme de plusieurs sous-roulements cycliques.

Par ailleurs, à partir de la solution de flots, nous prouvons l'existence d'un roulement droit par type d'engin sous certaines conditions décrites à la section 2.

1.2.5 PRESTO Phase 4 : ajout de passages en maintenance

Les contraintes de maintenance sont considérées en fin de processus, lors de la Phase 4.

Pour un engin de type p , il s'agit d'ajouter des créneaux de maintenance de durée dm_p au dépôt associé tous les pm_p jours.

Le roulement est parcouru ligne par ligne afin de trouver un creux de roulement permettant l'insertion d'un créneau de maintenance. Pour pouvoir insérer un créneau de maintenance dans un creux de roulement, il faut que l'intervalle de temps correspondant soit suffisant pour aller au dépôt, effectuer la maintenance et revenir en gare : l'intervalle doit donc être supérieur à la durée minimale d'immobilisation en maintenance (dm_p) à laquelle on ajoute le double du temps de parcours de la gare courante au dépôt.

Une fois le premier créneau inséré, on cherche un nouveau creux de roulement à moins de pm_p jours du premier créneau. S'il en existe un, un nouveau créneau de maintenance est créé et on cherche à insérer le prochain créneau selon la même procédure. S'il n'y a pas de creux de roulement, le roulement est modifié dans un voisinage par échange de fins de lignes de façon à en créer. Cependant, seules les lignes de roulement sans maintenance peuvent être modifiées. De ce fait, en fin de roulement, peu de modifications sont possibles et il devient difficile voire impossible d'ajouter de nouveaux créneaux. La maintenance n'est donc pas forcément respectée.

1.2.6 Caractéristiques de la méthode de résolution de PRESTO

La résolution du problème de planification des roulements d'engins par PRESTO est divisée en plusieurs phases.

Lors de la Phase 1, la solution de flots est calculée de façon à garantir l'existence de roulements admissibles (Marcos, 2006). De plus, le respect des flots assure l'optimalité de la couverture de la demande et des coûts de production.

Lors des Phases 2 et 3, le nombre de tâches couvertes, le nombre d'engins utilisés de chaque type et le nombre de kilomètres parcourus par type de déplacement et par type d'engins sont inchangés. Les critères optimisés en Phase 1 ne sont donc pas dégradés.

En Phase 4, l'ajout de créneaux de maintenance nécessite des déplacements vers le technicentre, entraînant parfois la suppression ou l'ajout de sillons à vide. À cette étape, les coûts des sillons à vide peuvent être dégradés. En revanche, la couverture de la demande, les coûts d'engagement et les coûts de déplacement en actif et en passif sont inchangés.

La Figure 1.10 illustre le fonctionnement séquentiel de PRESTO.

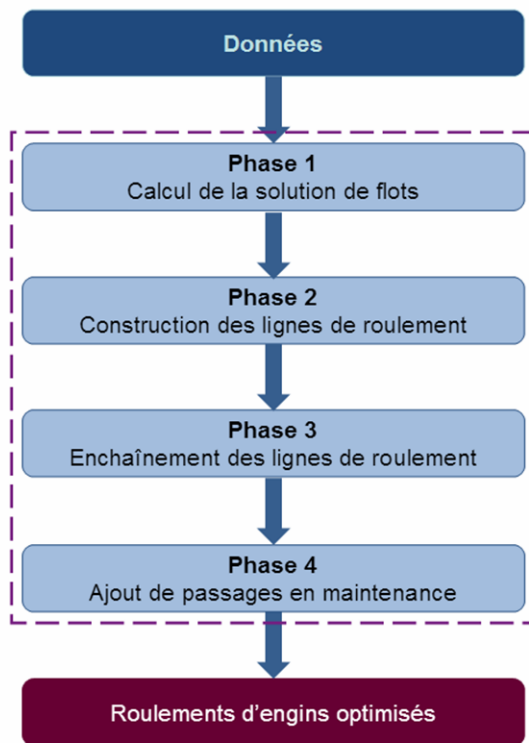


FIGURE 1.10 – Fonctionnement séquentiel de PRESTO.

1.3 Rôle de PRESTO dans notre travail

1.3.1 Limites de PRESTO

PRESTO propose des roulements d’engins couvrant au mieux la demande à faibles coûts de production en conception, c’est-à-dire plusieurs mois avant leur mise en œuvre opérationnelle.

Or des aléas peuvent apparaître, particulièrement en opérationnel : panne matérielle, indisponibilité de l’infrastructure, retard d’un agent de conduite, etc. On peut constater que de petits incidents engendrant quelques minutes de retard uniquement peuvent avoir un impact important sur le reste du plan de transport : beaucoup de trains touchés (retardés, modifiés, annulés) et beaucoup de retard en cumulé. Par conséquent, le coût et la qualité de service initialement prévus sont dégradés au cours du processus de production des trains.

La mise en place d’une planification robuste des roulements d’engins vise à prendre en compte l’incertitude liée aux aléas qui peuvent intervenir en opérationnel, afin de réduire les écarts entre les valeurs prévisionnelles et les valeurs réelles des coûts de production et de la qualité de service.

De plus, l’ajout de la maintenance lors de la Phase 4 ne garantit pas le respect des contraintes de maintenance au mieux et peut dégrader la qualité de la solution (coûts de production, robustesse dans le cas d’une planification robuste). Le traitement de la maintenance plus en amont dans la construction d’une solution permet des créneaux de maintenance plus réguliers en conservant une solution intéressante en termes de coûts de production et de robustesse.

1.3.2 Intérêt de PRESTO

PRESTO est un outil permettant de répondre rapidement par une bonne solution au problème de planification des roulements d'engins.

En effet, en sortie de la Phase 1, la solution de flots garantit une couverture maximale de la demande et des coûts de production optimaux. En particulier, le nombre d'engins utilisé est critique du point de vue du métier : il n'est pas envisageable d'ajouter un engin pour des raisons de robustesse. Il est donc intéressant et pertinent d'utiliser une telle solution dans le cadre de notre étude.

De plus, PRESTO est actuellement en production, maintenu et sujet à des évolutions régulières selon les besoins du métier. Intégrer un module supplémentaire à un tel outil facilite ainsi son utilisation à l'avenir, ce qui présente un réel intérêt industriel.

1.3.3 Articulation de notre travail avec PRESTO

Afin de garantir une solution couvrant au mieux la demande et à coûts de production optimaux ou peu dégradés, il nous a paru pertinent d'utiliser le résultat de la Phase 1 de PRESTO, en utilisant la solution de flots comme une donnée d'entrée pour notre travail. Plusieurs approches sont ensuite envisageables.

D'une part, la solution de flots fournit un nombre important d'informations :

- nombre d'engins de chaque type utilisés ;
- tâches couvertes ;
- sillons à vide utilisés ;
- flots (nombre d'engins de chaque type sur chaque tâche).

On peut décider de respecter exactement cette solution ou de ne conserver qu'une partie des informations dans le but d'avoir plus de marge de manœuvre.

D'autre part, les phases 2, 3 et 4 de PRESTO peuvent être également utilisées. Les approches que nous proposons en Partie II s'articulent différemment avec ces trois phases. Deux approches remplacent uniquement la Phase 2 de construction de lignes, cruciale dans la prise en compte de la robustesse, puis réutilisent les Phases 3 et 4 de PRESTO. Une troisième approche gère simultanément la construction d'une solution et la maintenance, traitées par les Phases 2 à 4 dans PRESTO.

Enfin, on cherche à obtenir une solution sous forme d'un roulement droit pour chaque type d'engin. Ainsi, tous les engins d'un même type effectuent les mêmes tâches, permettant une homogénéité de leur utilisation et de leur usure. Dans ce but, nous prouvons l'existence d'un roulement droit à partir de la solution de flots sous certaines conditions dans le Chapitre 2.

Chapitre 2

Existence d'un roulement droit par type d'engin

Afin de pouvoir utiliser les résultats de la Phase 1 de PRESTO (cf. section 1.3), nous cherchons à prouver que la solution de flots en sortie de P_{PRESTO} garantit l'existence d'une solution admissible sous forme d'un roulement droit par type d'engin.

Une fois la solution de flots calculée, on cherche à construire un ou plusieurs roulements d'engins respectant les flots sur les arcs. Le nombre d'engins de chaque type sur chaque arc est connu, et un roulement concerne un unique type d'engins. On peut donc traiter indépendamment chaque type d'engin, ce qui revient à résoudre N problèmes (un par type d'engin). La construction de roulements à partir de la solution de flots est alors un ensemble de sous-problèmes correspondant aux différents types d'engin. En entrée de chaque sous-problème, on considère le graphe spatio-temporel sur lequel les flots du type d'engin correspondant uniquement sont renseignés. Le graphe pour le flot de type p est appelé Gr^p dans la suite. Nous cherchons à garantir l'existence d'une solution sous forme d'un roulement droit pour chaque sous-problème. Si un tel roulement n'existe pas pour un type d'engin, nous connaissons le nombre de sous-roulements minimum dans la solution.

2.1 Définitions préliminaires

À partir du graphe spatio-temporel et de la solution de flots obtenue par la résolution P_{PRESTO} , on construit pour chaque type d'engin p le graphe Gr^p , sur lequel sont renseignées les valeurs du flot de type p uniquement. Les arcs de valeur de flot nulle pour ce type sont supprimés, laissant éventuellement apparaître des sommets isolés. Les sommets isolés sont également supprimés. On obtient un graphe avec une unique source et un puits, avec un flot de type p non nul circulant sur tous les arcs. On note $flot_p(a)$ le flot de type p sur l'arc a . On rappelle que la valeur de ce flot est le nombre d'engins de type p retenus dans la solution de flots de PRESTO (cf. section 1.2.2).

Ce graphe présente un certain nombre de propriétés issues du graphe spatio-temporel et de la solution de flots :

- réseau de transport : le graphe a une racine (le sommet source S) et n'a pas de circuit ;
- conservation des flots : pour tout sommet s , la somme des flots entrants est égale à la somme des flots sortants ;

- cyclicité : pour toute gare G , le nombre d’engins est le même en début et en fin d’horizon temporel, donc le flot passant par le sommet (G, D^S) est égal au flot passant par (G, D^P) (sommets définis à la section 1.2.2).

Le graphe Gr^p issu de la solution de flots illustrée à la Figure 1.8 est présenté en Figure 2.1.

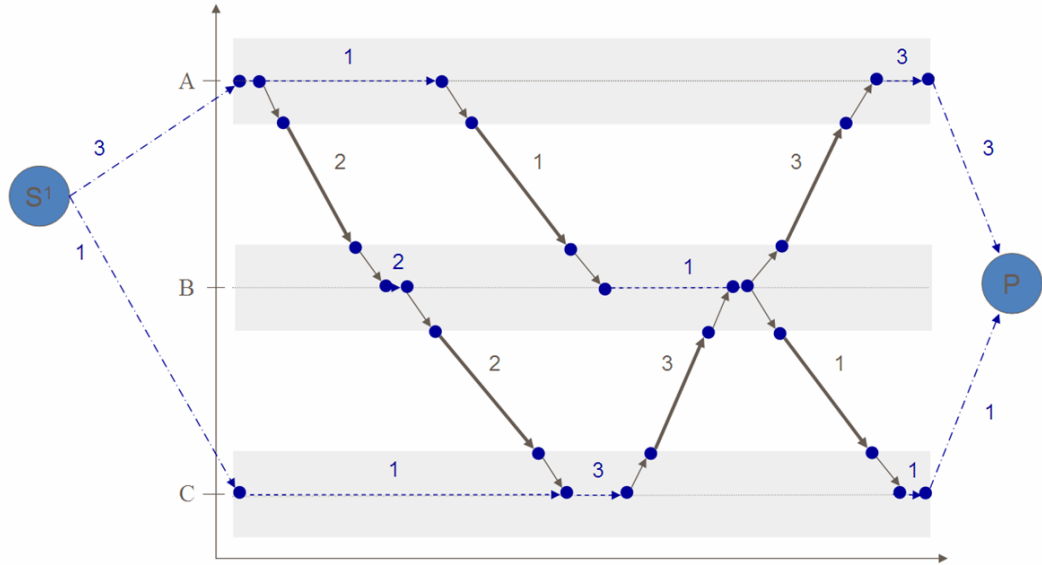


FIGURE 2.1 – Le graphe Gr^p correspondant au graphe de la Figure 1.8.

On définit deux graphes dérivés de Gr^p :

- Gr_{AR}^p : afin de modéliser le principe de cyclicité, les arcs-sources et les arcs-puits du graphe Gr^p sont supprimés et remplacés par des arcs-retours. Pour chaque gare G en laquelle le nombre d’engins en début et en fin d’horizon temporel est non nul, il existe un arc-retour sortant du sommet (G, D^P) et arrivant au sommet (G, D^S) . D’après la propriété de cyclicité du graphe Gr^p , le flot passant par le sommet (G, D^S) est égal au flot passant par le sommet (G, D^P) . Alors le flot sur l’arc retour en gare G vaut $flot_p(G, D^S)$ ($= flot_p(G, D^P)$). Le graphe Gr_{AR}^p issu du graphe Gr^p de la Figure 2.1 est présenté en Figure 2.2.
- Gr_{UN}^p : il s’agit du graphe Gr_{AR}^p dans lequel chaque flot est unitaire. Chaque arc a de Gr_{AR}^p dont le flot $flot_p(a)$ est strictement supérieur à 1 est dupliqué de façon à obtenir $flot_p(a)$ arcs de flot unitaire. La Figure 2.3 représente le graphe Gr_{UN}^p correspondant au graphe de la Figure 2.2.

Des propriétés de conservation de flots et de cyclicité du graphe Gr^p découle la propriété de conservation des flots dans les graphes Gr_{AR}^p et Gr_{UN}^p , qui se traduit par :

- pour tout sommet du graphe Gr_{AR}^p , la somme des flots entrants est égale à la somme des flots sortants ;
- pour tout sommet du graphe Gr_{UN}^p , le nombre d’arcs entrants est égal au nombre d’arcs sortants (puisque les flots sont unitaires).

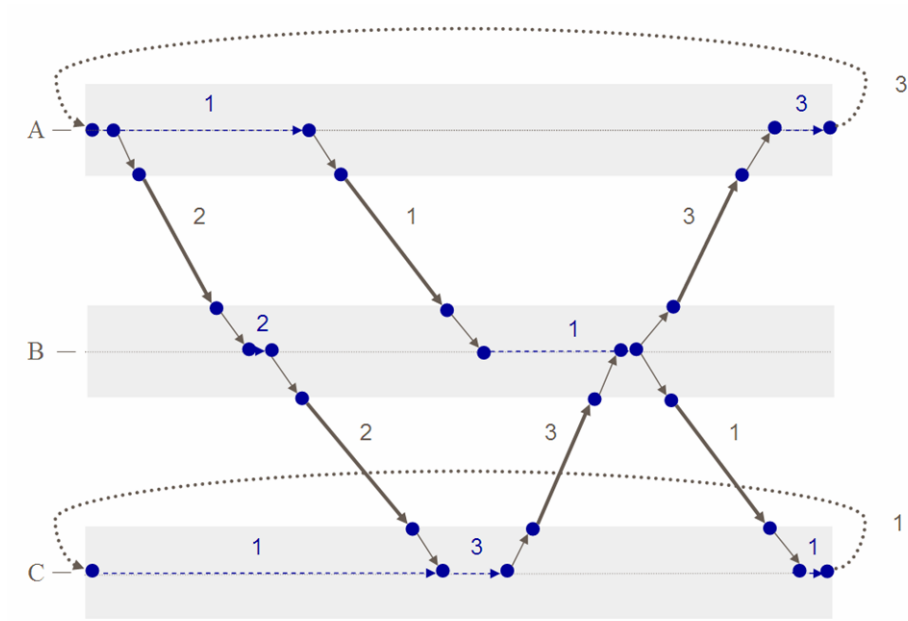


FIGURE 2.2 – Le graphe de type p avec arcs-retours (Gr_{AR}^p) issu du graphe Gr^p de la Figure 1.8.

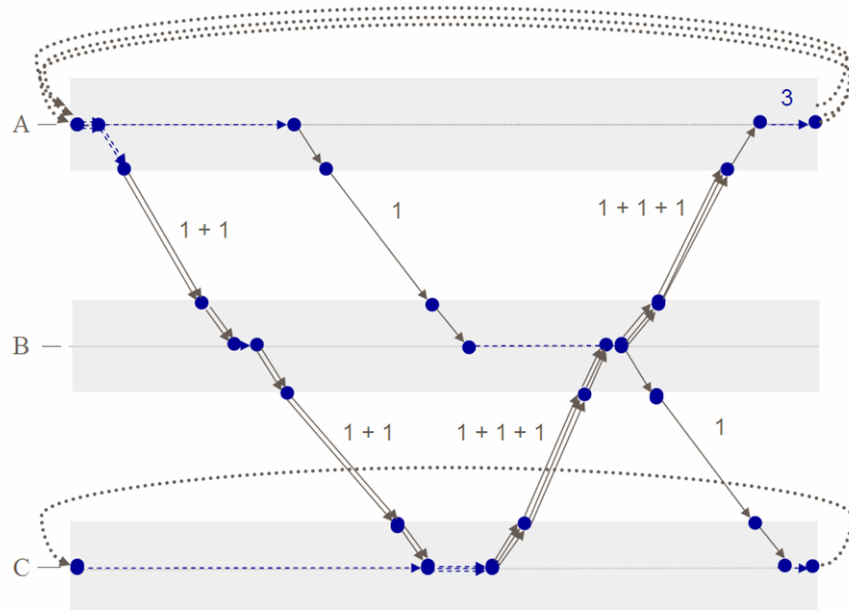


FIGURE 2.3 – Le graphe de type p avec arcs-retours et duplication des arcs multiples (Gr_{UN}^p) issu du graphe Gr_{AR}^p de la Figure 2.2.

Dans la suite, nous considérons les deux hypothèses Gr_{AR}^p est connexe et Gr_{AR}^p n'est pas connexe séparément.

2.2 Preuve de l'existence d'un roulement droit

2.2.1 Cas du graphe Gr_{AR}^p non connexe

Si Gr_{AR}^p n'est pas connexe, alors il peut être décomposé en composantes connexes. Dans ce cas, on traite chaque composante connexe comme un sous-problème indépendant (comme s'il s'agissait de types d'engin différents). On se ramène donc au cas où Gr_{AR}^p est connexe.

2.2.2 Cas du graphe Gr_{AR}^p connexe

Théorème. *À partir de la solution de flots, il existe au moins un roulement droit si et seulement si le graphe Gr_{AR}^p est connexe.*

(\Rightarrow) La première implication est directe.

Soit un roulement droit de N engins correspondant à la solution de flots calculée par PRESTO.

D'après le fonctionnement d'un roulement cyclique (cf. section 1.1.1), chaque engin effectue toutes les tâches du roulement sur N semaines.

Le parcours d'un engin sur N semaines est donc équivalent à un circuit passant au moins une fois par tous les arcs du graphe.

Le graphe Gr_{AR}^p est alors fortement connexe, donc connexe.

(\Leftarrow) Pour démontrer la seconde implication, on cherche à prouver plusieurs propositions :

$$Gr_{AR}^p \text{ connexe} \quad \Rightarrow \quad Gr_{UN}^p \text{ fortement connexe} \quad (1)$$

$$Gr_{UN}^p \text{ fortement connexe} \quad \Rightarrow \quad Gr_{UN}^p \text{ eulérien} \quad (2)$$

$$Gr_{UN}^p \text{ eulérien} \quad \Rightarrow \quad \exists \text{ un roulement droit} \quad (3)$$

1) Un raisonnement par l'absurde permet de démontrer la proposition (1).

Gr_{AR}^p est connexe, donc Gr_{UN}^p est connexe. On suppose que Gr_{UN}^p n'est pas fortement connexe.

Soit $CFC_1, CFC_2, \dots, CFC_q$ la décomposition en composantes fortement connexes de Gr_{UN}^p , avec $CFC_k = (V_k, E_k), \forall k \in [1, q]$.

Comme Gr_{UN}^p est connexe, il existe au moins un arc d'une des composantes CFC_i vers une composante CFC_j . On note $fluxout_i$ la somme des flots sur les arcs sortants de CFC_i et $fluxin_j$ la somme des flots sur les arcs entrants de CFC_j , soit ici $fluxout_i = fluxin_j > 0$.

Pour toute composante CFC_k , le nombre d'arcs $m_k \in E_k$ est le nombre d'arcs entrants en tout sommet de la composante, auquel on soustrait le nombre d'arcs entrants issus d'une autre composante fortement connexe, soit :

$$m_k = \left(\sum_{s \in V_k} |In(s)| \right) - fluxin_k$$

De même, m_k est le nombre d'arcs sortants en tout sommet de la composante, auquel on soustrait le nombre d'arcs sortants allant vers une autre composante fortement connexe, soit :

$$m_k = \left(\sum_{s \in V_k} |Out(s)| \right) - fluxout_k$$

D'après la propriété de conservation des flots dans ce graphe, on sait que :

$$\sum_{s \in V_k} |In(s)| = \sum_{s \in V_k} |Out(s)|$$

Alors $fluxin_k = fluxout_k$. Puisqu'il existe un flux positif entrant en CFC_j ($fluxin_j > 0$), il existe un flux équivalent sortant vers une autre composante CFC_l , qui ne peut pas être CFC_i , soit $fluxin_l > 0$. De même que pour CFC_j , il existe alors un flux sortant de CFC_l vers une autre composante, et ainsi de suite.

Comme le nombre de composantes fortement connexes est fini, on finit par retomber sur une composante déjà rencontrée, formant ainsi un circuit. Il y a alors contradiction avec le fait que $CFC_1, CFC_2, \dots, CFC_q$ soit la partition en composantes fortement connexes de Gr_{UN}^p .

Gr_{UN}^p est donc fortement connexe.

2) Le théorème d'Euler permet de prouver la proposition **(2)**.

D'après **(1)**, Gr_{UN}^p est fortement connexe. De plus, on sait que la propriété de conservation des flots s'applique à tout le graphe : pour tout sommet, le nombre d'arcs entrants est égal au nombre d'arcs sortants.

D'après le théorème d'Euler (version orientée) :

Un graphe orienté fortement connexe est Eulérien si et seulement si chacun de ses sommets est l'extrémité initiale et terminale du même nombre d'arcs.

Le graphe Gr_{UN}^p est donc eulérien.

3) On démontre ici la proposition **(3)**.

On sait désormais que le graphe Gr_{UN}^p est eulérien, c'est-à-dire qu'il existe un circuit dans Gr_{UN}^p passant une fois et une seule par chaque arc (de flot 1) du graphe.

À chaque flot de valeur 1 sortant d'un sommet (G, D^S) au début d'une gare G correspond un sous-chemin partant de (G, D^S) et arrivant en un sommet (G', D^P) en fin d'horizon temporel. L'ensemble de ces chemins respecte les flots sur les arcs du graphe initial Gr^p .

On associe à chacun de ces chemins un engin. Alors l'ensemble des engins forme un roulement cyclique respectant la solution de flots (cf. Figure 2.4). On a alors trouvé un roulement droit des N engins de type p .

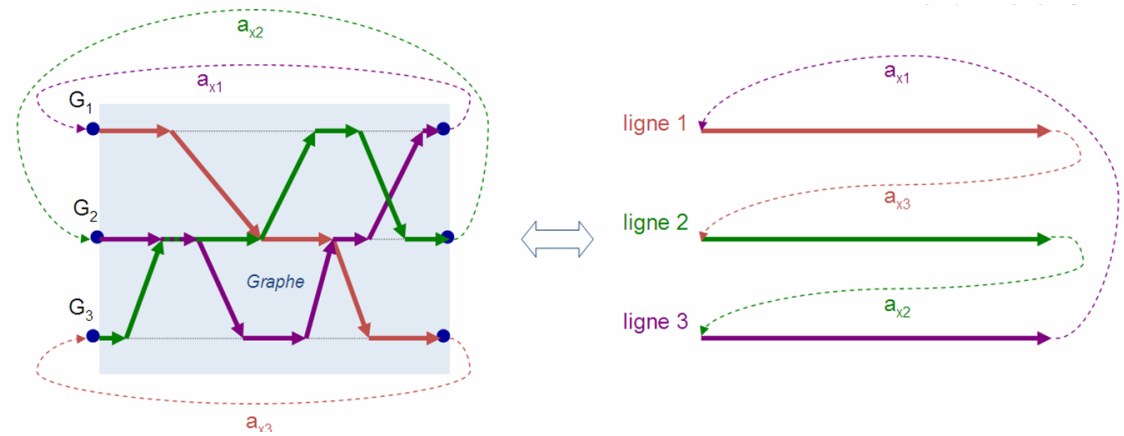


FIGURE 2.4 – Correspondance entre circuit eulérien et roulement d’engins.

2.2.3 Conclusion

On a prouvé que si Gr_{AR}^p est connexe, il existe au moins une solution sous forme d’un roulement droit des N engins du type p considéré. Dans ce cas, trouver un roulement droit revient à trouver un circuit eulérien dans le graphe Gr_{AR}^p .

Si Gr_{AR}^p n’est pas connexe, on peut décomposer le problème en plusieurs sous-problèmes qui donneront alors autant de sous-roulements. Le nombre minimum de sous-roulements d’une solution valide pour un type d’engin p à partir de la solution de flots est donc le nombre de composantes fortement connexes du graphe avec arcs-retours Gr_{AR}^p .

Chapitre 3

Etat de l'art

Ce chapitre est dédié à une étude de la littérature autour des problèmes de planification en milieu ferroviaire et de la robustesse. Cette étude apporte un éclairage sur les difficultés de l'application de la robustesse et sur les démarches adoptées en milieu industriel, en particulier en milieu ferroviaire.

3.1 Planification en milieu ferroviaire

On rappelle qu'en milieu ferroviaire, un plan de transport est le résultat d'un processus de planification en plusieurs étapes, dont certaines se placent au niveau tactique :

- planification des lignes ;
- planification des horaires ;
- planification du matériel roulant (engins) ;
- planification du personnel (agents).

L'analyse bibliographique présentée ici commence par une description de ces étapes.

3.1.1 Planification des lignes

On rappelle qu'un train est défini par une origine, une destination, des dessertes, des horaires et un régime (par exemple tous les jours du lundi au jeudi). Une ligne est un ensemble de trains qui diffèrent uniquement par leurs horaires de départ et d'arrivée. Le problème de planification des lignes consiste à définir pour chaque ligne l'origine, la destination, les dessertes et la fréquence de passage des trains de façon à satisfaire la demande estimée, c'est-à-dire de façon à garantir le transport de tous les passagers prévus de leur origine à leur destination. Deux objectifs principaux sont alors considérés : la qualité de service et les coûts de production. Ces objectifs ne sont pas forcément compatibles : minimiser les coûts de production peut détériorer la qualité de service et inversement. Les travaux sur ce sujet concernent alors souvent un des deux objectifs uniquement.

Bussieck et al (1996) s'intéressent au problème de planification des lignes en essayant d'améliorer la qualité du service fourni aux passagers à coûts fixes. Ils cherchent à minimiser le temps de trajet total de tous les passagers. Une des difficultés de ce problème est l'absence d'informations concernant les horaires des trains, puisque la grille horaire n'a pas encore été définie. Le temps d'attente en cas de correspondance est donc inconnu. Bussieck et al (1996) cherchent alors à minimiser le nombre de correspondances, voire à maximiser le nombre de passagers pouvant voyager sans correspondance.

En revanche, Goossens et al. (2005) cherchent à résoudre le problème en minimisant les coûts de production pour l'opérateur. Par ailleurs, ils intègrent la notion de types de missions : trains

régionaux, interrégionaux, etc. Des trains de types différents peuvent partager certaines gares mais n'auront pas le même comportement global. Par exemple, certains trains de type omnibus s'arrêteront à toutes les gares tandis que d'autres marqueront uniquement les arrêts principaux. Goossens et al. (2005) déterminent les différents types de missions correspondants simultanément.

3.1.2 Planification des horaires

Il s'agit de construire une grille horaire en fonction des lignes préétablies. Un train est caractérisé par un ensemble de dessertes, un horaire de départ à l'origine et un horaire d'arrivée à destination. Par ailleurs, une contrainte courante est de trouver une grille horaire périodique (cadencement), c'est-à-dire que les horaires se répètent selon une fréquence donnée. Par exemple, si la fréquence est d'une heure et s'il existe un train à 9h23, alors il y aura des trains à 10h23, 11h23, etc. Le problème de planification des horaires est un problème crucial particulièrement étudié dans la littérature. Szpigel (1973) et Petersen et Merchant (1981) sont parmi les premiers à l'avoir étudié. Depuis, de nombreux modèles et techniques de résolution ont été proposés. Caprara et al. (2002) ont proposé une modélisation par programmation linéaire, basée sur un graphe défini dans l'espace (gares) et dans le temps, et pour lequel le temps est discrétisé. Traditionnellement dans ces problèmes, la durée de tout trajet est connue. Afin d'avoir plus de marge de manœuvre et de garantir l'existence d'une solution admissible, Kroon et Peeters (2003) remettent en cause cette hypothèse et introduisent des variables correspondant à la durée des trajets. Ils peuvent ainsi par exemple allonger la durée d'un trajet par rapport à sa valeur nominale afin de garder un écart suffisant avec le train précédent.

Par ailleurs, la planification des horaires n'inclut pas forcément la planification des itinéraires dans une gare. En effet, on considère souvent une borne supérieure du nombre de trains pouvant être présents simultanément dans une gare sans s'intéresser davantage à leur itinéraire en gare. Il faut donc ensuite déterminer leur trajet au sein de la gare entre leur entrée et leur sortie, et en particulier le graphique d'occupation des voies (GOV) qui détermine à quelle voie stationne chaque train. Si à SNCF les deux problèmes sont généralement considérés séparément, ils sont pourtant très liés et font parfois l'objet d'une planification simultanée.

Planification du GOV

Le problème de planification des voies, connu sous le nom de *platforming* en anglais, est l'affectation des trains aux voies lorsqu'ils entrent en gare. Il s'agit en fait de déterminer pour tout train le chemin par lequel il va passer entre son entrée en gare et sa sortie, et la voie (platform en anglais) à laquelle il sera stationné. Ce problème est relativement simple pour des gares de petite taille, en raison du nombre de trains et de chemins possibles relativement faible. Cependant, il devient complexe pour des gares de grande taille. De plus, certaines contraintes telles qu'un intervalle de temps minimum entre deux trains qui empruntent une même voie doivent être respectées. Billionnet (2003) propose une modélisation par programmation en nombres entiers.

3.1.3 Planification des engins

La planification du matériel roulant consiste à affecter les ressources matérielles disponibles (locomotives, rames, etc.) aux trains définis précédemment, de façon à couvrir une demande de transport (nombre de passagers, capacité de traction). Nous rappelons que les ressources matérielles considérées sont appelées engins. Ce problème inclut un certain nombre de contraintes : incompatibilités matérielles (engin-engin ou engin-infrastructure), gestion des repositionnements

des engins en dépôts, maintenance, capacité des parcs de ressources limitée, etc. L'objectif principal est souvent de minimiser les coûts de production, en particulier le nombre d'engins utilisés. Entre autres, ce problème est décrit et traité par Assad (1980), Schrijver (1993), Cacchiani et al. (2010). On s'intéressera à la modélisation de Cacchiani et al. (2010) à la Section 8.1.4.

À SNCF, Marcos (2006) décrit le problème de planification des ressources matérielles dans le cas du matériel de traction (locomotives), pour lequel il propose une modélisation par programmation mathématique qui rappelle les problèmes de multiflot et une résolution en deux étapes utilisant un solveur de programmes linéaires puis une heuristique. Cette méthode, conçue pour le fret, peut être généralisée à d'autres types de matériel (TGV, TER, Intercités, etc.). Elle a d'ailleurs été mise en œuvre pour l'activité TER sous forme d'un logiciel, PRESTO, décrit à la section 1.2.

Les travaux présentés dans cette thèse portent sur le problème de planification des engins et utilisent le logiciel existant. Dans le contexte d'étude (TER : trains régionaux, transport de passagers), la demande est exprimée en nombre de passagers à transporter. Le problème de planification des roulements d'engins consiste donc à affecter les engins aux différents trains à produire, en répondant à la demande et en respectant des contraintes spatio-temporelles, techniques et réglementaires (maintenance, cyclicité).

3.1.4 Planification des agents

La planification du personnel intervient traditionnellement après la planification du matériel : d'une part pour des raisons pratiques (il est plus aisé d'affecter le personnel en fin de processus), d'autre part, les coûts d'utilisation élevés des engins sont dimensionnants. Une fois connus les trains et l'allocation du matériel roulant, il reste à affecter les agents de conduite (ADC) et les agents du service commercial des trains (ASCT) aux différents trains. Ce problème est sujet à beaucoup de contraintes. D'une part, il faut considérer les contraintes techniques. Par exemple, il existe des incompatibilités agents-engins : chaque agent a des qualifications lui permettant d'être affecté à certains types d'engins uniquement. D'autre part, la réglementation du travail doit être respectée (référentiel interne RH077 pour SNCF). Il s'agit ici de ne pas faire travailler un agent au-delà de la durée légale, de lui permettre de rentrer à son dépôt en fin de service, etc. Ce problème a notamment été étudié par Caprara et al (1998). Le problème classique a été revu par Kroon et al (2005), qui ont ajouté de nouvelles règles dans le modèle initial afin de tenir compte de certaines réclamations des agents. Par ailleurs, la planification des ressources matérielles a une influence sur l'affectation des agents et considérer ces deux problèmes séparément risque de mener à des solutions sous-optimales. Benhizia (2012) et Benhizia et al. (2013) ont étudié l'intérêt d'une planification dite intégrée, c'est-à-dire optimisant les deux ressources simultanément.

3.2 La robustesse : modélisation et approches de résolution

3.2.1 Motivation

En optimisation déterministe, on considère les données comme certaines, c'est-à-dire qu'on néglige toute incertitude potentielle sur ces données. Une solution optimale calculée d'après de telles hypothèses de départ mise en application dans des conditions réelles peut alors s'avérer très coûteuse (voire irréalisable) en cas d'imprévu. Une nouvelle solution doit souvent être mise en place, augmentant généralement le coût initialement prévu et dégradant la qualité du service proposé, comme des retards ou des suppressions de trains.

Ben-Tal et Nemirovski (1998) ont analysé le comportement des solutions optimales de divers programmes linéaires de la librairie NETLIB en cas de modification faible des données d'entrée. Ils sont arrivés à la conclusion suivante :

Dans les applications réelles de la programmation linéaire, nul ne peut ignorer l'éventualité qu'une faible incertitude dans les données puisse rendre la solution optimale courante inapplicable¹.

3.2.2 Approche stochastique

L'approche stochastique a été très souvent utilisée pour prendre en compte l'incertitude sur les données dans les problèmes d'optimisation. Il s'agit de représenter chaque donnée incertaine par une variable aléatoire définie sur un espace de probabilité et d'optimiser l'espérance d'une fonction des variables ainsi définies.

Dantzig (1955) a été un des premiers à s'intéresser à de telles techniques.

Birge (1997) propose une revue de la littérature concernant les différents modèles de programmation stochastique et leurs applications.

Un des inconvénients de l'approche stochastique vient du fait qu'on ne connaît pas forcément la loi s'appliquant à une donnée incertaine. De ce fait, il existe des approches dites robustes. Dans la suite, nous évoquons quelques exemples d'approches robustes classiques que nous détaillerons peu puisque nous considérons une approche assez différente dans nos travaux.

3.2.3 Approche par scénarios

Kouvelis et Yu (1997) proposent de traiter la robustesse avec une approche par scénarios. Cette approche consiste à prendre en compte l'incertitude sur les données en créant un ensemble de scénarios S . On note D les données d'entrée. Chaque scénario $s \in S$ correspond à une réalisation possible des données D^s . On note F_s l'ensemble des solutions réalisables en présence du scénario s . À toute solution $X \in F_s$ est associé un coût $f(X, D^s)$ (ici à minimiser) dépendant également des données d'entrée. On note X_s^* la solution optimale du problème en présence du scénario s .

Une solution robuste doit être admissible pour tout scénario réalisable ($X \in \bigcap_{s \in S} F_s$), ce qui peut paraître restrictif. Mais Kouvelis et Yu (1997) montrent que dans la plupart des cas, l'ensemble des solutions admissibles est inchangé lors de la réalisation des différents scénarios et les données d'entrée des scénarios sont définies sur les coefficients de l'objectif.

Trois critères de robustesse sont définis :

- robustesse absolue : on optimise la valeur du pire cas, c'est-à-dire le pire coût parmi toutes solutions admissibles sur tous les scénarios (critère le moins risqué mais le plus coûteux) ;

$$\min_{X \in \bigcap_{s \in S} F_s} \max_{s \in S} f(X, D^s)$$

- Déviation robuste : on minimise l'écart maximum avec la solution déterministe optimale sur tous les scénarios ;

$$\min_{X \in \bigcap_{s \in S} F_s} \max_{s \in S} (f(X, D^s) - f(X_s^*, D^s))$$

- Robustesse relative : on minimise l'écart relatif avec la solution déterministe optimale sur tous les scénarios.

$$\min_{X \in \bigcap_{s \in S} F_s} \max_{s \in S} \frac{f(X, D^s) - f(X_s^*, D^s)}{f(X_s^*, D^s)}$$

1. "In real-world applications of Linear Programming, one cannot ignore the possibility that a small uncertainty in the data can make the usual optimal solution completely meaningless from a practical viewpoint."

L'optimisation déterministe est alors un cas particulier de la méthode pour laquelle on aurait utilisé un unique scénario correspondant au cas où les données incertaines prennent leur valeurs nominales. Cette méthode a cependant deux principaux inconvénients. D'une part, le nombre de scénarios pris en compte est fini. D'autre part, une résolution exacte devient difficile lorsqu'on augmente le nombre de scénarios pris en compte. Il faut donc déterminer un nombre raisonnable de scénarios couvrant au mieux les possibilités de réalisation. On note par ailleurs que dans le cas de la planification des roulements d'engins, l'ensemble des solutions réalisables change selon les scénarios : les données incertaines étant les horaires des tâches, un retard sur une tâche peut restreindre l'ensemble des enchaînements possibles.

3.2.4 Approches par intervalles d'incertitude

Une approche par intervalles d'incertitude consiste à représenter chaque donnée incertaine d'un problème par l'intervalle des valeurs qu'elle peut prendre.

Soyster (1973) est l'un des premiers à proposer une approche robuste pour prendre en compte l'incertitude sur les données dans un programme linéaire. Soit le programme linéaire suivant :

$$\left\{ \begin{array}{l} \max \quad \sum_j c_j x_j \\ \text{s.c.} \quad \sum_j a_{ij} x_j \leq b_i \\ \quad \quad x_j \geq 0 \end{array} \right.$$

Pour prendre en compte l'incertitude sur les données du problème, on remplace chaque coefficient a_{ij} incertain par une variable aléatoire \tilde{a}_{ij} définie dans un intervalle d'incertitude $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$, avec $\hat{a}_{ij} \geq 0$. Pour tout i , on note J_i l'ensemble des indices j pour lesquels les coefficients a_{ij} sont incertains.

Soyster (1973) définit une solution robuste à ce problème comme une solution admissible pour toute réalisation possible des données. Il cherche alors une solution au pire cas, c'est-à-dire une solution au problème dans lequel chaque coefficient de la matrice prend sa pire valeur.

Il considère alors le problème robuste suivant :

$$\left\{ \begin{array}{l} \max \quad \sum_j c_j x_j \\ \text{s.c.} \quad \sum_j a_{ij} x_j + \sum_{j \in J_i} \hat{a}_{ij} x_j \leq b_i \quad \forall i \\ \quad \quad x_j \geq 0 \quad \quad \quad \forall j \end{array} \right.$$

Ce programme est toujours linéaire et résolu comme un problème déterministe. Il permet de calculer une solution absolument robuste. Une telle méthode peut être qualifiée de conservative puisqu'elle revient à choisir la pire réalisation possible pour l'ensemble des données incertaines. Ce qui peut avoir pour conséquence de dégrader fortement l'objectif par rapport à la solution optimale déterministe.

Ben-Tal et Nemirovski (1998) cherchent alors à contrôler le degré de conservatisme de la solution robuste. Le modèle qu'ils étudient est le suivant :

$$\left\{ \begin{array}{l} \min \quad c^T x \\ \text{s.c.} \quad Ax \geq b \end{array} \right.$$

où A et b sont sujets à l'incertitude. Comme dans le cas précédent, l'incertitude est prise en

compte uniquement dans les contraintes, mais il s'agit en fait du cas général, puisque dans le cas où l'incertitude porte sur la fonction objectif, ils reformulent le problème de la façon suivante :

$$\begin{cases} \min & t \\ \text{s.c.} & c^T x \leq t \\ & Ax \geq b \end{cases}$$

Les contraintes considérées sont dites "dures", elles ne doivent donc en aucun cas être violées. Si A et b peuvent prendre toutes les valeurs d'un ensemble d'incertitude \mathcal{U} , alors les contraintes doivent être satisfaites pour toute réalisation de (A, b) . Ben-Tal et Nemirovski (1998) définissent un équivalent robuste au problème initial en définissant des ensembles d'incertitude \mathcal{U} ellipsoïdaux :

$$\begin{cases} \min & c^T x \\ \text{s.c.} & Ax \geq b \quad \forall (A, b) \in \mathcal{U} \end{cases}$$

Toute solution de ce problème est appelée solution robuste. Il s'agit alors de trouver l'optimum robuste, c'est-à-dire une solution robuste avec la meilleure valeur possible pour la fonction objectif. Mais en envisageant le pire cas pour chaque coefficient, la solution est trop conservatrice. Pour définir les ensembles d'incertitude \mathcal{U} , Ben-Tal et Nemirovski (1998) introduisent alors un paramètre θ compris entre 0 et 1 permettant de mesurer le risque à prendre, c'est-à-dire de contrôler le degré de conservatisme : pour $\theta = 0$, on retrouve la solution optimale au problème initial (dite nominale), tandis que pour $\theta = 1$, la solution est conservatrice. Cependant, l'équivalent robuste n'est plus linéaire.

Bertsimas et Sim (2001, 2002) proposent une modélisation sous forme d'un programme linéaire permettant aussi de contrôler le degré de conservatisme à l'aide de nouveaux paramètres. Ils supposent que l'incertitude dans les données se situe dans le premier membre dans l'expression des contraintes. Si elle se situe dans le second membre, elle est transférée dans le premier. Par ailleurs, seule une partie des données incertaines va varier. Le programme linéaire de base est le suivant :

$$\begin{cases} \max & c^T x \\ \text{s.c.} & Ax \leq b \\ & l \leq x \leq u \end{cases}$$

On s'intéresse ici à l'incertitude dans l'objectif dans le membre de gauche des contraintes.

Chaque coefficient incertain a_{ij} de la matrice A peut prendre les valeurs comprises entre a_{ij}^- et a_{ij}^+ . On note \widehat{a}_{ij} sa perturbation maximale, soit $\widehat{a}_{ij} = a_{ij}^+ - a_{ij} = a_{ij} - a_{ij}^-$.

Afin de pouvoir contrôler le degré de conservatisme de la solution, un nouveau paramètre Γ_i est intégré au programme linéaire pour l'ensemble des coefficients de la ligne i de la matrice A . Ce paramètre, entier dans cet exemple, est compris entre 0 et le nombre total de données incertaines ($|J_i|$). En effet, on considère que les $|J_i|$ coefficients incertains ne vont pas tous varier. Γ_i correspond alors au nombre de données incertaines qui vont varier simultanément sur la ligne i . Les données variant choisies prennent comme valeur le pire cas, à savoir $a_{ij} + a_{ij}^+$, et les données ne variant pas prennent leur valeur nominale. On choisit alors le pire cas parmi toutes les possibilités d'ensembles de Γ_i valeurs variant.

La formulation robuste du programme mathématique s'écrit de la façon suivante :

$$\left\{ \begin{array}{l} \min \quad c^T x \\ \text{s.c.} \quad \sum_{j \in J} a_{ij} x_j + \max_{S_i | S_i \subseteq J_i, |S_i| = \Gamma_i} \left(\sum_{j \in S_i} \hat{a}_{ij} y_j \right) \leq b_i \quad \forall i \\ \quad \quad \quad -y_j \leq x_j \leq y_j \quad \quad \quad \forall j \\ \quad \quad \quad l \leq x \leq u \\ \quad \quad \quad y \geq 0 \end{array} \right.$$

y représente la valeur absolue de x . Pour tout i , Γ_i coefficients de la matrice prennent leur pire valeur $a_{ij} + \hat{a}_{ij}$. Pour tout ensemble S_i de Γ_i indices j , on choisit celui dont la somme des perturbations est maximale (S_i^*) et on lui impose de respecter la contrainte $\sum_{j \in J} a_{ij} x_j + \sum_{j \in S_i^*} \hat{a}_{ij} y_j \leq b_i$. Ainsi, pour toute autre réalisation des données telle que Γ_i données varient simultanément, la contrainte sera respectée.

Bertsimas et Sim (2001) prouvent alors que ce programme a un équivalent linéaire (ici pour x positif ou nul, donc y n'apparaît plus) :

$$\left\{ \begin{array}{l} \max \quad c^T x \\ \text{s.c.} \quad \sum_j a_{ij} x_j + z_i \Gamma_i + \sum_{j \in J_i} p_{ij} \leq b_i \quad \forall i \\ \quad \quad \quad z_i + p_{ij} \geq \hat{a}_{ij} x_j \quad \quad \quad \forall i, j \in J_i \\ \quad \quad \quad l_j \leq x_j \leq u_j \quad \quad \quad \forall j \\ \quad \quad \quad p_{ij} \geq 0, \forall i, j \in J_i \\ \quad \quad \quad z_i \geq 0 \quad \quad \quad \forall i \end{array} \right.$$

Ils peuvent ensuite calculer une borne supérieure de la probabilité de violer la contrainte i en fonction de Γ_i .

Mise en application

Bertsimas et Sim (2001) appliquent la méthode par intervalles qu'ils proposent à différents problèmes connus, tels que le problème du sac-à-dos. Ils illustrent à travers ces trois exemples le prix de la robustesse en fonction du degré de conservatisme. Ils constatent que plus une solution est robuste, plus les coûts sont dégradés par rapport à la solution optimale déterministe. Il faut donc faire un compromis entre coût et robustesse.

Le Maître (2008) aborde les différentes méthodes classiques par scénarios et par intervalles et essaye de les appliquer au problème industriel de migration de matériel d'un réseau de télécommunications. Concernant la méthode par intervalles, certains problèmes de modélisation l'ont orientée vers une méthode approchée également. Mais les résultats obtenus ne sont pas concluants.

3.2.5 Variantes des approches classiques

Light robustness (Fischetti et Monaci (2009))

Face à des solutions robustes souvent très coûteuses par rapport à l'objectif déterministe, Fischetti et Monaci (2009) introduisent la notion de "light robustness". Le modèle proposé est basé sur une modélisation des données par intervalles d'incertitude. Ses caractéristiques sont les suivantes :

- Mise en place d'une détérioration maximale de l'objectif à atteindre ;
- Détermination d'un objectif de robustesse à atteindre ;
- Intégration de variables de relâchement permettant la violation de certaines contraintes de robustesse "locales" en introduisant des pénalités pour non respect des contraintes, afin de garantir l'existence d'une solution admissible.

Nous nous intéressons ensuite à deux méthodes de light robustness proposées par Fischetti et Monaci (2009) : (LR_BS) et (HLR).

Première méthode (LR_BS) Le problème général considéré est le suivant :

$$\left\{ \begin{array}{l} \min z = \sum_{j \in N} c_j x_j \end{array} \right. \quad (3.1)$$

$$\left\{ \begin{array}{l} \sum_{j \in N} a_{ij} x_j \leq b_i \end{array} \right. \quad i \in M \quad (3.2)$$

$$\left\{ \begin{array}{l} x_j \geq 0 \end{array} \right. \quad j \in N \quad (3.3)$$

où N est l'ensemble des indices des variables, M l'ensemble des indices des contraintes et z le coût de la solution.

Une première approche est basée sur la méthode robuste de Bertsimas et Sim (2001), qu'on note par la suite (BS). On cherche la solution la plus robuste possible qui ne soit "pas trop éloignée" de la solution déterministe. Dans ce but, plusieurs modifications du programme linéaire de base ont été opérées :

- Un paramètre δ de détérioration maximale de la solution est intégré au programme, imposant une borne supérieure à la valeur objectif.
- Des variables de relâchement γ_i sont introduites afin de pouvoir violer si besoin la contrainte de base du modèle par intervalles de Bertsimas et Sim (2001) et ainsi garantir la faisabilité du programme. Cela induit cependant des pénalités w_i en cas de contrainte non respectée. Le nouvel objectif est alors la minimisation du nombre des contraintes violées.

Fischetti et Monaci (2009) proposent alors le modèle robuste flexible suivant :

$$\left\{ \begin{array}{l} \min z' = \sum_{i \in M} w_i \gamma_i \end{array} \right. \quad (3.4)$$

$$\left\{ \begin{array}{l} \sum_{j \in N} a_{ij} x_j + \max_{S \subseteq N, |S| \leq \Gamma_i} \{ \sum_{j \in S} \hat{a}_{ij} x_j \} - \gamma_i \leq b_i \end{array} \right. \quad i \in M \quad (3.5)$$

$$\left\{ \begin{array}{l} \sum_{j \in N} a_{ij} x_j \leq b_i \end{array} \right. \quad i \in M \quad (3.6)$$

$$\left\{ \begin{array}{l} \sum_{j \in N} c_j x_j \leq (1 + \delta) z^* \end{array} \right. \quad (3.7)$$

$$\left\{ \begin{array}{l} x_j \geq 0 \end{array} \right. \quad j \in N \quad (3.8)$$

$$\left\{ \begin{array}{l} \gamma_i \geq 0 \end{array} \right. \quad i \in M \quad (3.9)$$

Les contraintes (3.6) correspondent aux contraintes de base (3.2) inviolables. Les contraintes (3.5) s'inspirent des contraintes robustes de la méthode de Bertsimas et Sim (2002) pour Γ_i entier, auxquelles ont été ajoutées les variables de relâchement γ_i , elles-mêmes pénalisées dans l'objectif. Elles permettent donc de faire tendre la solution vers une solution robuste, en conservant la solution nominale comme solution admissible. Enfin, la contrainte (3.7) contrôle la dégradation de l'objectif par rapport à la solution déterministe optimale (de coût z^*). Ainsi, cette formulation permet de trouver une solution "aussi robuste que possible" en fonction du coût maximum autorisé. Pour $\delta = 0$, on retrouve le problème déterministe.

Seconde méthode (HLR) Une heuristique introduit des variables de relâchement dans le problème nominal. Une première étape consiste à déterminer l'ensemble des contraintes de base pouvant être potentiellement violées (incertitude). Puis, d'après une variable de détérioration maximale de la solution nominale, sont calculés le relâchement minimum à appliquer aux contraintes concernées et le relâchement moyen. Suit alors une étape de rééquilibrage entre les contraintes pour éviter les cas extrêmes.

Mise en application de (LR_BS) et (HLR) Les méthodes présentées dans ce document sont comparées d'après leur application à plusieurs cas : le problème du sac à dos, un problème basique de portfolio, et la détermination du planning (sillons) dans le domaine ferroviaire.

Les méthodes (BS), (LR_BS) et (HLR) sont d'abord appliquées successivement à plusieurs problèmes d'optimisation (dont le problème du sac à dos). On arrive à la conclusion que les méthodes de light robustness (LR_BS) et (HLR) offrent des solutions de robustesse équivalente pour un temps de calcul plus rapide.

Recoverable robustness (Liebchen et al (2009))

Chercher une solution robuste couvrant toute réalisation possible des données peut être très coûteux. Liebchen et al (2009) introduisent la notion de "recoverable robustness" (robustesse avec réparation) pour parer ce problème. Leur approche consiste à chercher une solution qui couvre une partie des scénarios réalisables uniquement, mais qui est réparable dans les autres cas. Une solution est réparable pour un scénario et pour un certain type de réparation si, en cas de réalisation dudit scénario, elle peut être rétablie en une solution admissible dans les limites du type de réparation (par exemple le coût de réparation ne doit pas dépasser un seuil donné).

Principe de la méthode La modélisation du problème de robustesse avec réparation passe par 3 étapes :

- Step O : Formulation du problème d'optimisation original ;
- Step S : Détermination des scénarios à considérer (incertitude sur les données) ;
- Step R : Définition des limites de réparabilité.

Les étapes O et S sont les étapes habituelles des méthodes robustes classiques (PLNE et modélisation de l'incertitude). La troisième étape est nouvelle et définit une classe A des algorithmes de réparation admissibles. Il s'agit alors de trouver une solution au problème d'optimisation (O) qui soit réparable pour l'ensemble des scénarios considérés (S) et dans les limites de réparabilité imposées en utilisant un algorithme A dans l'ensemble des algorithmes possibles \mathcal{A} .

Un couple solution est un couple (x, A) qui couvre l'incertitude, c'est-à-dire tel que x est une solution admissible pour un certain nombre de scénarios, et $A(x)$ est admissible pour les scénarios non couverts.

Modèles mathématiques On note P l'ensemble des solutions admissibles du problème déterministe et P_s l'ensemble des solutions admissibles dans le cas de la réalisation du scénario s . Le programme mathématique proposé consiste à trouver un couple solution $(x, A) \in P \times \mathcal{A}$ qui minimise les coûts $f(x)$, en s'assurant que pour toute réalisation d'un scénario s dans l'ensemble des scénarios considérés S , l'application de l'algorithme A à la solution x permet de conserver l'admissibilité. Le modèle proposé est le suivant :

$$\begin{cases} \min_{(x,A) \in P \times \mathcal{A}} & f(x) \\ \text{s.c.} & A(x, s) \in P_s \quad \forall s \in S \end{cases}$$

Ce modèle peut être légèrement modifié pour s'adapter au cas où l'algorithme choisi dépend du scénario qui se réalise. En effet, un même algorithme peut ne pas forcément permettre de rétablir x en une solution admissible pour tous les scénarios réalisables, il faudra donc retenir plusieurs algorithmes et choisir celui qui convient en cas de réalisation de tel ou tel scénario. Par ailleurs, on peut vouloir imposer des limites de coût à la réparation (par exemple l'écart par rapport à la solution robuste x). Ces limites sont introduites dans le programme sous la forme de variables $\lambda \in \Lambda$ et telles que $\lambda < \lambda_0$. Le modèle général devient alors le suivant :

$$\begin{cases} \min_{(x,A,\lambda) \in P \times \mathcal{A} \times \Lambda} & f(x) + g(\lambda) \\ \text{s.c.} & A(x, s, \lambda) \in P_s \quad \forall s \in S \end{cases}$$

Mise en application La méthode proposée est destinée à une utilisation en milieu ferroviaire. Plus précisément, les problèmes de construction de planning et de gestion des retards sont détaillés. Cependant, aucune information chiffrée n'est exposée.

Un **retard source** est défini comme un arrêt prolongé en gare non causé par un précédent retard, pouvant entraîner d'autres retards. Une bonne planification (un bon planning et a fortiori un bon plan de transport) doit contenir des zones "tampons" qui peuvent absorber les "petits" retards et ainsi limiter les effets d'un tel retard sur le reste du système.

L'étape de définition des limites de réparabilité (Step R) consiste à définir des algorithmes de réparation à mettre en place en opérationnel. Il existe deux types d'opérations de réparation en milieu ferroviaire : retarder ou annuler des tâches. Retarder les tâches propage cependant le retard au reste du système.

Différents algorithmes de réparation concernant la "gestion des retards" sont présentés. Le premier algorithme autorise uniquement le retardement des trains et non leur annulation, le second utilise les deux types d'opérations, puis des améliorations sont proposées dans un souci de réalisme quant au contexte (décisions à prendre rapidement en temps réel suite à la réalisation de scénarios, mais sans connaissance des scénarios qui vont se réaliser par la suite, et en suivant certaines règles de gestion des retards). L'expression donnée pour calculer le prix de la robustesse correspond au temps de trajet supplémentaire dans le planning robuste obtenu par rapport à la solution déterministe optimale.

3.2.6 Approche structurelle

Les méthodes classiques ne sont pas toujours applicables à des problèmes industriels. Dans le cas présenté par Le Maître (2008), la méthode par intervalles n'est pas applicable car elle requiert une modélisation précise. Il nous a également semblé que notre modélisation ne permettait pas l'application d'une méthode par intervalles (cf. section 4.4.1). Quant à une méthode par scénarios, une résolution exacte n'est pas toujours envisageable pour des raisons de complexité de calcul.

Smith et Johnson (2006) traitent de la planification robuste du matériel dans l'aéronautique. Ils n'utilisent pas les méthodes classiques, mais appliquent des règles liées au contexte d'étude et permettant d'agir directement sur la structure de la solution. D'une part, ils basent leur planification sur un réseau dit en étoile (un aéroport principal est considéré comme étant point de départ et d'arrivée de tous les vols et les autres aéroports n'ont pas de connexions entre eux). D'autre

part, ils tentent d'appliquer une règle dite de "pureté", qui consiste à n'affecter que des engins d'un même type à une ligne commerciale. En effet, en cas de problèmes, il est alors plus facile de remplacer un avion par un autre. Ils obtiennent ainsi des solutions satisfaisantes.

Aloulou et al (2013) s'intéressent au même problème de planification robuste du matériel, mais ils proposent également d'ajuster les horaires des vols pour gagner en robustesse. Le programme mis en œuvre permet d'ajouter des marges temporelles entre deux vols de correspondance : en cas de retard, on cherche à garantir les correspondances des passagers et du matériel.

Une approche structurelle nous a semblé adaptée pour la résolution du problème de planification robuste des roulements d'engins en milieu ferroviaire. La mise en place d'une telle approche nécessite en premier lieu de définir la robustesse précisément pour notre contexte d'étude, puis d'identifier des règles ou des critères de robustesse permettant de construire ou d'évaluer des solutions robustes.

3.3 Robustesse en milieu ferroviaire

3.3.1 Définitions de la robustesse

Définir la robustesse n'est pas une tâche aisée. Globalement, on considère qu'une solution est robuste si elle résiste à une légère modification des données. Cette définition très générale est à préciser selon le contexte, en particulier : qu'appelle-t-on "résister" et "modification légère" ? Résister peut signifier que la solution doit rester admissible ou qu'elle est peu dégradée, ce qui introduit à nouveau une imprécision ("peu" dégradée).

La robustesse est à définir en fonction du contexte. En milieu ferroviaire, beaucoup de définitions de la robustesse ont été données dans la littérature. On retiendra notamment une proposition de Kroon et al (2008), qui appellent robustesse d'un plan de transport sa capacité à "faire face à de relativement faibles perturbations en opérationnel" ². Il reste là encore à préciser certains termes en fonction du contexte. En effet, selon le pays, voire la région, et le type d'activité, il est possible que les définitions varient.

Au sein même de SNCF, il existe plusieurs définitions de la robustesse. Dans un référentiel interne (Référentiel Infrastructure de la Direction déléguée Production-Méthodes, édité en 2001), un graphique de circulation (ensemble des circulations ferroviaires) sera déclaré robuste si :

- un petit incident ³ n'entraîne pas d'effet "boule de neige" (à savoir pas d'aggravation sensible des retards des trains successifs par rapport au retard initial) ;
- les effets locaux de cet incident sont résorbés en une heure maximum après la fin de l'incident lui-même ;
- la circulation d'un train isolé affecté d'un retard quelconque par rapport à son sillon théorique ne génère pas de perturbation importante.

Par ailleurs, Chandesris (2005) définit la robustesse d'un plan de transport comme son *"aptitude à résister et à s'autoréguler en un temps limité, en présence d'une perturbation"*.

D'après ces premières définitions et suite à plusieurs rencontres avec des experts internes à SNCF, nous considérons que la robustesse d'un plan de transport est son aptitude à résister, à

2. "to deal as well as possible with relatively small disturbances in the real-time operations"

3. On entend par petit incident un incident générant un retard d'au plus 10 minutes au premier train touché, parfois moins en fonction des contraintes locales.

s'autoréguler en un temps limité, ou à être facilement réparable en présence d'une faible perturbation :

- résister : aucun impact sur le plan de transport ;
- s'autoréguler : l'impact sur le plan de transport est limité, le retard est absorbé rapidement sans créer un effet "boule de neige" ;
- facilement réparable : le plan de transport doit être modifié mais il est facilement réparable par l'application de procédures préconçues lors de l'étude ou de la conception.

Ces notions sont illustrées par des exemples simples à la section 4.1. De plus, certaines précisions sont à apporter selon le contexte de l'étude, ici les trains régionaux (TER). Nous considérons par exemple qu'un train est à l'heure s'il a moins de deux minutes de retard, et qu'une "faible perturbation" est une perturbation engendrant un retard d'au plus 15 minutes.

3.3.2 Caractérisation de la robustesse

Une fois la robustesse définie, il faut la caractériser. Dans ce but, nous avons déterminé des indicateurs de robustesse. À partir de tels indicateurs, il est possible de mesurer la robustesse d'un plan de transport et de construire des plans de transport robustes.

Nous distinguerons deux catégories d'indicateurs de robustesse : les indicateurs d'évaluation et les indicateurs de construction.

Les indicateurs d'évaluation permettent de mesurer la robustesse d'un plan de transport existant. En particulier, certains indicateurs ne peuvent être calculés qu'a posteriori. Par exemple, les retards ne sont connus qu'une fois que le plan de transport a été testé en réalité.

Les indicateurs de construction sont utilisés pour guider la construction de plans de transport robustes. En effet, il existe des règles dont le respect assure une certaine robustesse du plan de transport. Par exemple, en ajoutant quelques minutes de marge à la durée théorique des trajets, le plan de transport résistera plus aux retards. On peut alors utiliser ces règles pour définir des indicateurs de construction : l'optimisation de ces indicateurs lors de la conception du plan de transport en améliorera la robustesse. Les indicateurs de construction peuvent également être utilisés pour l'évaluation.

Dans la suite, nous présentons des indicateurs de robustesse d'un plan de transport répartis dans ces deux catégories.

Indicateurs d'évaluation

Dans la littérature, nous avons trouvé un certain nombre d'indicateurs pertinents pour l'évaluation de la robustesse d'un plan de transport.

Dans le cadre de la planification des ressources, Nielsen et al. (2006) proposent plusieurs indicateurs de robustesse autour du retard : le *retard moyen* ou le retard cumulé. Il s'agit de la moyenne ou de la somme des retards à l'arrivée de tous les trains du plan de transport.

Afin d'évaluer la robustesse d'un plan de transport préétabli, ils ont donc cherché à calculer ces indicateurs par une méthode de simulation. Une perturbation sur un train est modélisée par un nombre généré aléatoirement ajouté au temps de trajet nominal. Nielsen et al. (2006) définissent des scénarios en générant de telles perturbations pour chaque train. Le retard cumulé ou le retard moyen sont alors évalués pour chaque scénario.

Hofman et Madsen (2005) utilisent un indicateur lié à la régularité pour mesurer la robustesse d'un plan de transport. Cependant, partant du constat que la suppression de trains augmente la régularité, ils complètent cet indicateur par des indicateurs de fiabilité. Les indicateurs qu'ils présentent sont définis par la suite.

La *régularité* est le taux de trains partis à l'heure prévue. Il est calculé comme suit :

$$1 - \frac{\text{Nombre départs en retard}}{\text{Nombre de départs total}}$$

Cet indicateur est aussi utilisé dans le cadre d'une stratégie de réparation : une régularité en temps réel inférieure à 95% implique une re planification au niveau opérationnel.

La *fiabilité* est le taux de trains réellement partis conformément aux prévisions. Il est calculé de la façon suivante :

$$\frac{\text{Nombre de départs réels}}{\text{Nombre de départs prévus}}$$

Cet indicateur considère tous les trains avec la même importance. Il serait possible de pénaliser plus fortement certains trains prioritaires selon leur type, leur temps de trajet, etc. Enfin, le *nombre de trains affectés par des perturbations* est aussi pris en compte dans la robustesse. Il s'agit du nombre de trains remplacés, "détournés" (changement d'itinéraire ou d'horaires) ou annulés. Hofman et Madsen (2005) utilisent ces indicateurs dans le cadre de la planification robuste des ressources matérielles. Ils utilisent ensuite l'outil de simulation Arena de DBS S-tog (compagnie de métro danoise à Copenhague) pour évaluer la robustesse des différents roulements en calculant la valeur attendue des indicateurs présentés.

À SNCF, Chandesris (2005) a travaillé à la mise en place d'un outil de simulation du système ferroviaire afin d'en tester la robustesse. Elle définit alors un indicateur : le *temps de retour à la normale*, qui correspond au temps de rétablissement du plan de transport par rapport au plan d'origine suite à un incident. Cet indicateur est mesuré en prenant l'intervalle de temps entre le premier train perturbé (retard source) et le dernier train dont le retard est la conséquence du retard source. Un train est en retard s'il arrive avec un retard supérieur à un seuil préalablement défini. Ce nouvel indicateur permet d'évaluer la capacité d'un plan de transport à s'autoréguler.

Les différents indicateurs présentés jusqu'alors correspondent à une vision "système". Ils ne permettent pas d'évaluer la qualité de service telle qu'elle peut être perçue par les voyageurs. Takeuchi et al. (2007) s'intéressent à une vision "passagers" de la planification des horaires. Les auteurs définissent l'*inconfort des passagers* ou le temps perdu ("disutility") en calculant un indice en fonction des paramètres suivants :

- Temps d'attente total T_W ;
- Temps d'embarquement T_R ;
- Temps de parcours t_k sur une section j (entre deux gares) ;
- Taux de surpeuplement C_k sur une section k ;
- Nombre de correspondances N .

L'indice d'inconfort u défini par Takeuchi et al. (2007) est alors calculé en fonction de ces paramètres :

$$u = \alpha \sum_k \{t_k (\frac{C_k}{100})^\beta\} + \gamma N + \delta T_W + T_R$$

On note que différents temps d'attente sont pris en compte. On cherche ici à calculer un inconfort ressenti. Or, une attente sur le quai paraît inutile, tandis que le temps d'embarquement est

nécessaire : le passager est actif et ressent moins l'attente. Par ailleurs, il est considéré qu'une fois embarqué, un passager vit mieux un trajet assis dans une rame dont le taux d'occupation avoisine les 60% qu'en cas de forte affluence.

Pour chaque passager i , on obtient alors un indice d'inconfort prévu u_i . En présence d'un incident j , l'inconfort attendu pour ce même passager est noté $u_i^{(j)}$. L'indicateur de robustesse proposé est alors la différence entre les valeurs de l'indice attendue et réelle.

On illustre cette idée sur un exemple simple fictif schématisé par la Figure 3.1. Dans cet exemple, on confond temps d'attente et temps d'embarquement ($\delta = 1$).

Soit un passager entrant dans le système en gare A , où il prend un train après 10 minutes d'attente. Après un premier trajet T_1 de 45 minutes vers la gare B et un temps de correspondance de 10 minutes également, il effectue un second trajet T_2 de 53 minutes vers la gare C , sa destination. Il sort alors du système. Dans tous les cas, le taux de surpeuplement est de 70%.

L'indice d'inconfort calculé pour ce passager est alors le suivant (pour $\alpha = \beta = 1$ et $\gamma = 100$) :

$$u = 45 \times \left(\frac{70}{100}\right) + 53 \times \left(\frac{70}{100}\right) + 100 \times 1 + 10 + 10 = 188,6$$

Or cet itinéraire, en vert sur la Figure 3.1, est l'itinéraire prévu par le passager. On considère maintenant un incident j , engendrant un retard de 12 minutes sur le trajet T_1 .

On considère l'apparition d'un incident j lors du trajet T_1 , provoquant un retard de 12 minutes à l'arrivée. Le passager manque la correspondance et doit prendre la suivante. Il effectue le trajet T_3 , omnibus, qui dure une heure et 20 minutes. Comme beaucoup de passagers ont subi le même imprévu, le taux de surpeuplement est de 110%. Le nouvel indice d'inconfort est alors le suivant :

$$u^{(j)} = 57 \times \left(\frac{70}{100}\right) + 80 \times \left(\frac{70}{100}\right) + 100 \times 1 + 10 + 8 = 213,9$$

Dans le cas où il n'y aurait pas de trajet T_3 , le passager pourrait avoir à effectuer deux trajets pour atteindre sa destination et augmenter ainsi le nombre de correspondances à 2, détériorant significativement l'indice d'inconfort (+ 100 par rapport au précédent).

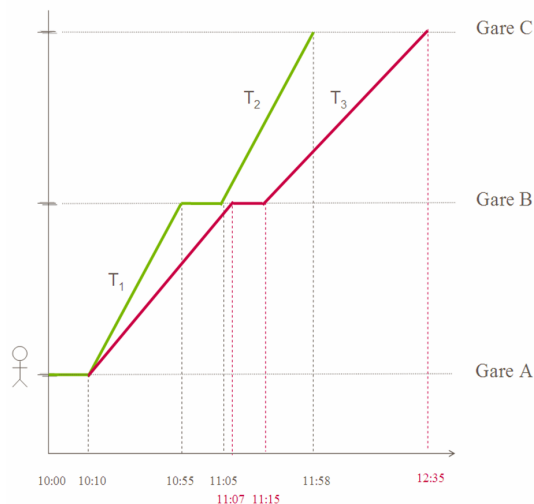


FIGURE 3.1 – Robustesse orientée passager : différence entre inconforts prévu et réalisé.

En conception, les valeurs réalisées ne sont évidemment pas connues. Takeuchi et al. (2007) calculent l'indice en partant d'une représentation probabiliste des incidents et en utilisant la méthode de Monte-Carlo. L'indicateur de robustesse est la valeur attendue de l'augmentation totale

de l'inconfort des passagers : pour chaque scénario d'incident, on somme l'augmentation de l'inconfort pour chaque passager. Puis on fait une moyenne sur l'ensemble des scénarios. On obtient la formule suivante :

$$\sum_j \left\{ \frac{1}{n^{(j)}} \sum_{i=1}^{n^{(j)}} \max\{u_i^{(j)} - u_i, 0\} \right\} \quad \text{Dégradation de la valeur prévue de l'indice d'inconfort (3.10)}$$

Les indicateurs d'évaluation de la robustesse d'un plan de transport présentés dans notre état de l'art sont recensés dans la Table 3.1.

<i>Indicateurs d'évaluation</i>	Retard moyen et retard cumulé	Régularité	Fiabilité	Nombre de trains affectés par des perturbations	Temps de retour à la normale	Inconfort des passagers
Principe	Moyenne ou somme des retards de tous les trains d'un système	Taux de trains arrivés à l'heure	Taux de trains réelement partis conformément aux prévisions	Nombre de trains remplacés, détournés, ou annulés	Temps de rétablissement du plan de transport par rapport au plan d'origine après un incident	Dégradation de la valeur prévue de l'indice d'inconfort : différence entre les valeurs prévues et réelle de l'indice d'inconfort des passagers
Contexte, application	Évaluation d'une grille horaire (par simulation)	Planification des ressources matérielles (Evaluation par simulation)	Planification des ressources matérielles (Evaluation par simulation)	Planification des ressources matérielles (Évaluation par simulation)	Évaluation d'une grille horaire (par simulation)	Évaluation d'une grille horaire (par simulation)
Références	Nielsen et al. (2006)	Hofman et Madsen (2005)	Hofman et Madsen (2005)	Hofman et Madsen (2005)	Chandesris (2005)	Takeuchi et al. (2007)

TABLE 3.1 – Indicateurs d'évaluation de la robustesse d'un plan de transport

Indicateurs de construction

L'optimisation de certains indicateurs lors de l'élaboration d'un plan de transport permet de gagner en robustesse.

Dans le cadre de la planification des horaires, Hofman et Madsen (2005) proposent l'ajout de *marges* temporelles (buffer times) au planning, afin d'absorber de faibles retards. Pour leurs expérimentations, ils ajoutent des retards aux trains selon des probabilités de retard correspondant à une régularité réaliste. Les résultats sont illustrés par le graphique en Figure 3.2. Ils constatent une régularité à 70% pour des marges de deux minutes, alors que des marges de cinq minutes permettent d'observer une régularité de 90%. Il est intéressant de noter qu'au delà de 6 minutes de marge, la régularité est constante. Si l'ajout de quelques minutes de marge permet de gagner en robustesse (mesurée par la régularité), il est inutile d'ajouter des marges trop importantes.

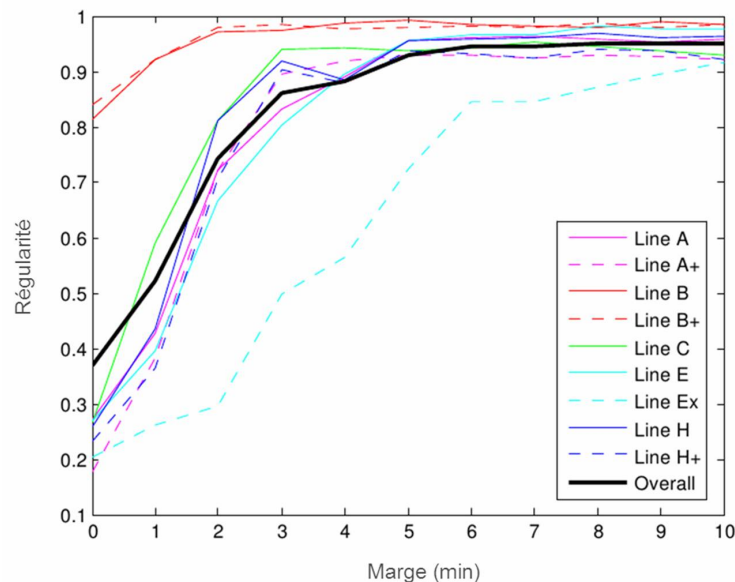


FIGURE 3.2 – Impact de l'ajout de marges (buffer times) sur la régularité.

On a vu précédemment que Nielsen et al. (2006) utilisent des indicateurs liés au retard pour évaluer la robustesse de roulements d'engins. Dans le but de construire des roulements robustes, ils proposent aussi d'autres indicateurs :

- l'augmentation des *temps de retournement*,
- la limitation du nombre de *recompositions*.

Le temps de retournement correspond au temps de stationnement en gare d'un engin entre deux trajets. Plus ce temps est long, plus les petits retards pris sur le trajet pourront être absorbés facilement. Le principe de cet indicateur est illustré en Figure 3.3.

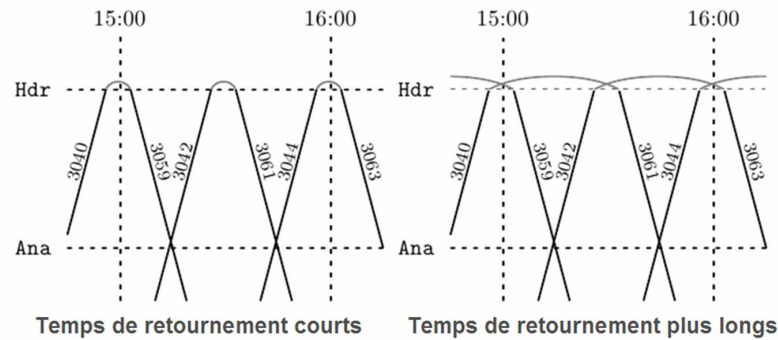


FIGURE 3.3 – Augmentation des temps de retournement.

Cependant, augmenter ces temps de retournement demande l'utilisation d'engins supplémentaires. C'est ce qu'on appelle le coût de la robustesse. Nielsen et al. (2006) proposent un exemple simulé dans lequel l'ajout de deux engins permet de réduire de 60% le retard propagé. Par ailleurs, on note que des temps de retournement trop longs peuvent saturer la capacité des gares, ce sujet n'est pas abordé dans l'article.

Lorsque plusieurs engins sont affectés à un même train, ils forment une unité multiple (UM). Les engins doivent être couplés avant le départ, puis découplés en fin de trajet. Ces opérations de composition et de décomposition représentent un risque supplémentaire de retard. Nielsen et al. (2006) cherchent donc à les minimiser, en gardant par exemple une même UM pour plusieurs trains successifs. Suite à des expérimentations, ils ont cherché à calculer le coût de cette robustesse, exprimé comme un surplus d'engins, et le gain en robustesse mesuré par le retard cumulé.

Dans leurs expérimentations, le nombre d'engins optimal est 25 et le nombre d'opérations de recomposition peut aller jusqu'à 60. Lorsqu'aucune recomposition n'est autorisée, trois engins supplémentaires (+12%) sont nécessaires et le retard cumulé est réduit de 20%. Cependant, le gain sur le retard cumulé reste quasiment le même en autorisant cinq recompositions, ce qui permet de rester à coûts constants.

Pour le même problème de planification des engins, Vromans et al. (2003) proposent d'homogénéiser les *intervalles de temps entre les trains* ("headways" illustrés en Figure 3.4) et d'utiliser les engins d'un même type pour des trajets identiques.

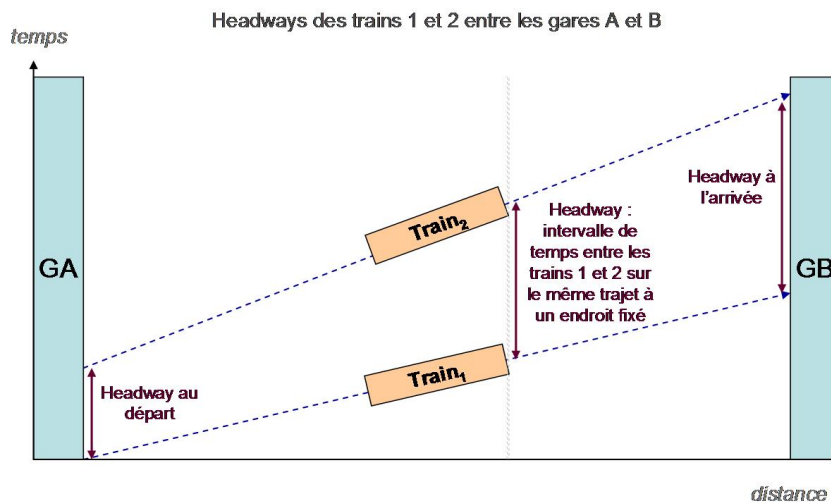


FIGURE 3.4 – Définition des intervalles de temps entre deux trains ou headways.

Ils définissent dans ce but deux indices appelés SSHR ("*Sum of Shortest Headway Reciprocals*") et SAHR ("*Sum of Arrival Headway Reciprocals*"). Le SSHR consiste en la somme des inverses des intervalles de temps minimum entre deux trains qui se suivent sur le même trajet. Le SAHR suit la même idée en considérant les intervalles de temps à l'arrivée uniquement. On note h_i^- l'intervalle de temps minimum entre le train i et son successeur et h_i^A l'intervalle à l'arrivée.

Les deux indicateurs se calculent alors de la façon suivante :

$$\begin{cases} SSHR = \sum_{i=1}^n \frac{1}{h_i^-} \\ SSAR = \sum_{i=1}^n \frac{1}{h_i^A} \end{cases}$$

Sur les exemples (a) et (b) de la Figure 3.5, l'écart entre deux trains successifs est constant, donc les valeurs des SSHR et des SAHR sont les mêmes, on obtient :

$$\begin{cases} SSHR(a) = \frac{1}{15} + \frac{1}{15} + \frac{1}{15} + \frac{1}{15} = 0,27 \\ SSHR(b) = \frac{1}{5} + \frac{1}{25} + \frac{1}{5} + \frac{1}{25} = 0,48 \end{cases}$$

Plus le SSHR est petit, plus les intervalles entre les trains sont homogènes, donc plus le plan de transport est robuste.

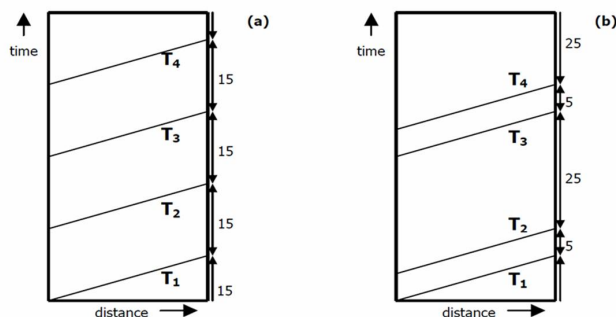


FIGURE 3.5 – Calcul des indicateurs SSHR et SAHR sur deux exemples.

À SNCF, Vianey (2007) a développé un modèle permettant de minimiser le *retard propagé* lors de la construction des roulements d'engins dans PRESTO. Dans ce but, le PLNE P_{PRESTO} est modifié : des retards indépendants sont distribués sur les tâches, des variables de retard, des variables de décision supplémentaires et des contraintes sont ajoutées. Des expérimentations ont été menées sur un cas réel issu de données du RER D : 150 tâches sur une journée puis 297 tâches sur deux jours à couvrir par au plus 50 engins. Par absence de données, les retards indépendants ont été générés arbitrairement. On observe alors une réduction de 50% du retard propagé sans surcoût d'engins par rapport à PRESTO. Ensuite, l'ajout d'un engin permet à nouveau un gain de 50% par rapport au résultat précédent. Cependant, le modèle obtenu est lourd et rend la résolution complexe : le temps de résolution sans surcoût d'engins est inférieur à 20 minutes pour une journée mais dépasse les 15 heures pour deux jours. De plus, de nombreux sillons à vide sont ajoutés. Enfin, ce type de méthode requiert une distribution de retards réaliste.

Enfin, Nielsen et al. (2006) proposent aussi des indicateurs pour la planification robuste des agents de conduite (ADC) et des agents du service commercial des trains (ASCT) :

- ajout de *marges aux transferts des agents* : augmentation de la durée des transferts des agents entre deux trains ;
- limitation du nombre de *recompositions d'équipes* : à chaque train est affectée une équipe de plusieurs agents, l'idée est de conserver au maximum la composition des équipes lors des transferts ;
- limitation des *combinaisons agents-engins* : il s'agit de conserver les mêmes affectations d'agents aux engins.

D'une part, en cas d'incident, les marges permettent d'absorber le retard. D'autre part, si deux équipes sont modifiées pour former deux équipes différentes sur les trains suivants, un retard d'une équipe a des répercussions sur tous les trains recevant un des agents de l'ancienne équipe. Mais si les équipes ne sont pas modifiées, seul un train partira en retard. Le principe est le même concernant les affectations d'agents aux engins.

D'après les expérimentations menées, l'ajout de cinq minutes de marge à chaque transfert permet une réduction de près de 20% du retard cumulé. La limitation des changements de composition des équipes permet une diminution de 4% du retard cumulé. Enfin, la conservation des mêmes affectations d'agents aux engins réduit le retard cumulé de 29% environ.

Les indicateurs de construction sont synthétisés dans le tableau 3.2.

<i>Indicateurs de construction</i>	Marges	Temps de retournement et intervalles entre les trains	Recompositions	Retard propagé	Transferts des agents	Recompositions agents-agents et combinaisons agents-engins
Principe	Ajout de marges de quelques minutes aux durées des trajets	Augmentation du temps de stationnement en gare et des intervalles entre les trains qui se suivent	Limitation du nombre d'opérations de recombinaison	Minimisation du retard propagé d'après une estimation des retards indépendants	Ajout de marges aux transferts des agents	Conservation des équipes d'agents et des affectations agents-engins
Contexte, application	Planification des horaires	Planification des engins	Planification des engins	Planification des engins	Planification des agents	Planification des agents
Évaluation de la robustesse	Régularité	Retard cumulé	Retard cumulé	Retard propagé	Retard cumulé	Retard cumulé
Résultats	70% de régularité pour des marges de 2 minutes, 90% de régularité pour des marges de 5 minutes	60% de réduction du retard cumulé, surcoût de 2 engins	20% de réduction du retard cumulé, surcoût de 0 à 3 engins	50% de réduction du retard propagé sans surcoût, 75% de réduction avec un surcoût d'un engin	20% de réduction du retard cumulé	29% de réduction du retard cumulé
Référence	Hofman et Madsen (2005)	Nielsen et al. (2006), Vromans et al. (2003)	Nielsen et al. (2006)	Rapport technique interne SNCF	Nielsen et al. (2006)	Nielsen et al. (2006)

TABLE 3.2 – Indicateurs de construction de plans de transport robustes

Chapitre 4

La robustesse appliquée à notre problème

On a vu à la section 1.3 qu'une solution déterministe peut être très fortement modifiée en cas de perturbation. Nous cherchons donc à construire des roulements d'engins robustes, afin de limiter l'impact des aléas sur le plan de transport. Dans ce but, nous avons d'abord cherché à définir et à caractériser la robustesse dans le contexte d'étude, puis nous avons envisagé plusieurs approches pour la planification de roulements d'engins robustes.

4.1 Notre définition de la robustesse

La robustesse est une notion vague qui doit être définie précisément selon le contexte. Nous avons donc cherché à définir la robustesse dans le contexte étudié :

- Transport ferroviaire
 - Transport express régional (TER) français
 - Planification des ressources matérielles (engins)

Nous proposons la définition suivante :

La robustesse d'un plan de transport est son aptitude à résister et à s'autoréguler en un temps limité ou à être facilement réparable en présence d'une perturbation.

Résister L'apparition d'une perturbation n'a pas d'impact sur le reste de la solution.

Par exemple, en Figure 4.1, on considère deux tâches T_1 et T_2 enchaînées par un même engin. On schématise en vert la grille horaire prévisionnelle et en rose la grille horaire réalisée. La tâche T_1 a 7 minutes de retard à l'arrivée, mais le temps de retournement est suffisamment long pour absorber ce retard et la tâche T_2 n'a aucun retard au départ. L'incident en tâche T_1 n'a donc aucun impact sur les tâches suivantes.

S'autoréguler Suite à l'apparition d'une perturbation, la solution est rétablie en un temps limité, un petit incident n'entraîne pas d'effet "boule de neige".

Soient trois tâches T_1, T_2 et T_3 enchaînées par un même engin, illustrées en Figure 4.2. Suite à l'apparition d'un incident, on observe un retard de 8 minutes à l'arrivée de la tâche T_1 , soit 2 minutes seulement avant le départ de la tâche T_2 . Or on considère un temps de crochet de 9 minutes, l'engin doit donc stationner 7 minutes supplémentaires en gare avant le second départ. Ce retard de 7 minutes sur la tâche T_2 peut cependant être absorbé entre les tâches T_2 et T_3 , grâce au temps de retournement de 23 minutes. Au départ de la tâche T_3 , le plan de transport initial est rétabli, l'impact de l'incident apparu est limité.

Être facilement réparable Suite à l'apparition d'une perturbation, la solution peut être rétablie par l'application d'une procédure de réparation prédéfinie en conception.

En Figure 4.3, trois tâches T_1, T_2 et T_3 sont à nouveau enchaînées par un même engin. On estime que le retard de 53 minutes sur la tâche T_1 engendrerait trop de perturbations sur le plan de transport. On engage donc un engin de réserve pour remplacer le premier engin dès la tâche T_2 . Le plan de transport est modifié mais il n'y a aucun impact sur la qualité de service. L'engagement d'un engin de réserve en gare B est une procédure qui doit avoir été prévue lors de la planification en cas d'incident.

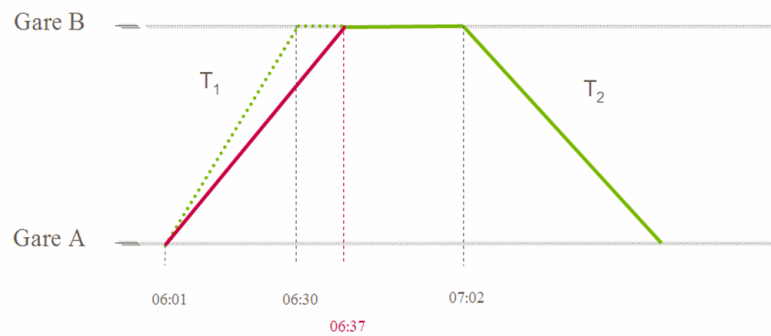


FIGURE 4.1 – Exemple de résistance à l'apparition d'une perturbation.

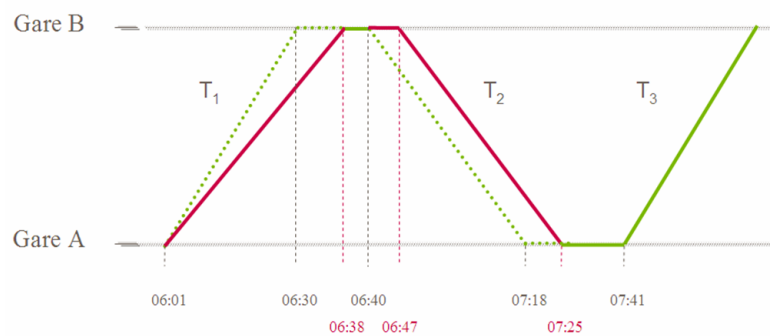


FIGURE 4.2 – Exemple d'autorégulation face à l'apparition d'une perturbation.

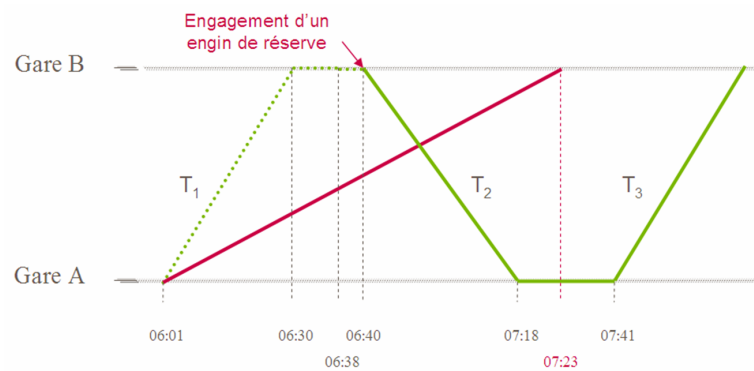


FIGURE 4.3 – Exemple de réparation face à l’apparition d’une perturbation.

Dans le cadre de notre étude, nous cherchons à construire des plans de transport qui résistent ou qui s’autorégulent en présence d’une perturbation. La réparabilité d’un plan de transport n’est pas abordée.

4.2 Approche structurelle

Lors de l’analyse bibliographique, on a relevé un certain nombre de méthodes utilisant des règles permettant de garantir un certain seuil de robustesse en agissant directement sur la structure de la solution (cf. section 3.3.2). Pour intégrer de la robustesse dans le processus de planification des roulements d’engins, on pourrait donc définir des critères de robustesse propres à notre contexte d’étude et agir sur la structure de la solution pour qu’elle respecte au mieux ces critères.

On a établi à la section 1.3.3 que le problème de planification robuste pouvait être décomposé en deux parties : la couverture maximale de la demande avec optimisation des coûts de production (Phase 1 de PRESTO), puis la construction de roulements robustes respectant la solution de flots. On cherche donc à prendre en compte des critères de robustesse lors de cette seconde partie.

Dans PRESTO, la construction de roulements d’engins est faite par les Phases 2, 3 et 4. Il est alors possible d’intégrer la robustesse lors de l’une ou plusieurs de ces phases, ou de remplacer les trois phases par une méthode plus globale, diminuant ainsi le risque de sous-optimalité.

Nous cherchons alors à identifier et à formaliser des critères de robustesse adaptés au contexte et au problème étudiés, permettant de construire des solutions robustes selon la définition donnée à la section 4.1.

Nous séparons ensuite les critères sélectionnés en deux catégories : les critères de construction sont intégrés à la méthode de planification afin de construire des roulements robustes, tandis que les critères d’évaluation seront uniquement utilisés pour l’évaluation a posteriori de la robustesse de roulements d’engins existants.

Les critères de robustesse de construction choisis sont les suivants :

- temps de retournement :
 - maximisation du temps de retournement minimum (t_{min}) ;
 - homogénéisation des temps de retournement ($invret$) ;

- pénalisation des temps de retournement intermédiaires (*tretInt*);
- recompositions : minimisation du nombre de recompositions (*recomp*).

Les critères de robustesse d'évaluation choisis sont les suivants :

- charge de travail : homogénéisation de la durée de circulation des engins (*chge*);
- retards :
 - maximisation de la régularité (*reg*);
 - minimisation du retard propagé moyen ou cumulé (*retprop*);
 - minimisation du retard en fin d'horizon temporel (*retHT*);
- situation perturbée : minimisation du temps de retour à la normale (*TRN*).

Il convient de préciser que nous définissons ces critères de robustesse pour un nombre d'engins fixé.

En effet, l'homogénéisation de la charge de travail ou des temps de retournement n'a de sens que pour une charge par engin moyenne ou un temps de retournement moyen fixes ou peu variables.

De plus, si on s'autorisait à augmenter le nombre d'engins, la "maximisation" du temps de retournement minimum tendrait à ne faire qu'une tâche par ligne de roulement, or il s'agit davantage d'éviter des temps de retournement trop courts lorsque c'est possible.

Nous détaillons ensuite les critères de robustesse listés.

4.2.1 Critères de construction

Temps de retournement

Soit un engin k enchaînant deux tâches T_i et T_j . Le temps retournement entre ces deux tâches est défini par $t_{ij} = D_j^{deb} - D_i^{fin}$.

Les temps de retournement sont des acteurs essentiels de la robustesse d'un plan de transport. En effet, plus ils sont courts, plus le risque de propager du retard augmente. À l'inverse, si des engins arrivant en gare ne sont pas réutilisés avant un temps plus long, un retard sur ces engins pourra être plus facilement absorbé par ce temps supplémentaire et aura beaucoup moins d'impact sur le reste du plan de transport. Pour autant, des temps de retournement trop longs peuvent saturer la capacité des gares.

– Maximisation du temps de retournement minimum : *tmin*

Afin de rendre un plan de transport robuste, on va donc chercher à maximiser le temps de retournement minimum du roulement. On définit :

$$tmin(rlt) = \min_{(T_i, T_j) \in rlt} (t_{ij})$$

Cependant, ce n'est pas suffisant. S'il existe un temps de retournement très court inévitable, alors ce critère ne poussera pas à améliorer d'autres temps de retournement courts.

– Homogénéisation des temps de retournement : *invret*

On cherche donc aussi à uniformiser les temps de retournement dans le roulement. Dans ce

but, nous cherchons à minimiser la somme des inverses de tous les temps de retournement du roulement. On rappelle qu'il y a un temps de retournement minimum obligatoire $tc(G)$, appelé temps de crochet, à respecter en chaque gare G . Un temps de retournement ne peut donc pas être nul. On note $invret(rlt)$ la somme des inverses des temps de retournement du roulement rlt , soit :

$$invret(rlt) = \sum_{(T_i, T_j) \in rlt} \frac{1}{t_{ij}}$$

Soit un roulement de deux lignes, illustrées par la Figure 4.4. Le temps de retournement minimum du roulement est de 10 minutes. Or les deux dernières tâches (de B vers C puis de C vers B) doivent être enchaînées par un même engin, il n'existe pas de solution admissible permettant de couvrir ces deux tâches par des engins différents. Le temps de retournement entre ces tâches étant de 10 minutes, le critère $tmin$ ne pourra pas être amélioré. De ce fait, la solution présentée est optimale pour ce critère.

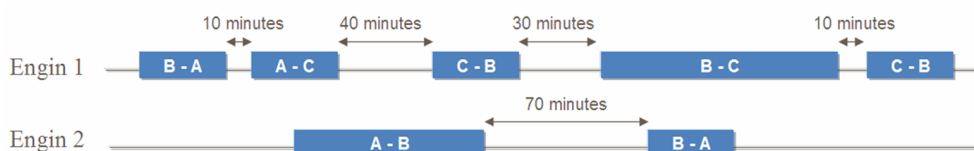


FIGURE 4.4 – Temps de retournement non homogènes malgré un temps minimum optimisé.

Or d'autres temps de retournement sont courts (10 minutes entre les deux premières tâches de la première ligne) ou beaucoup plus longs (70 minutes sur la seconde ligne). La solution pourrait être améliorée en homogénéisant les temps de retournement. Minimiser la somme des inverses des temps de retournement d'un roulement ($invret$) revient à les homogénéiser. En appliquant le critère de minimisation de $invret$, on trouverait la solution de la Figure 4.5. En effet, les valeurs de $invret$ sur ces deux solutions sont calculées comme suit :

- $invret(\text{fig 4.4}) = \frac{1}{10} + \frac{1}{40} + \frac{1}{30} + \frac{1}{10} + \frac{1}{70} = 0,2726$
- $invret(\text{fig 4.5}) = \frac{1}{40} + \frac{1}{55} + \frac{1}{30} + \frac{1}{45} + \frac{1}{10} = 0,1987$

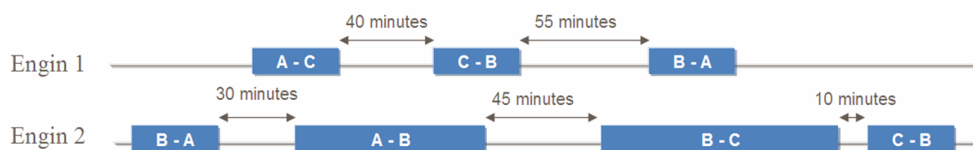


FIGURE 4.5 – Résultat attendu après homogénéisation des temps de retournement.

– Pénalisation des temps de retournement intermédiaires : $tretInt$

Les temps de retournement doivent être impérativement supérieurs aux temps de crochet. La structure du graphe de PRESTO présenté à la section 1.2.1 permet de garantir le respect de cette contrainte.

Par ailleurs, les temps de retournement supérieurs à un certain seuil, environ 20 minutes en général pour TER, sont déconseillés afin d'éviter la saturation de la capacité dans les gares. Un engin ne peut donc pas stationner en gare plus longtemps.

Cependant, si le temps de retournement est suffisamment long, au-delà de 40 minutes pour

TER, l'engin peut être envoyé au remisage, puis revenir en gare avant le prochain départ. Les temps de retournement intermédiaires, c'est-à-dire à la fois trop longs pour permettre à un engin de rester en gare et trop courts pour pouvoir effectuer l'opération de remisage, doivent être évités. On note :

$$tretInt(rlt) = \sum_{\substack{(T_i, T_j) \in rlt \\ t_{ij} \in [20, 40]}} 1$$

Recompositions : *recomp*

Soient deux tâches T_i et T_j couvertes par deux engins ($flot_p(T_i) = flot_p(T_j) = 2$). Une opération de couplage en début de chaque tâche et une opération de découplage en fin de chaque tâche sont nécessaires pour coupler ou découpler les UM. Ces opérations appelées recompositions (cf. section 4.2) augmentent le risque de retard. Cependant, si l'UM est la même sur les deux tâches, il est inutile de découpler puis de recoupler les engins entre les tâches. On évite donc deux recompositions. On cherche à éviter autant de recompositions que possible dans le roulement, afin de diminuer le risque de retard sur ces opérations.

Dans le cas où l'UM sur les tâches est formée de trois engins ($flot_p(T_i) = flot_p(T_j) = 3$), on compte deux recompositions pour les connecter puis deux recompositions pour les découpler en fin de tâche.

En généralisant, une UM formée de n engins nécessite $n - 1$ recompositions pour coupler les engins entre eux, puis $n - 1$ recompositions pour les découpler, soit $2.(n - 1)$ recompositions.

En revanche, si p engins parmi les n engins de l'UM couvrant T_i sont réutilisés pour couvrir T_j , il n'est pas nécessaire de découpler ces p engins. On économise $2.(p - 1)$ recompositions, seules $2.(n - p)$ recompositions sont nécessaires entre les deux tâches.

Si les deux tâches sont couvertes par la même UM de n engins, aucune recomposition ne sera nécessaire entre les deux tâches puisque l'UM peut être conservée.

Soient trois engins 1, 2 et 3 de même type couvrant quatre tâches T_1, T_2, T_3 et T_4 . Dans l'exemple de la Figure 4.6, les engins 1 et 2 doivent être couplés pour couvrir la tâche T_2 , puis découplés. Ensuite, les engins 1 et 3 sont couplés pour couvrir la tâche T_3 puis découplés. Il y a donc quatre recompositions en tout.

Or, en fin de tâche T_2 , on pourrait garder l'UM composée des engins 1 et 2 afin de couvrir la tâche T_3 . On évite ainsi deux opérations, comme illustré par la Figure 4.7.

On note que dans le cas de la Figure 4.7, la minimisation du nombre de recompositions a entraîné une dégradation de l'homogénéisation des temps de retournement.

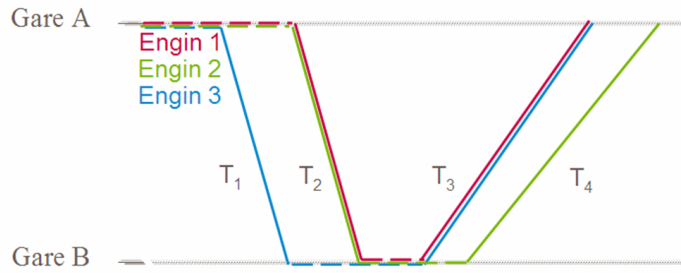


FIGURE 4.6 – Exemple de planification des engins engendrant quatre recompositions.

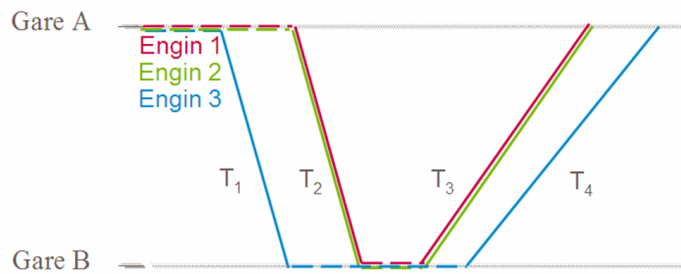


FIGURE 4.7 – Exemple de planification des engins engendrant deux recompositions.

4.2.2 Critères d'évaluation

Charge de travail : $chge$

On définit la charge de travail d'un engin comme la durée pendant laquelle il effectue des tâches sur un horizon temporel. Un roulement dont les engins ont tous une charge de travail équivalente est plus robuste qu'un roulement dont certains engins sont surchargés. On cherche donc à équilibrer la charge de travail entre les engins. Dans ce but, on s'inspire du critère d'homogénéisation des temps de retournement : on minimise la somme des inverses des charges de travail par engin (ligne de roulement).

$$chge(rlt) = \sum_{T_i \in rlt} D_i^{fin} - D_i^{deb}$$

Il est intéressant de noter que la durée d'une ligne de roulement est la somme des durées de ses tâches et de ses temps de retournement. En homogénéisant les temps de retournement dans le roulement, on homogénéise également les durées des intervalles entre les temps de retournement, c'est-à-dire les tâches. Les critères de répartition des temps de retournement et de répartition de la charge de travail ne sont pas identiques mais fortement corrélés : l'optimisation du premier tend à optimiser le second. On ne cherchera donc pas directement à homogénéiser la charge de travail, on l'évaluera cependant a posteriori.

Retards

- Maximisation de la régularité : reg

Une tâche est en retard si sa date d'arrivée réelle est supérieure à sa date d'arrivée théorique augmentée d'une marge de régularité prédéfinie (généralement 2 à 5 minutes). Par exemple, une tâche arrivant à 8 heures et 37 minutes au lieu de 8 heures et 35 minutes n'est pas en retard.

La régularité est le taux de tâches à l'heure à l'arrivée. Elle se calcule donc de la façon suivante :

$$reg = 1 - \frac{\text{Nombre d'arrivées en retard}}{\text{Nombre d'arrivées total}}$$

On a choisi de calculer la régularité à l'arrivée, on pourrait aussi mesurer la régularité au départ (taux de trains partant à l'heure, généralement appelée ponctualité).

– **Minimisation du retard propagé : *retprop***

Lorsqu'un incident se produit, il peut générer un premier retard qu'on appelle retard source ou retard primaire. Si un train arrive avec beaucoup de retard, le ou les trains suivants peuvent être eux aussi retardés. Ces nouveaux retards sont des retards propagés : ils ne sont pas directement dus à un incident, mais à la propagation d'un retard précédent.

Soit un engin k enchaînant deux tâches T_1 et T_2 . T_1 arrive à 00h30 en gare B , dont le temps de crochet est de 7 minutes, et repart 10 minutes après (cf. schéma de gauche de la Figure 4.8). On considère un incident engendrant un retard Ra_1 de 8 minutes à l'arrivée de T_1 . Le retard est partiellement absorbé par le temps de retournement entre T_1 et T_2 , mais il faut au minimum 7 minutes entre l'arrivée de T_1 et le départ de T_2 (temps de crochet). La tâche T_2 aura donc un retard départ, appelé retard propagé et noté Rp_2 , tel que :

$$Rp_2 = Ra_1 - (t_{1,2} - tc(B))$$

$$Rp_2 = 8 - (10 - 7) = 5$$

Le retard propagé au départ de la tâche T_2 est alors de 5 minutes, comme illustré par le second schéma de la Figure 4.8 (tâches initialement prévues en vert et réalisées en rose).

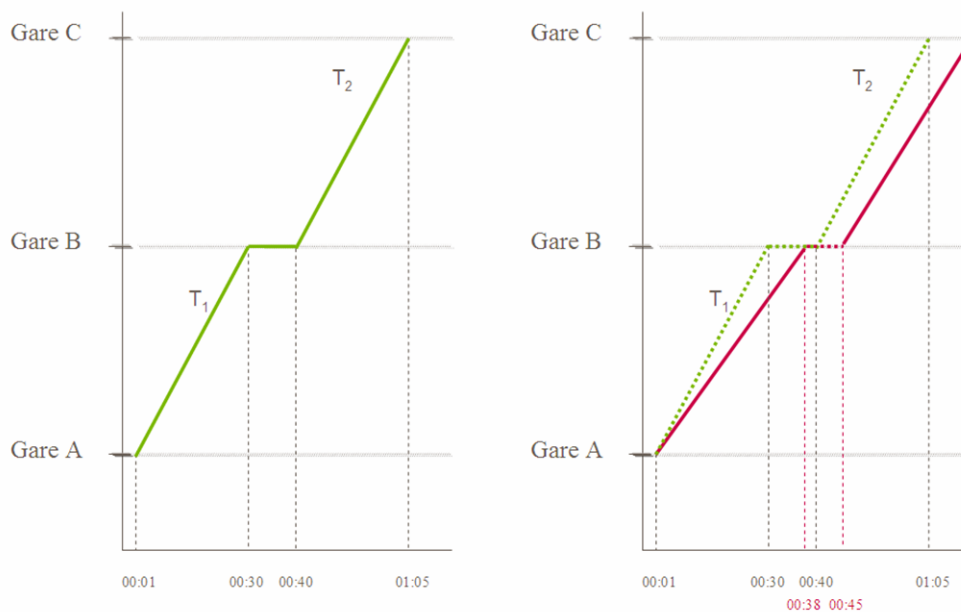


FIGURE 4.8 – Illustration du retard propagé sur un exemple simple.

Le critère *retprop* correspond alors à la somme des retards propagés (il pourrait correspondre à la moyenne) :

$$retprop(rlt) = \sum_{T_i \in rlt} Rp_i$$

– **Minimisation du retard en fin d’horizon temporel : *retHT***

Soit un engin ayant du retard à l’arrivée de sa dernière tâche. Si le retard ne peut pas être absorbé avant la fin de l’horizon temporel, il y a un retard résiduel. Ce retard en fin d’horizon temporel est à éviter afin de pouvoir recommencer un cycle sereinement. Le critère *retHT* est la somme des retards résiduels :

$$retHT(rlt) = \sum_{k \in N} \text{Retard résiduel}(k)$$

Situation perturbée : *TRN*

Lorsqu’un incident apparaît, on passe en situation perturbée et on cherche à rétablir une situation normale. L’intervalle de temps entre le début de l’incident et la fin de la situation perturbée est appelé temps de retour à la normale et doit être minimisé. Le critère *TRN* est la somme de ces temps de retour à la normale :

$$TRN(rlt) = \sum_{rs \in \text{retard source}} (\text{Fin de situation perturbée} - \text{Apparition de } rs)$$

Soit un engin enchaînant six tâches T_1 à T_6 (cf. Figure 4.9, schéma du haut). On considère un incident sur la tâche T_2 engendrant un certain retard (cf. schéma du bas). On est alors en situation perturbée. Le retard est propagé aux tâches T_3 et T_4 . En revanche, il est complètement absorbé entre les tâches T_4 et T_5 , permettant un départ de T_5 sans retard. On est alors à nouveau en situation normale. Le temps de retour à la normale est le temps qu’il a fallu pour rétablir une situation normale (situation rétablie à la date d_2) après l’apparition de la perturbation (à la date d_1), soit $d_2 - d_1$.

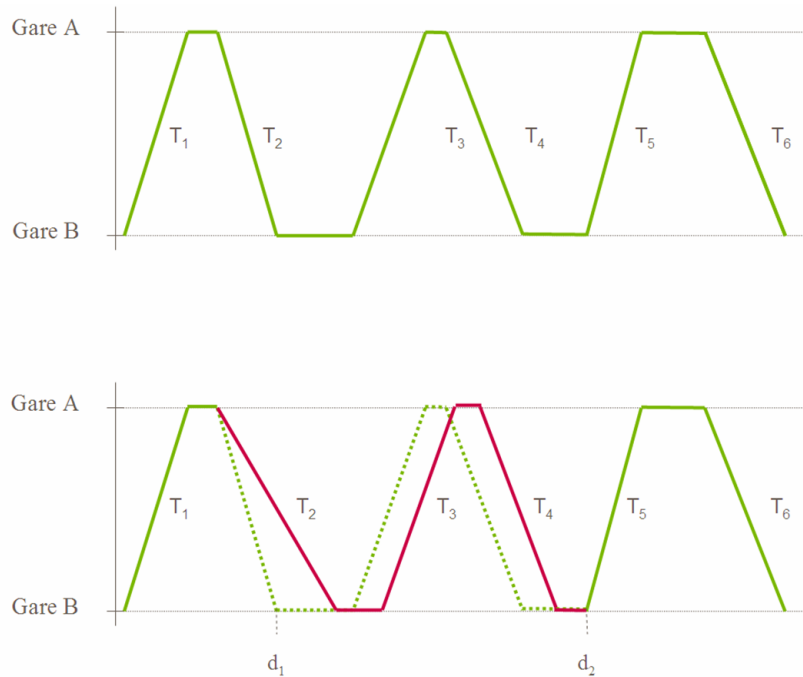


FIGURE 4.9 – Situation normale (en haut) et situation perturbée (en bas) avec temps de retour à la normale ($d_2 - d_1$) sur un exemple simple.

4.3 Approche par scénarios

Données incertaines

Dans notre problème, l'apparition d'incidents peut engendrer des retards sur les tâches. L'incertitude sur les données d'entrée concerne alors les horaires de départ et d'arrivée des tâches. Elle peut être modélisée sous forme de scénarios de retards : un scénario consiste en un ensemble de retards, chaque retard étant représenté par un nombre de minutes à ajouter à la date d'arrivée d'une tâche. On appelle scénario nominal le scénario sans retard, c'est-à-dire pour lequel les horaires des tâches n'ont pas été modifiés.

Définition de l'ensemble des scénarios à considérer

La pertinence d'une approche par scénarios dépend fortement des scénarios générés et nécessite donc un premier travail de détermination de l'ensemble des scénarios à considérer.

Pour notre problème, nous ne cherchons pas une solution absolument robuste, c'est-à-dire robuste à tous les scénarios qui peuvent se produire. En effet, l'ensemble des scénarios réalisables est trop conséquent et une telle solution, si elle existe, risque d'être trop coûteuse.

Nous cherchons donc une solution robuste à de petits incidents. Nous définissons un petit incident comme un incident ayant pour conséquence directe un retard inférieur à un certain seuil, par exemple 15 minutes. Au niveau du métier, on considère qu'au-delà d'un tel seuil, il faut modifier la solution.

La définition de l'ensemble des scénarios à prendre en compte est difficile. En effet, nous cherchons à produire des scénarios réalistes couvrant au mieux les configurations de retards possibles parmi ceux considérés. Dans ce but, nous exploitons une base de données recensant l'historique des retards sur tous les trains en France.

Cependant, la dépendance des données entre elles ne permet pas de générer des scénarios réalistes simplement. En effet, un retard de quelques minutes va avoir un impact sur le reste du plan de transport. Il est donc difficile de déterminer si un retard est indépendant ou dû à un précédent aléa. De plus, l'impact d'un incident sur un plan de transport va aussi dépendre de la structure du roulement : un retard apparu sur un roulement n'aurait peut-être pas été observé sur un autre roulement. Il faudrait donc considérer uniquement ce qui est exogène au roulement.

Nous choisissons d'utiliser une procédure de génération de retards simplifiée, détaillée à la section 9.2.

Prise en compte des scénarios dans les données

Les scénarios définis doivent être pris en compte dans les données. Nous avons alors identifié deux techniques différentes : soit on modifie directement les données incertaines, c'est-à-dire les horaires des tâches, soit on renseigne des données supplémentaires sur les tâches, par exemple sous forme d'un nombre de minutes de retard potentiel.

Dans le premier cas, les horaires sont modifiés et on peut résoudre le problème de façon déterministe en considérant ces nouvelles données. Cependant, la structure du graphe de PRESTO décrit à la section 1.2.1 est aussi modifiée, rendant parfois certains enchaînements impossibles.

Par exemple, en Figure 4.10, les tâches T_1 et T_2 peuvent être enchaînées par un même engin sur le graphique de gauche représentant les données nominales (sans retard). En revanche, si on considère un scénario qui affecte 10 minutes de retard à T_1 , il faut deux engins différents pour couvrir les deux tâches (graphique de droite). Ce phénomène peut avoir un impact sur la couverture des tâches ou sur le nombre d'engins utilisés. On peut donc dégrader les coûts par rapport à l'objectif nominal (sans retard), alors qu'on cherche à rester à coûts de production constants dans notre approche.

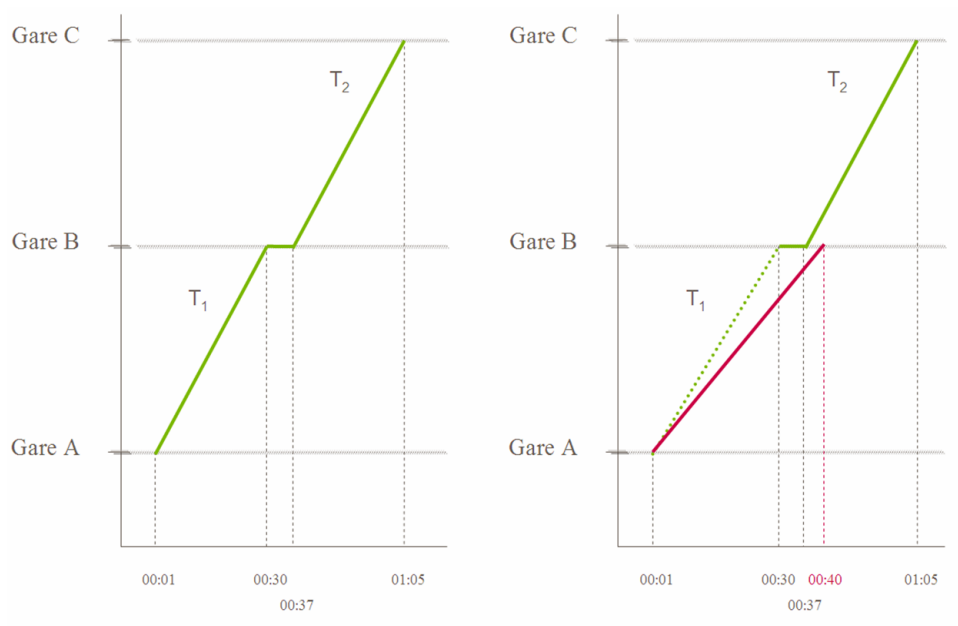


FIGURE 4.10 – Exemple de modification des données selon un scénario.

Dans le second cas, on peut affecter un retard aux différents trains sous la forme d'un poids correspondant à une durée (par exemple le nombre de minutes de retard) sans modifier directement les données (horaires des tâches). De cette façon, le graphe n'est pas modifié et la solution calculée par PRESTO est toujours admissible. Ces scénarios de retards donnent simplement des informations supplémentaires, afin par exemple de maximiser la régularité ou minimiser le retard propagé.

Résolution

Une approche utilisant un unique scénario pour lequel le problème est résolu de façon déterministe est pertinente si le retard est toujours du même ordre sur chaque tâche. Or ce n'est pas le cas du système ferroviaire. Pour quelques trains dits malades, on observe régulièrement les mêmes retards, mais il s'agit de cas isolés.

L'utilisation d'une approche par scénarios est plus intéressante si on peut générer plusieurs scénarios. Il faut alors déterminer ce qu'on attend d'une solution robuste par rapport à ces scénarios et choisir une méthode de résolution adaptée. Or une solution robuste à l'ensemble des scénarios n'est pas forcément une solution optimale pour l'un des scénarios. Il faut donc pouvoir mettre en œuvre une méthode de résolution prenant en compte plusieurs scénarios.

Difficultés et intérêt de l'approche

- Plusieurs difficultés apparaissent concernant la mise en œuvre d'une approche par scénarios :
- définition de l'ensemble des scénarios réalistes à considérer ;
 - génération d'un sous-ensemble de scénarios efficace pour couvrir au mieux l'ensemble des scénarios définis ;
 - prise en compte de plusieurs scénarios par une méthode de résolution adaptée.

Nous avons décidé de ne pas utiliser de scénarios de retards dans le cadre de la construction des roulements d’engins. En revanche, cette approche est retenue pour l’évaluation par simulation de solutions préalablement calculées : le principe et le fonctionnement de la simulation sont détaillés au Chapitre 9.

D’une part, la contrainte de génération de scénarios réalistes est moins critique. En effet, il s’agit de comparer des solutions entre elles afin d’en déduire la plus robuste. Si ces solutions sont éprouvées sur les mêmes scénarios, la comparaison est pertinente, quel que soit le degré de réalisme des scénarios.

D’autre part, on peut lancer autant de simulations que de scénarios à tester. La seule difficulté liée au nombre de scénarios est alors la présentation des résultats de façon lisible.

Il reste cependant à déterminer l’ensemble des scénarios à considérer (cf. Chapitre 9).

4.4 Discussion autour de quelques approches classiques non retenues

Les approches basées sur des intervalles d’incertitude ou des scénarios paraissent assez théoriques. Nous présentons ici quelques éléments de réflexion ayant orienté notre choix pour une méthode moins classique (cf. section 4.2).

4.4.1 Approche par intervalles

Le principe d’une approche par intervalles est de représenter chaque donnée incertaine d’un programme par l’intervalle borné de l’ensemble des valeurs qu’elle peut prendre. Bertsimas et Sim (2001) et Ben-Tal et Nemirovski (1998) appliquent ce principe à un programme linéaire dont les données incertaines sont les coefficients de la matrice des contraintes ou de la fonction objectif.

Dans PRESTO, une fois les données modélisées sous forme de graphe spatio-temporel, ce graphe est exploité pour formuler un PLNE (P_{PRESTO}). Il s’agirait donc de modifier P_{PRESTO} afin que la solution proposée après sa résolution soit robuste.

Les aléas considérés peuvent causer une modification des horaires des tâches. Or les sommets étant définis par des couples gare-date, c’est la structure même du graphe qui est modifiée lors de l’apparition d’aléas, ce qui complique le modèle : les données incertaines ne concernent pas des coefficients de la matrice ou de l’objectif, mais elles modifient la structure de certaines contraintes.

L’impact de l’apparition de retards sur la structure du graphe est illustré par la Figure 4.11. Le schéma de gauche représente une partie du graphe de modélisation des données nominales, tandis que le schéma de droite correspond au même graphe dans le cas d’un retard de dix minutes sur la tâche T_1 . Les arcs-connexions (cf. section 1.2.1) n’apparaissent pas sur les schémas afin de les simplifier. Concrètement, la modification de l’horaire d’arrivée de T_1 supprime la possibilité d’enchaîner T_1 et T_2 ou T_1 et T_3 .

Au niveau de P_{PRESTO} , certaines contraintes sont modifiées. C’est le cas des contraintes de conservation des flots garantissant le même nombre d’engins entrants et sortants pour chaque sommet.

On rappelle que la contrainte de conservation des flots à un sommet s s’écrit :

$$\sum_{a \in In(s)} x_a = \sum_{a \in Out(s)} x_a$$

avec x_a le nombre d'engins circulant sur l'arc a .

Dans le cas nominal, la contrainte de conservation des flots au sommet s_1 s'écrit

$$x_{a_1} + x_{T_1} = x_{a_2} + x_{T_2} + x_{T_3}.$$

Dans le cas perturbé, on ajoute dix minutes à l'horaire d'arrivée de T_1 (cf. schéma de droite sur la Figure 4.11). Alors les contraintes de conservation des flots en s_1 et en s_2 sont les suivantes :

- $x_{a_1} = x_{a_2} + x_{T_2} + x_{T_3}$ (en s_1) ;
- $x_{a_2} + x_{T_1} = x_{a_3}$ (en s_2).

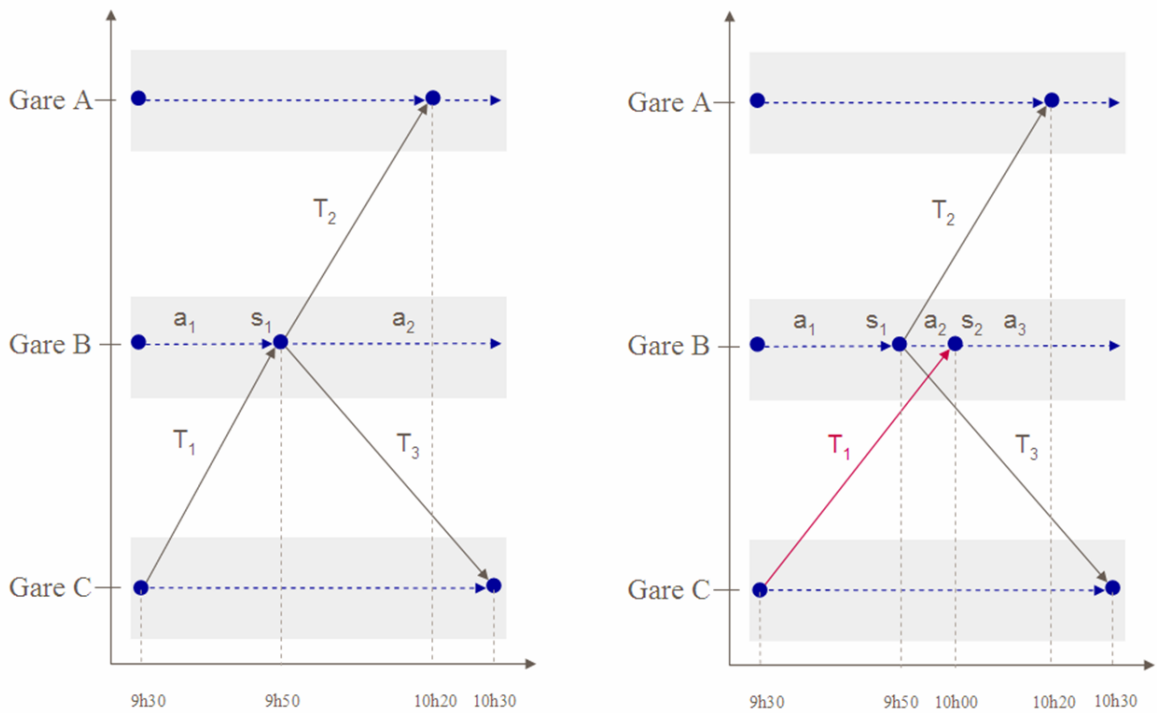


FIGURE 4.11 – Impact de la modification des données sur le graphe de PRESTO.

L'application d'une approche par intervalles impacterait la structure de P_{PRESTO} du PLNE, elle n'est donc pas envisagée en l'état. Elle nécessiterait l'expression d'un nouveau programme linéaire ou au moins une adaptation de P_{PRESTO} .

4.4.2 Light robustness

Le principe de light robustness proposé par Fischetti et Monaci (2009) s'appuie sur une approche par intervalles : chaque donnée incertaine (modélisée par un coefficient de la matrice ou de l'objectif d'un programme linéaire) est représentée par l'intervalle des valeurs qu'elle peut prendre. Cependant, la dégradation des coûts est contrôlée, permettant un compromis entre coûts et robustesse.

En voulant appliquer ce type d'approche à notre problème, on rencontrerait les mêmes difficultés que pour une approche par intervalles classique (cf. section 4.4.1).

4.4.3 Recoverable robustness

Plutôt que de chercher à être robuste à l'ensemble des scénarios réalisables à prendre en compte, on peut chercher à appliquer le principe de "recoverable robustness" (cf. section 3.2.5) : on construit des roulements robustes à un sous-ensemble des scénarios à considérer et on définit une stratégie de réparation facile et peu coûteuse à mettre en place en cas d'apparition d'un scénario du sous-ensemble complémentaire.

La mise en œuvre d'une telle approche permettant de contrôler le coût de la robustesse n'est pas évidente. Comme pour la méthode par scénarios initiale, il faut définir l'ensemble des scénarios à considérer, mais il faut aussi diviser cet ensemble en deux sous-ensembles. De plus, il faut faire le choix d'une méthode de résolution adaptée à la prise en compte de plusieurs scénarios et pouvoir prévoir des procédures de réparation de façon automatisée.

4.4.4 Synthèse

Le tableau 4.1 est un récapitulatif des méthodes évoquées et des avantages et des inconvénients de leur application à notre problème.

<i>Approche</i>	Principe	Avantages	Inconvénients, difficultés	Décision
Approche structurale	Intégration de la robustesse sous forme de critères afin d'agir sur la structure de la solution	Intégration simple des critères, coûts constants ou contrôlés	Définition de critères de robustesse, modélisation et méthode de résolution	Approche retenue pour la planification
Approche par scénarios	Incertitude sur les données modélisée sous forme de scénarios de retards	Modélisation directe des aléas pouvant apparaître	Définition de l'ensemble des scénarios à considérer, méthode de résolution complexe	Approche retenue pour l'évaluation
Approche par intervalles	Chaque donnée incertaine est représentée par l'intervalle des valeurs qu'elle peut prendre	Modélisation des aléas et contrôle du degré de robustesse	Modification de la structure du graphe et du PLNE existant, difficilement applicable	Approche non retenue
Light robustness	Variante de l'approche par intervalles avec contrôle de la dégradation des coûts	Contrôle des coûts : compromis entre coûts et robustesse	Modification de la structure du graphe et du PLNE existant, difficilement applicable	Approche non retenue
Recoverable robustness	Prise en compte des scénarios de retards directement ou par la mise en place de procédures de réparation	Différenciation des aléas à considérer directement et de ceux à réparer en opérationnel	Définition des scénarios, méthodes de résolution et de génération des procédures de réparation complexes	Approche non retenue

TABLE 4.1 – Synthèse de la discussion autour des approches d'intégration de la robustesse évoquées.

Les approches robustes classiques nous ont semblé théoriques par rapport à un contexte industriel. D'après les contraintes de notre problème et nos impressions a priori, nous avons choisi d'utiliser en priorité la méthode présentée dans la section 4.2, qui consiste à agir sur la structure d'une solution pour la rendre robuste selon des règles métier.

Deuxième partie

Approches de résolution

Dans cette partie, nous cherchons à répondre au problème de planification robuste des roulements d'engins. À partir de l'existant PRESTO, on cherche à intégrer de la robustesse sous forme de scénarios de retards et de critères de robustesse, afin de construire des roulements d'engins robustes en contrôlant les coûts. Nous présentons trois approches : une approche combinant une heuristique gloutonne et une recherche locale et deux approches par programmation linéaire, dont une orientée chemins et résolue de façon heuristique en utilisant la génération de colonnes (CGC) et une plutôt orientée tâches résolue exactement (MPT).

Chapitre 5

Vue d'ensemble

Afin de construire des roulements d'engins robustes, nous proposons trois approches permettant la résolution de problèmes comparables mais distincts, correspondant à des besoins différents :

- l'approche par heuristique gloutonne et recherche locale (HGL) donne une solution proche de celle de PRESTO rapidement, en conservant les mêmes coûts de production et en prenant en compte certains des critères de robustesse de construction proposés en section 4.2 ;
- l'approche par programmation linéaire discrète résolue de façon heuristique par génération de colonnes (CGC) garantit également des coûts de production égaux à ceux calculés par PRESTO, en intégrant au mieux certains critères de robustesse ;
- la dernière approche par programmation linéaire discrète résolue exactement (MPT) s'éloigne plus de la solution de PRESTO et permet une dégradation des coûts contrôlée afin de proposer des solutions plus robustes et respectant au mieux la maintenance.

5.1 Données et notations principales

Données d'entrée

Chaque approche est basée sur la solution de flots de PRESTO (cf. section 1.2.2) afin de pouvoir garantir une solution à coûts de production constants ou peu dégradés. Cependant, les informations conservées diffèrent selon l'approche utilisée.

L'étape de calcul des flots (P_{PRESTO}) de PRESTO est conservée et sa sortie fait partie intégrante des données d'entrée des trois problèmes traités. Chaque méthode a donc en entrée un certain nombre d'informations :

- des données structurelles : l'ensemble des gares, le nombre et l'emplacement des dépôts (pour la maintenance), les sillons à vide possibles (paires origine-destination, distance et temps de parcours) ;
- des données de demande : l'ensemble des tâches définies par des dates et des gares de départ et d'arrivée, la demande et le nombre d'engins maximum sur chacune ;
- des données techniques et réglementaires : pour chaque type d'engin on connaît le dépôt, la fréquence de passage et la durée d'immobilisation en maintenance, la distance et le temps de parcours de chaque gare au dépôt, et le nombre d'engins disponibles ;
- des données issues de PRESTO : le nombre d'engins utilisés pour chaque type d'engin, l'ensemble des tâches couvertes, les flots sur chaque arc et les sillons à vide horodatés utilisés.

Les approches ne se conforment pas toutes de la même façon aux données fournies par la solution de flots de PRESTO. Les approches (HGL) et (CGC) respectent exactement la solution de flots : nombre d’engins de chaque type sur chaque arc, sillons à vide horodatés. Ainsi, on conserve le même nombre d’engins, les mêmes déplacements en actif et en passif, et les mêmes sillons à vide. Les coûts d’engagement des engins et les coûts des déplacements en actif, en passif et des sillons à vide sont alors ceux calculés par PRESTO et sont par conséquent optimaux pour la demande couverte.

L’approche (MPT) récupère le nombre d’engins utilisés et les tâches couvertes, assurant ainsi une couverture maximale de la demande et des coûts limités : les coûts d’utilisation des engins sont élevés, il est impératif de les respecter. En revanche, elle permet une redistribution des flots et des sillons à vide : plutôt que de respecter exactement les flots calculés par la résolution de P_{PRESTO} , on répart des demandes et des capacités sur les tâches. De même, on n’utilise pas les sillons à vide de la solution de flots, on répart des paires origine-destination autorisées et utilisées par PRESTO pour créer des sillons à vide améliorant la solution en termes de robustesse.

Notations principales du problème

On rappelle ici les notations définies au Chapitre 1 qui seront utilisées dans cette partie :

N	Nombre d’engins (de lignes).
G_i^{dep}	Gare de départ de la tâche i .
G_i^{arr}	Gare d’arrivée de la tâche i .
D_i^{deb}	Date de début de la tâche i .
D_i^{fin}	Date de fin de la tâche i .
$dem(i)$	Nombre d’engins minimum requis pour couvrir la tâche i .
$cap(i)$	Nombre d’engins maximum autorisé sur la tâche i .
t_{ij}	Temps de retournement entre deux tâches T_i et T_j .

On définit ensuite de nouvelles notations utiles :

\mathcal{T}	Ensemble des tâches réelles (tâches à couvrir de demande non nulle).
$ \mathcal{T} $	Nombre de tâches à couvrir.
\mathcal{G}	Ensemble des gares.
$ \mathcal{G} $	Nombre de gares.
L	Horizon temporel du roulement.
M	Durée minimum entre la fin d’une ligne et le début de la suivante (cf. section 1.1.3).

5.2 Division du problème en sous-problèmes indépendants

Un point commun à toutes les approches est la division du problème en sous-problèmes indépendants par type d’engin et dans le cas particulier présenté en section 2 de non connexité du graphe Gr_{AR} .

D’une part, chaque type d’engin est considéré indépendamment. Pour les méthodes (HGL) et (CGC), les flots calculés lors de la Phase 1 de PRESTO sont respectés, il n’y a donc pas de contraintes couplantes. En revanche, la méthode (MPT) utilise la contrainte sur le nombre d’engins maximum sur un arc-tâche, couplante dans le cas d’UM composées d’engins de type différent. S’agissant d’un cas particulier, on permet une approximation.

D'autre part, pour les méthodes (CGC) et (MPT), on considère pour chaque type d'engin p le graphe Gr_{AR}^p décrit à la section 2.1. Selon le résultat obtenu au Chapitre 2, si ce graphe est connexe, on sait qu'il existe un roulement droit. Sinon, on définit un sous-problème par composante connexe¹. On résout alors plusieurs sous-problèmes.

Les méthodes (CGC) sans maintenance et (MPT) fournissent un roulement droit pour chaque sous-problème. Dans le cas d'un graphe Gr_{AR}^p non connexe, on aura autant de sous-roulements que de composantes connexes. Cependant, l'ajout de la maintenance par la Phase 4 de PRESTO ne garantit plus un nombre de sous-roulements minimum pour la méthode (CGC).

Dans les chapitres suivants, on considère un seul sous-problème indépendant.

5.3 Prise en compte de la robustesse et de la maintenance dans chaque approche

Les trois approches que nous proposons ici intègrent la robustesse sous forme de critères de robustesse à optimiser afin de construire des roulements robustes, mais toutes n'optimisent pas les mêmes critères. Par ailleurs, les approches ne gèrent pas toutes la maintenance directement.

On rappelle ici les critères retenus en section 4.2 :

- maximisation du temps de retournement minimum ($tmin$) ;
- homogénéisation des temps de retournement ($invret$) ;
- pénalisation des temps de retournement intermédiaires ($tretInt$) ;
- minimisation du nombre de recompositions ($recomp$).

On note que dans le cas où il existe des UM formées par des engins de type différent, le critère $recomp$ est couplant : l'indépendance des problèmes par type d'engin ne permet pas une gestion optimale du critère. On définit donc une approximation du critère dans chaque problème.

D'autres critères ont été retenus (cf. section 4.2), mais uniquement pour l'évaluation des solutions par simulation (cf. Chapitre 9). Ils ne sont pas évoqués lors de la description des approches proposées.

L'approche (HGL) prend en compte trois des critères de robustesse énumérés ci-dessus : $tmin$, $invret$ et $recomp$. La maintenance est ajoutée a posteriori par la Phase 4 de PRESTO.

L'approche (CGC) permet de modéliser les critères $tmin$ et $invret$, mais la combinaison de plusieurs critères complique la résolution du problème, seul $invret$ est alors retenu lors des tests. La maintenance est ajoutée a posteriori par la Phase 4 de PRESTO.

L'approche (MPT) permet de modéliser les critères $invret$, $tretInt$ et $recomp$, mais le modèle obtenu étant complexe, $recomp$ n'est pas pris en compte dans les tests. La maintenance et les critères de robustesse sont intégrés simultanément dans l'approche.

1. Concrètement, on utilise un algorithme de marquage pour la décomposition en composantes connexes.

5.4 Articulation de chaque approche avec PRESTO

On rappelle que chaque approche est basée sur la solution de flots de PRESTO. Dans tous les cas, la Phase 1 de PRESTO est conservée et les méthodes de résolution des trois approches interviennent en sortie de cette phase. Ensuite, les méthodes s'articulent différemment avec PRESTO.

La résolution par l'approche (HGL) remplace uniquement l'étape de construction des chemins (Phase 2). On effectue ensuite les étapes d'enchaînement de lignes et d'ajout de maintenance de PRESTO (Phases 3 et 4).

L'approche (CGC) a deux variantes. La première est applicable facilement et remplace les Phases 2 et 3 de PRESTO, puis la maintenance est effectuée par la Phase 4. La seconde variante s'appuie sur la première et permet en plus d'introduire des créneaux de maintenance dans le roulement en cours de construction. Elle n'a donc pas besoin du post-traitement d'ajout de la maintenance de PRESTO (Phase 4). Cette seconde variante n'a pas été implémentée et ne sera par conséquent pas détaillée par la suite.

La résolution par l'approche (MPT) permet d'obtenir directement une solution (roulements d'engins cycliques robustes avec maintenance), sans utiliser les Phases 2, 3 et 4 de PRESTO.

Le fonctionnement de PRESTO avec les méthodes proposées est illustré en figure 5.1.

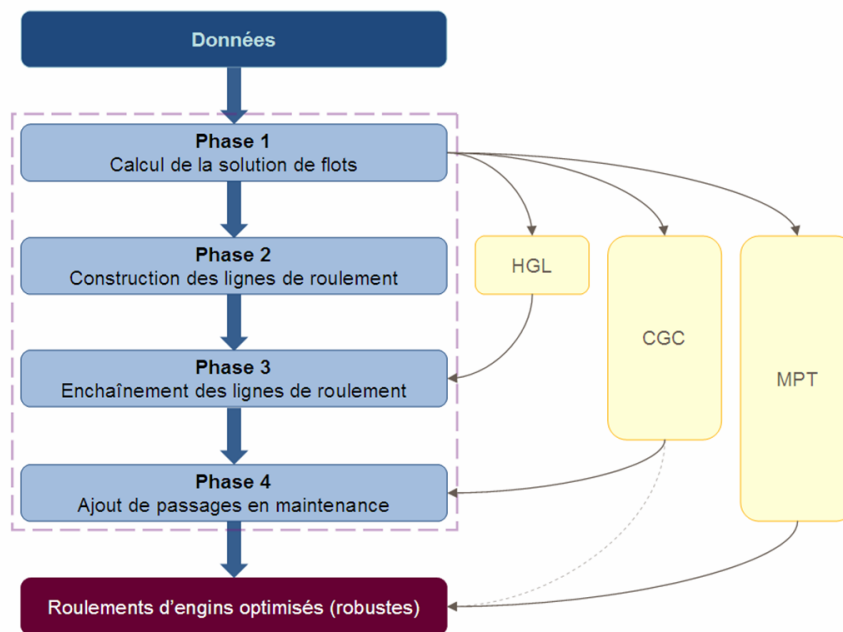


FIGURE 5.1 – Intégration des méthodes proposées dans PRESTO.

5.5 Résultats

En sortie de PRESTO avec robustesse intégrée, on obtient pour chacune des trois approches proposées des roulements d'engins :

- cycliques ;
- optimisés : coûts de production minimisés ou contrôlés ;
- robustes : selon les critères sélectionnés ;
- avec maintenance ;

- avec un nombre minimum de sous-roulements, voire composés uniquement de roulements droits (un par type d’engin).

Cependant, les roulements obtenus par les trois approches auront des caractéristiques différentes.

L’approche (HGL) donne une solution rapidement, proche de celle de PRESTO tout en améliorant les critères de robustesse. Cependant, les Phases 3 et 4, en particulier l’ajout de la maintenance, dégradent partiellement l’optimisation des critères de robustesse. De plus, comme pour la solution de PRESTO, la fréquence de passage en maintenance n’est pas forcément respectée.

L’approche (CGC) gère simultanément la construction et l’enchaînement des lignes, permettant ainsi une meilleure optimisation des critères de robustesse, mais présente les mêmes difficultés que l’approche (HGL) concernant le respect des contraintes de maintenance. La seconde variante (non testée) peut pallier cette faiblesse, mais l’application est complexe, en particulier concernant le temps d’exécution et l’utilisation de la mémoire des moteurs de calcul. En sortie de la première variante, on obtient un roulement droit par type d’engin et les gains en robustesse sont équivalents à ceux de l’approche (HGL). Les solutions obtenues restent proches de celles de PRESTO.

La dernière approche (MPT) étant moins contrainte par la solution de flots, les roulements obtenus sont plus éloignés de ceux de PRESTO, avec une variation possible des coûts de production, mais plus robustes et plus réguliers en termes de maintenance. On obtient une solution sous forme d’un roulement droit par sous-problème indépendant.

5.6 Instances cibles

Des tests unitaires sur des jeux fictifs ou de petite taille et des tests fonctionnels sur trois instances réelles ont été réalisés. Ces tests sont utiles pour fixer les modèles et les méthodes de résolution et valider le fonctionnement des méthodes.

Pour fixer chaque modèle (expression des contraintes, valeurs des paramètres, etc.) et chaque méthode de résolution, nous avons testé plusieurs variantes (plusieurs valeurs pour un paramètre, plusieurs types de voisinage, etc.), afin de sélectionner la plus pertinente selon les besoins.

On s’assure aussi de la validité des solutions obtenues : roulements cohérents, amélioration des critères sélectionnés selon les attentes.

Les trois instances choisies sont représentatives de l’échantillon des instances réelles à notre disposition, en considérant des caractéristiques telles que le type d’engin ou le nombre de tâches à couvrir récapitulées dans le tableau 5.1.

<i>Instance</i>	Région	Type d’engin	Nombre de tâches à couvrir	Nombre de tâches couvertes	Nombre d’engins minimum
1	Bretagne	B82500	208	207	7
2	Rhône-Alpes	BB67300	314	314	11
3	Bretagne	Z27500 3C	624	597	15

TABLE 5.1 – Instances cibles permettant de fixer les méthodes et de valider le fonctionnement des approches proposées

Des résultats obtenus par chaque méthode une fois entièrement fixée sont présentés et com-

parés à ceux de PRESTO afin de valider le fonctionnement des approches proposées (cf. sections 6.4, 7.4 et 8.4).

Par la suite, neuf instances réelles ont été sélectionnées afin de mener une analyse comparative des solutions des différentes approches (cf. Chapitre 10).

Pour chacune de ces instances, il existe un roulement droit. D'une part, la version de PRESTO utilisée traite des instances composées d'un unique type d'engin. D'autre part, le cas de non connexité du graphe Gr_{AR} est un cas particulier.

Chapitre 6

Approche par heuristique gloutonne et recherche locale

L'approche (HGL) a pour but de construire des roulements d'engins robustes à coûts de production constants rapidement. Des critères de robustesse sont optimisés lors d'une phase robuste qui intervient après la Phase 1 de PRESTO : P_{PRESTO} n'est pas modifié afin de récupérer la solution optimale en termes de coûts de production. Le problème considéré est très contraint par la solution de flots issue de P_{PRESTO} (cf. section 1.2.2), afin de pouvoir le résoudre rapidement par une heuristique. L'articulation de la méthode (HGL) avec PRESTO est illustrée en Figure 6.1.

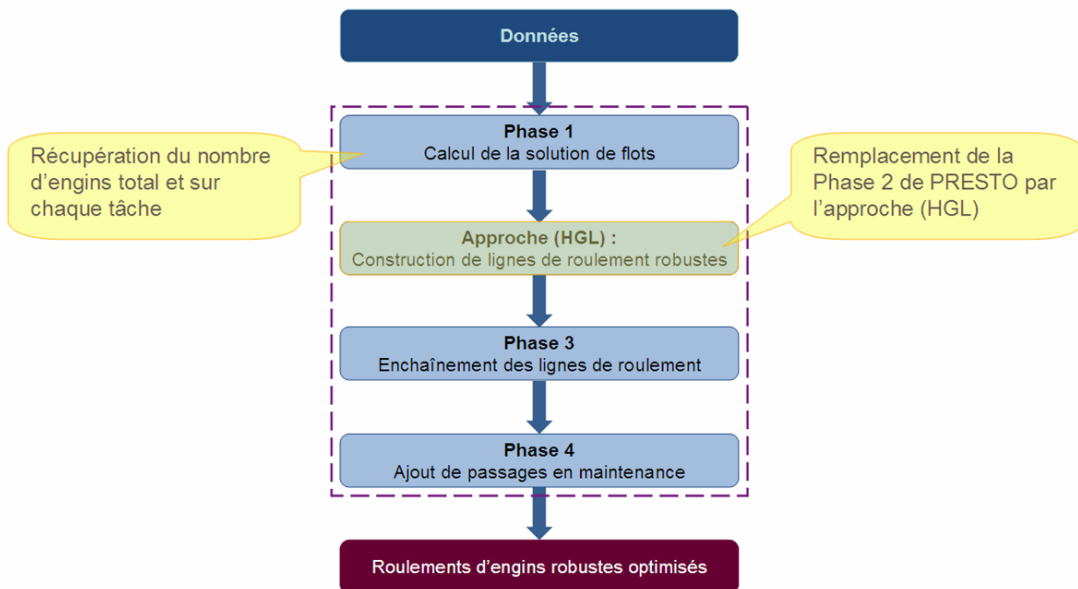


FIGURE 6.1 – Intégration de l'approche (HGL) dans PRESTO.

6.1 Spécifications du problème considéré

Principe

Afin de rester à coûts de production constants, la solution de flots est conservée et constitue une entrée du problème étudié. On considère ici les sillons à vide (W) comme des tâches qu'il faut couvrir en respectant exactement le flot calculé lors de la résolution de P_{PRESTO} . On connaît donc le nombre d'engins total et le nombre d'engins du type considéré circulant sur chaque tâche (on traite ici indépendamment chaque type d'engin, cf. section 5.2).

Lors de la Phase 2 de PRESTO (cf. Figure 1.10), on cherche à construire des lignes qui respectent les flots calculés en Phase 1, puisque le respect de ces flots garantit une solution optimale en termes de couverture de la demande et de coûts de production. L'approche (HGL) remplace la Phase 2 de PRESTO : elle permet de construire des lignes de roulement robustes. Dans ce but, on construit des chemins respectant les flots sur les tâches (cf. Figure 1.8) en intégrant des critères de robustesse sélectionnés.

L'enchaînement des lignes de roulement robustes ainsi construites et l'ajout de créneaux de maintenance se font ensuite par les Phases 3 et 4 de PRESTO.

En sortie, on obtient alors des roulements robustes¹.

Critères de robustesse

Afin d'obtenir des roulements robustes, on intègre les critères de robustesse suivants, détaillés à la section 4.2 :

- homogénéisation des temps de retournement (*invret*) ;
- maximisation du temps de retournement minimum (*tmin*) ;
- minimisation du nombre de recompositions (*comp*).

6.2 Modélisation par un graphe issu du graphe de PRESTO

Afin de modéliser les données d'entrée du problème spécifié précédemment (cf. section 6.1), on récupère le graphe spatio-temporel de PRESTO décrit à la section 1.2.1, sur les arcs duquel sont renseignés les flots correspondant au type d'engins considéré uniquement. Seule la source correspondant au type considéré est conservée (et notée S dans la suite).

À partir de ce graphe, il s'agit de trouver N chemins de S à P couvrant exactement les flots indiqués sur les arcs, en intégrant les critères de robustesse.

6.3 Méthode de résolution

Le problème considéré ici est résolu en deux étapes distinctes.

Il s'agit en premier lieu de construire une solution initiale, c'est-à-dire un ensemble de lignes de roulement à partir de la solution de flots de PRESTO (cf. Figure 1.8). Les critères de robustesse énumérés à la section 6.1 sont déjà pris en compte lors de cette construction. Dans ce but, on utilise

1. On ne garantit pas l'obtention d'un roulement droit dans cette méthode.

une heuristique qui construit des chemins dans le graphe de manière gloutonne.

On cherche ensuite à améliorer cette solution par recherche locale : la solution est modifiée localement dans un voisinage, puis on évalue les critères de robustesse sélectionnés. À chaque itération, on garde la meilleure solution.

6.3.1 Evaluation d'un roulement

Ligne et roulement partiels

Une *ligne partielle* est une ligne de roulement tronquée, dont les tâches doivent se suivre. On complète une ligne partielle en ne remettant pas en question l'enchaînement des tâches déjà affectées : on peut ajouter des tâches à la suite de la dernière tâche de la ligne partielle uniquement.

Un *roulement partiel* est un ensemble de lignes de roulement partielles ou complètes. En plus de ne pas respecter les valeurs des flots, il peut donc aussi violer les contraintes de cyclicité.

Calcul des critères de robustesse

Nous calculons ici les critères de robustesse énumérés à la section 6.1 pour une ligne partielle ou complète, ou pour un roulement partiel.

Calcul de *invret* On rappelle que *invret* est la somme des inverses des temps de retournement d'un roulement (cf. section 4.2). Sa minimisation permet d'homogénéiser les temps de retournement du roulement.

***invret* d'une ligne** Dans le cas d'une ligne, on ne somme que les inverses des temps de retournement connus. Soit une ligne ℓ partielle ou complète consistant en une suite de tâches (T_1, \dots, T_n) , le calcul est le suivant :

$$invret(\ell) = \sum_{i=1}^{n-1} \frac{1}{t_{i,i+1}}$$

***invret* d'un roulement** la valeur du critère *invret* d'un roulement partiel est alors égale à la somme des valeurs des *invret* des lignes de roulement partielles :

$$invret = \sum_{\ell=1}^N invret(\ell)$$

La valeur du critère *invret* sur le roulement complet sera forcément supérieure à toute valeur de ce critère sur un roulement partiel.

Calcul de *tmin* Le critère *tmin* est le minimum des temps de retournement d'un roulement (cf. section 4.2).

***tmin* d'une ligne** Dans le cas d'une ligne, on détermine le minimum des temps de retournement existants. Soit une ligne partielle ou complète consistant en une suite de tâches (T_1, \dots, T_n) , *tmin* se calcule comme suit :

$$tmin(\ell) = \min_{i \in \llbracket 1, n-1 \rrbracket} (t_{i,i+1})$$

$tmin$ d'un roulement La valeur du critère $tmin$ d'un roulement partiel est alors le minimum des valeurs de $tmin$ de toutes les lignes :

$$tmin = \min_{\ell \in \llbracket 1, N \rrbracket} (tmin(\ell))$$

La valeur de $tmin$ d'un roulement complet est nécessairement inférieur ou égal au $tmin$ sur un roulement partiel (ou sur une ligne).

Calcul de $recomp$ On rappelle que $recomp$ est le critère consistant à minimiser le nombre de recompositions d'un roulement (cf. section 4.2).

$recomp$ d'une ligne Le critère $recomp$ concernant des UM uniquement, il n'existe pas pour une ligne, puisqu'une ligne correspond à un unique engin.

On propose alors un nouveau critère $recomp(\ell)$ pour une ligne ℓ : le nombre de recompositions concernant ℓ . Il s'agit de compter le nombre de recompositions engendrées par les affectations de tâches à cette ligne. En effet, puisqu'on connaît les flots sur chaque tâche, on sait sur quelles tâches seront les futures UM.

Soit un engin couvrant deux tâches qui se suivent :

- si les deux tâches sont en US, aucune recomposition ne sera nécessaire ;
- si la première est en UM et la seconde en US, une recomposition sera nécessaire pour découpler l'engin de l'UM entre les tâches, on compte donc une recomposition ;
- si la première est en US et la seconde en UM, une recomposition sera nécessaire pour coupler l'engin à l'UM entre les tâches, on compte donc une recomposition ;
- si les deux tâches sont en UM, aucune recomposition ne sera nécessaire.

Soit la ligne ℓ_1 de la Figure 6.2, correspondant à l'engin 1 de la Figure 6.3. L'engin 1 enchaîne trois tâches T_1 , T_3 et T_4 . La tâche T_3 est couverte par une UM. Mais l'engin 1 couvre d'abord la tâche T_1 en US. Il y aura donc au moins une recomposition pour coupler l'engin 1 aux autres engins de l'UM couvrant la tâche T_3 . De même, il y aura une recomposition supplémentaire après T_3 . On note alors $recomp(\ell_1) = 2$.

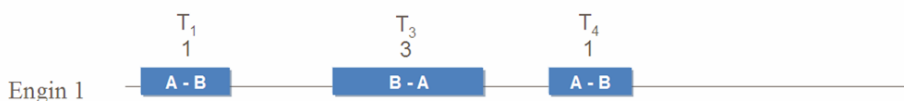


FIGURE 6.2 – Exemple de roulement nécessitant des recompositions.

$recomp$ d'un roulement Pour calculer le critère $recomp$ d'un roulement ou d'un roulement partiel rlt , on additionne les recompositions nécessaires à chaque UM.

On considère l'exemple de la Figure 6.3. La tâche T_2 est couverte par une UM formée des engins 2 et 3, on compte une recomposition avant la tâche. Puis l'UM est conservée pour la tâche T_3 . Il est donc inutile de découpler puis de recoupler les mêmes engins. Cependant, l'engin 1 est ajouté à l'UM, il y a donc une recomposition. Puis les trois engins sont découplés pour couvrir trois tâches différentes, on compte donc deux recompositions. Enfin, deux recompositions sont nécessaires pour

coupler puis découpler les engins en UM sur la tâche T_7 . Au total, on compte six recompositions : $recomp(rlt) = 6$.

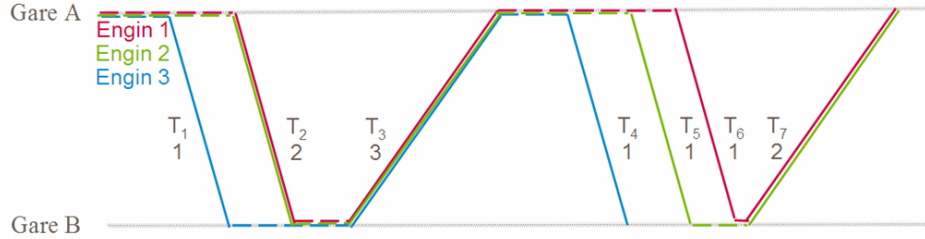


FIGURE 6.3 – Exemple de roulement nécessitant des recompositions.

Le nombre de recompositions dans un roulement rlt est alors nécessairement supérieur ou égal au nombre de recompositions engendrées par chaque ligne, mais inférieur ou égal à la somme des nombres de recompositions engendrées par toutes les lignes :

$$\max_{\ell \in \llbracket 1, N \rrbracket} (recomp(\ell)) \leq recomp(rlt) \leq \sum_{\ell=1}^N (recomp(\ell))$$

Agrégation des critères

Afin de comparer deux solutions, on commence par calculer les critères de robustesse de chacune selon les méthodes indiquées ci-dessus. Il faut ensuite agréger ces critères.

Une somme pondérée des critères est difficile à interpréter d'un point de vue métier. Or l'utilisateur doit pouvoir guider l'outil pour obtenir une solution robuste selon ses attentes.

Une hiérarchie des critères définie par l'utilisateur est alors considérée comme une donnée d'entrée du problème. Concrètement, l'utilisateur numérote les critères définis à la section 6.1 de 1 à 3. On obtient ainsi une liste des critères (c_1, c_2, c_3) ordonnée du plus important au moins significatif.

On choisit ensuite la meilleure solution selon l'ordre lexicographique².

6.3.2 Heuristique gloutonne de construction d'une solution initiale

Initialement, le roulement est vide, et nous cherchons à le construire en affectant les tâches une à une aux lignes de roulement. Les tâches sont traitées dans l'ordre chronologique de leurs dates de début.

Lors du traitement d'une tâche, on parcourt l'ensemble des lignes de roulement partielles afin de tester l'affectation de la tâche courante à chaque ligne. Une tâche T_i peut être affectée à une ligne si elle peut suivre la dernière tâche T_j de cette ligne, c'est-à-dire si T_j est telle que : $G_j^{arr} = G_i^{dep}$ et $D_i^{deb} \geq D_j^{fin} + tc(G_j^{arr})$.

Lorsque T_i peut être affectée à plusieurs lignes, on compare les valeurs des critères de robustesse partiels pour chaque affectation. On choisit ensuite la meilleure affectation en fonction des valeurs

2. On choisit la solution pour laquelle c_1 prend la meilleure valeur. À valeurs équivalentes pour c_1 , on s'intéresse à c_2 . À valeurs équivalentes pour c_2 , on s'intéresse à c_3 .

des critères.

Afin de garantir le respect des flots calculés de la solution de flots, une tâche T_i couverte par une UM est dupliquée de façon à être traitée $flot(T_i)$ fois. Elle est alors affectée à $flot(T_i)$ lignes de roulement.

On construit ainsi les lignes de roulement tâche par tâche. L'Algorithme 1 détaille l'heuristique gloutonne.

Algorithme 1 Heuristique gloutonne de construction d'une solution initiale

```

// Initialisation de la solution
 $S = \emptyset$ 
// Parcours des tâches à couvrir dans l'ordre chronologique
pour tout  $T_i \in \mathcal{T}$  faire
  // Traitement de la tâche autant de fois que nécessaire selon son flot
  pour tout  $k = 1..flot(T_i)$  faire
    // Parcours des lignes de roulement
    pour tout  $l \in [1, N]$  faire
      Soit  $T_j$  la dernière tâche de  $l$ 
      // Si  $T_i$  peut suivre  $T_j$  sur  $l$ 
      si  $G_j^{arr} = G_i^{dep}$  et  $D_i^{deb} \geq D_j^{fin} + tc(G_j^{arr})$  alors
        Soit  $S'$  la solution courante mise à jour avec l'affectation de  $T_i$  à  $l$ 
        Evaluation des critères de robustesse de  $S'$ 
        si  $S'$  domine  $S$  ou s'il s'agit d'une première affectation de  $T_i$  alors
           $S \leftarrow S'$ 
        fin si
      fin si
    fin pour
  fin pour
fin pour

```

6.3.3 Amélioration de la solution courante par recherche locale

Une fois qu'on a obtenu une solution initiale par l'heuristique gloutonne, on cherche à améliorer cette solution par recherche locale : la solution est modifiée localement dans un voisinage. Si la modification effectuée permet d'améliorer les critères de robustesse, la nouvelle solution est conservée.

On répète ensuite cette opération jusqu'à atteindre une condition d'arrêt : on choisit d'arrêter la recherche locale après un temps d'exécution fixé par l'utilisateur.

Voisinage par croisement de lignes

On note S la solution courante. Initialement, S correspond à la solution obtenue par l'heuristique gloutonne.

Parmi les N lignes de S , on s'intéresse aux N^- lignes les moins robustes (d'après les critères de robustesse dans l'ordre lexicographique).

Une tâche T_i est sélectionnée aléatoirement sur l'une de ces N^- lignes. On note T_j la tâche qui précède T_i sur cette ligne l . On cherche ensuite une paire de tâches T'_i et T'_j qui se suivent sur

une ligne ℓ' telles que T_i peut suivre T'_j et T'_i peut suivre T_j . Le voisinage par croisement de lignes consiste alors à échanger les deux fins de lignes commençant par T_i et T'_i (croisement des lignes ℓ et ℓ'). On note que la fin de ligne de ℓ ici peut correspondre à la quasi-totalité de la ligne si T_i est par exemple la deuxième ou la troisième tâche de ℓ (de même pour ℓ').

On obtient une nouvelle solution S' . On note que le voisinage par croisement de ligne conserve la cyclicité puisque le nombre d'engins en chaque gare en début et en fin d'horizon temporel n'est pas modifié. On garde alors uniquement la solution la plus robuste (désormais notée S).

Tant que la condition d'arrêt n'est pas respectée, c'est-à-dire tant qu'on n'a pas atteint le temps de calcul imposé, on répète ces différentes étapes.

Algorithme

La procédure de recherche locale est reprise par l'Algorithme 2.

Algorithme 2 Procédure de recherche locale

Soit S la solution courante (solution initiale calculée par l'heuristique)
tant que Condition d'arrêt non respectée **faire**
 $S' \leftarrow S$
 Tirage aléatoire d'une ligne ℓ parmi les N^- lignes de S' les moins robustes
 Tirage aléatoire d'une tâche T_i sur une ligne ℓ parmi ces N^- lignes
 Tirage aléatoire d'une ligne ℓ' pouvant être croisée avec ℓ à la tâche T_i
 Croisement de ℓ et ℓ'
 si S' domine S **alors**
 $S \leftarrow S'$
 fin si
fin tant que

Le fonctionnement de la procédure de recherche locale est illustré sur l'exemple de la Figure 6.4. On considère six tâches T_1 à T_6 , chacune devant être couverte par un engin. Une solution initiale possible est donnée en Figure 6.5.

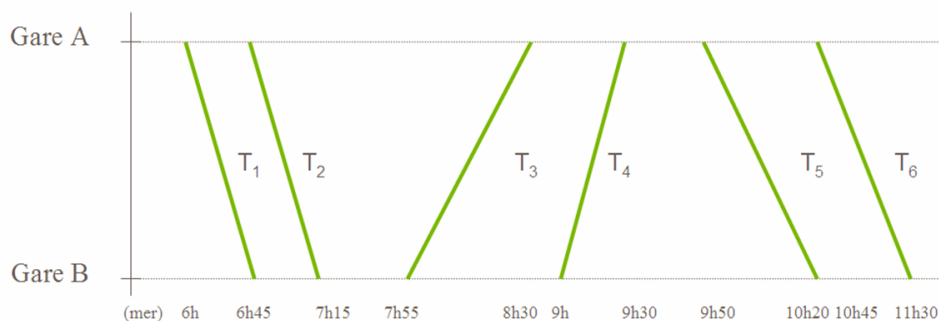


FIGURE 6.4 – Recherche locale : tâches à couvrir.

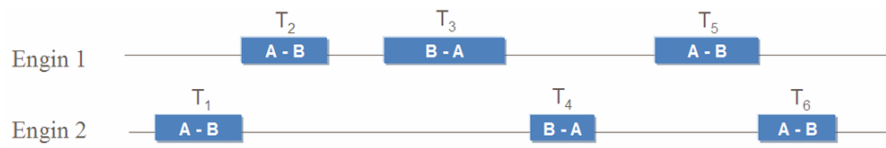


FIGURE 6.5 – Recherche locale : solution initiale obtenue pour la demande de la Figure 6.4.

La tâche T_3 de prédécesseur T_2 est sélectionnée aléatoirement. Comme indiqué sur la Figure 6.6, on remarque que sur la ligne 2, T_4 pourrait suivre T_2 , car T_2 arrive en gare B (gare de départ de T_4), et l'intervalle entre T_2 et T_4 est suffisant (si on considère un temps de crochet de 8 minutes pour la gare B). On intervertit alors la fin de la ligne 1 et la fin de la ligne 2 et on obtient une nouvelle solution (Figure 6.7).

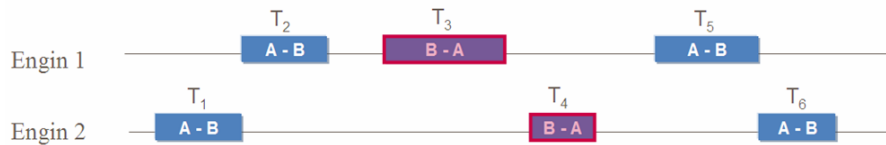


FIGURE 6.6 – Recherche locale : sélection des deux tâches à échanger.

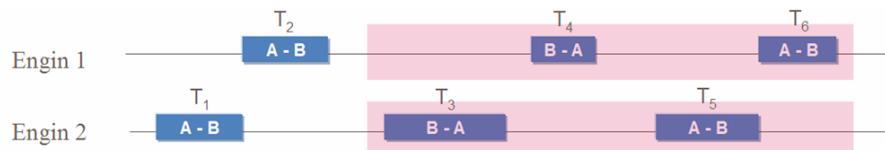


FIGURE 6.7 – Recherche locale : construction d'une nouvelle solution par croisement des lignes.

6.4 Premiers résultats et conclusions

Les tests présentés dans cette section permettent de comparer les solutions obtenues par l'approche (HGL) aux solutions existantes (obtenues par PRESTO) sur les trois instances cibles décrites en section 5.6. Nous faisons dans cette section quelques premières observations sur le comportement de la méthode de résolution proposée.

6.4.1 Tests sur les instances cibles

On cherche ici à comparer les solutions obtenues par PRESTO et par l'approche (HGL) pour les instances cibles présentées à la section 5.6. Pour l'approche (HGL), on choisit un temps d'exécution de la recherche locale limité à 5 minutes. D'après des tests effectués préalablement, il apparaît que c'est une durée suffisante pour obtenir une bonne solution.

Tests sans maintenance

Nous favorisons ici le critère *invret*. La hiérarchie des critères utilisée est alors la suivante :

1. homogénéisation des temps de retournement par minimisation de la somme des inverses de ces temps (*invret*) ;
2. maximisation du temps de retournement minimum (*tmin*) ;
3. minimisation du nombre de recompositions (*recomp*).

Les premiers résultats présentés concernent les méthodes sans maintenance. L'impact du post-traitement d'ajout de la maintenance sur la robustesse des solutions est abordé dans la Table 6.2.

La Table 6.1 présente les résultats pour les instances cibles (cf. Table 5.1). Les valeurs de plusieurs critères sont fournies afin de comparer les solutions de PRESTO et de (HGL) :

- le temps d'exécution total : pour l'approche (HGL), il correspond au temps d'exécution de PRESTO auquel on ajoute le temps d'exécution de (HGL) ;
- le critère *invret* : valeur absolue et gain relatif de l'approche (HGL) par rapport à PRESTO ;
- le critère *recomp* : valeur absolue et gain relatif de l'approche (HGL) par rapport à PRESTO.

Instance	Instance 1		Instance 2		Instance 3	
	PRESTO	HGL	PRESTO	HGL	PRESTO	HGL
Temps d'exécution total (en secondes)	300	600	60	300	60	300
Valeur de <i>invret</i>	9,18	7,92	10,52	7,27	117,60	113,97
Gain relatif sur <i>invret</i>	-	13,7%	-	30,9%	-	3,1%
Valeur de <i>recomp</i>	8	8	0	0	40	44
Gain relatif sur <i>recomp</i> par rapport à PRESTO	-	0%	-	0%	-	-10%

TABLE 6.1 – Premiers résultats de PRESTO et de (HGL) favorisant *invret* sur les instances cibles

Des tests complémentaires nous ont permis de constater que les résultats sur ces trois instances étaient quasiment identiques :

- pour un temps d'exécution à moins d'une minute ;
- en favorisant le critère *recomp* avant *invret*.

Le gain sur le critère étudié varie en fonction de l'instance entre 3% (instance 3) et 30% (instance 2). On obtient donc des résultats encourageants en termes de robustesse.

Les solutions permettant un gain de robustesse (sur *invret*) ont été calculées rapidement : en moins d'une minute, l'approche (HGL) permet de donner une bonne solution pour les trois instances cibles testées.

Impact de l'ajout de la maintenance sur les critères de robustesse

La Table 6.1 correspond aux solutions sans maintenance. Les méthodes PRESTO et (HGL) ajoutent toutes deux la maintenance par le même post-traitement en fin de processus (Phase 4 de PRESTO, cf. Figure 5.1). Or ce post-traitement modifie les lignes de roulement, risquant ainsi de dégrader les critères de robustesse optimisés.

Dans la Table 6.2, nous présentons les valeurs après ajout de créneaux de maintenance (se référer à la Table 6.1 pour les valeurs de *invret* avant maintenance). Seules les instances cibles 1 et 2 sont concernées. En effet, pour l'instance cible 3, PRESTO ne trouve pas de solution avec maintenance, ce qui peut venir des données d'entrée, par exemple s'il n'y pas d'évolution possible

vers un dépôt, ou au nombre d’engins calculé, par exemple si on ne peut pas générer de creux de roulement assez long pour insérer des créneaux de maintenance.

Instance	Instance 1		Instance 2	
	PRESTO	HGL	PRESTO	HGL
Valeur de <i>invret</i> après maintenance	15,50	15,09	21,62	18,83
Gain relatif sur <i>invret</i>	-	2,6%	-	12,9%

TABLE 6.2 – Impact de l’ajout de la maintenance sur la valeur de *invret*

Pour les deux instances testées, la valeur de *invret* est très significativement dégradée après ajout de la maintenance, puisque le gain passe de plus de 13% à moins de 3% pour l’instance 1, et de plus de 30% à moins de 13% pour l’instance 2.

On constate cependant que les solutions obtenues par l’approche (HGL) restent plus intéressantes que les solutions obtenues par PRESTO.

6.4.2 Premières conclusions

On traite ici un problème très contraint puisqu’on impose un certain nombre de données supplémentaires issues de la résolution de P_{PRESTO} de PRESTO :

- le nombre d’engins utilisés ;
- le nombre d’engins de chaque type circulant sur chaque tâche (flots) ;
- les sillons à vide (dates, OD et nombre d’engins de chaque type circulant).

Ces informations étant fixées a priori, on garantit l’obtention d’une solution à coûts constants par rapport à la solution de PRESTO, c’est-à-dire à coûts de production minimaux.

On obtient un gain intéressant en termes de robustesse.

Cependant, la maintenance étant ajoutée en fin de processus et sans objectif, les critères de robustesse sont dégradés entre la solution intermédiaire en sortie de l’étape robuste sans maintenance et la solution en fin de processus. La solution obtenue par la méthode (HGL) reste malgré tout intéressante par rapport à celle de PRESTO.

L’algorithme utilisé est peu gourmand en mémoire et rapide. Ainsi, il permet de balayer un ensemble de solutions important de façon à trouver une solution intéressante en un temps limité.

Par ailleurs, l’approche (HGL) en l’état n’est pas adaptée à un problème étendu (flots ou sillons à vide non fixés).

6.4.3 Industrialisation et perspectives

L’approche (HGL) consiste en un module complètement intégré à PRESTO : le format des entrées et des sorties de PRESTO est conservé. L’exécution est rapide et permet d’obtenir des solutions améliorant les critères de robustesse. L’approche (HGL) est donc intéressante et facilement industrialisable. C’est cette méthode qui a été choisie pour les premiers tests d’industrialisation.

Cependant, bien qu’il y ait un gain en robustesse globalement, le principal inconvénient de l’approche (HGL) est la dégradation des critères de robustesse par le post-traitement d’ajout de

créneaux de maintenance. Pour remédier à cette faiblesse, on peut envisager deux démarches : tenter d'intégrer la maintenance lors de la construction des lignes, ou tenter de prendre en compte les critères de robustesse lors du post-traitement de maintenance. Dans les deux cas, rien ne garantit l'existence d'une heuristique rapide telle que celle mise en place. On risque donc de perdre le premier intérêt de l'approche (HGL), c'est-à-dire la possibilité d'obtenir une solution intéressante rapidement.

Chapitre 7

Approche utilisant la technique de génération de colonnes

L'approche (CGC) a pour but de construire des roulements d'engins robustes à coûts de production constants par l'intégration de critères de robustesse. Après la Phase 1 de PRESTO permettant de calculer la solution de flots, un PLNE P_{CGC} correspondant à une vision chemins est résolu de façon heuristique par génération de colonnes afin de construire un roulement d'engins robuste sans maintenance. L'articulation de la méthode (CGC) avec PRESTO est illustrée en Figure 7.1.

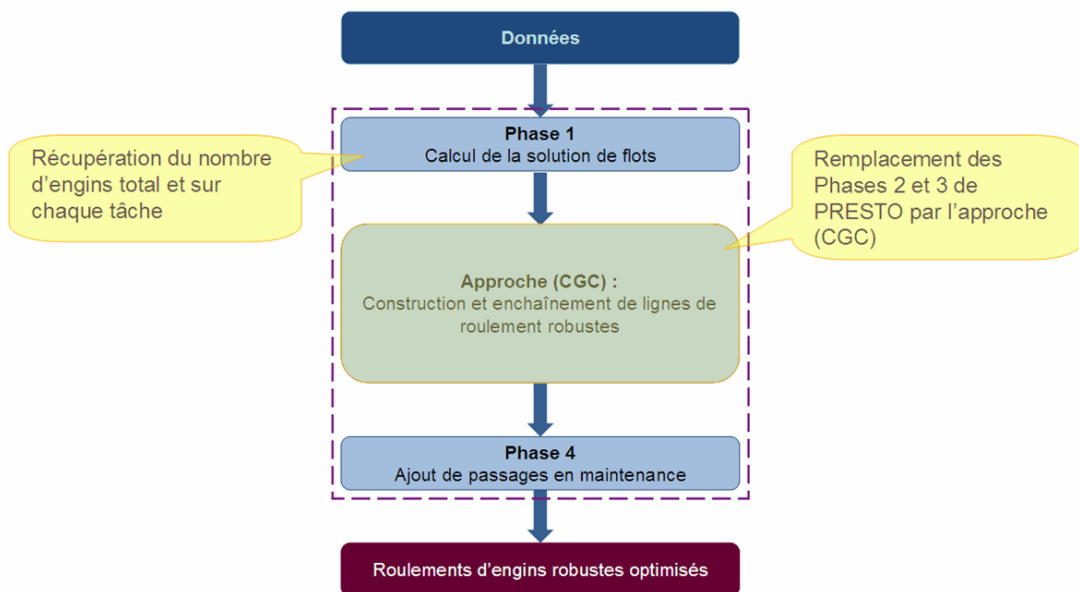


FIGURE 7.1 – Intégration de l'approche utilisant la génération de colonnes dans PRESTO.

7.1 Spécifications du problème considéré

Comme pour l'approche précédente (HGL) au Chapitre 6, la solution de flots obtenue par la résolution de P_{PRESTO} est considérée comme une donnée d'entrée pour cette approche (CGC) : on connaît le nombre d'engins total, les tâches et les sillons à vide à couvrir, le nombre d'engins

circulant sur chaque arc du graphe de PRESTO (décrit à la section 1.2.1). En respectant ces données, on garantit le respect des coûts de production optimaux calculés par PRESTO.

L'approche (CGC) remplace les Phases 2 et 3 de PRESTO par une unique phase de construction et d'enchaînement de lignes de roulement robustes. La robustesse est intégrée ici sous la forme du critère *invret* qui consiste à homogénéiser les temps de retournement dans le roulement en minimisant la somme des inverses de ces temps de retournement. Le problème considéré est alors le suivant : en partant de la solution de flots, on cherche à construire N chemins de la source au puits minimisant le critère *invret* et à les affecter aux N lignes, de façon à constituer un roulement cyclique où chaque tâche est couverte par exactement le nombre de lignes nécessaire.

L'ajout de créneaux de maintenance se fait ensuite par la Phase 4 de PRESTO.

En sortie, on obtient des roulements d'engins robustes.

7.2 Modélisation du problème

Afin de modéliser le problème (correspondant au type d'engin p considéré) traité par l'approche (CGC) par un PLNE P_{CGC} , on se base sur le graphe Gr^p (cf. section 7.2.1). Chaque variable de P_{CGC} correspond à l'affectation d'un chemin c de S à P dans le graphe à une ligne de roulement. Dans ce cas, les caractéristiques de chaque chemin sont connues à l'avance : tâches couvertes et ordre de ces tâches dans le chemin, dates des première et dernière tâches, valeurs des critères de robustesse.

7.2.1 Modélisation par un graphe issu du graphe de PRESTO

De même que pour l'approche (HGL), les données d'entrée du problème sont modélisées par le graphe Gr^p issu de la solution de flots de PRESTO et dont la construction est détaillée à la section 2.1 (cf. exemple en Figure 2.1) :

- seuls les flots correspondant au type courant sont renseignés ;
- seule la source correspondant au type considéré est conservée (notée S dans la suite) ;
- les arcs de flot nul et les sommets isolés sont supprimés.

À partir de ce graphe, il s'agit de trouver N chemins de S à P couvrant exactement les flots indiqués sur les arcs, en intégrant le critère de robustesse *invret*.

7.2.2 Notations

En plus des notations principales énumérées à la section 5.1, on définit des notations permettant de manipuler un chemin c de la source S au puits P dans le graphe :

$\mathcal{T}(c)$	Ensemble des tâches du chemin c .
G_c^{dep}	Gare de départ du chemin c .
G_c^{arr}	Gare d'arrivée du chemin c .
D_c^{deb}	Date de début de la première tâche du chemin c .
D_c^{fin}	Date de fin de la dernière tâche du chemin c .
$invret(c)$	Somme des inverses des temps de retournement du chemin c .
\mathcal{C}_{max}	Nombre maximum de chemins possibles dans le graphe.
$\delta^-(v)$	Ensemble des sommets prédécesseurs du sommet v .

7.2.3 Modélisation des critères de robustesse

Dans cette approche, seul le critère *invret* défini à la section 4.2.1 est intégré. On rappelle qu'afin d'homogénéiser les temps de retournement dans le roulement *rlt*, ce critère consiste à minimiser la somme des inverses des temps de retournement du roulement :

$$invret(rlt) = \sum_{(T_i, T_j) \in rlt} \frac{1}{t_{ij}}$$

On peut aussi l'exprimer comme la somme des *invret* des lignes du roulement :

$$invret(rlt) = \sum_{\ell=1}^N invret(\ell)$$

Comme à chaque ligne est affecté un chemin *c*, on peut écrire :

$$invret(rlt) = \sum_{c \text{ affecté à une ligne } \ell} invret(c).$$

7.2.4 Modélisation du problème par un PLNE

On considère le problème de construction d'un roulement d'engins à partir de la solution de flots de PRESTO comme un problème d'affectation de chemins de la source au puits dans le graphe aux lignes de roulement (engins). On modélise ce problème par un programme linéaire en nombres entiers P_{CGC} , dont les variables de décision $X_{c,\ell}$ sont telles que :

$$X_{c,\ell} \begin{cases} = 1 & \text{si le chemin } c \text{ est affecté à la ligne } \ell \text{ du roulement.} \\ = 0 & \text{sinon.} \end{cases}$$

avec

$$\begin{aligned} \ell \in [1, N] & \quad \text{Indice de la ligne de roulement traitée.} \\ c \in [1, \mathcal{C}_{max}] & \quad \text{Indice du chemin traité.} \end{aligned}$$

Le PLNE obtenu est le suivant :

$$P_{CGC} : \min \left(\sum_{\ell=1}^N \sum_c^{\mathcal{C}_{max}} (invret(c) \cdot X_{c,\ell}) \right)$$

$$\sum_{\substack{c=1 \\ T \in \mathcal{T}(c)}}^{\mathcal{C}_{max}} \sum_{\ell=1}^N X_{c,\ell} = flot(T) \quad T = 1, \dots, |\mathcal{T}| \quad (7.1)$$

$$\sum_{c=1}^{\mathcal{C}_{max}} X_{c,\ell} = 1 \quad \ell = 1, \dots, N \quad (7.2)$$

$$\sum_{c=1}^{\mathcal{C}_{max}} G_c^{arr} \cdot X_{c,\ell} - \sum_{c=1}^{\mathcal{C}_{max}} G_c^{dep} \cdot X_{c,(\ell \bmod N)+1} = 0 \quad \ell = 1, \dots, N \quad (7.3)$$

$$\begin{aligned} \sum_{c=1}^{\mathcal{C}_{max}} D_c^{deb} \cdot X_{c,(\ell \bmod N)+1} + L - \sum_{c=1}^{\mathcal{C}_{max}} D_c^{fin} \cdot X_{c,\ell} &\geq M \quad \ell = 1, \dots, N \\ X_{c,\ell} &\in \{0, 1\} \end{aligned} \quad (7.4)$$

Une tâche T est couverte si $\text{flot}(T)$ engins y sont affectés : les contraintes (7.1) assurent que pour chaque tâche T on a bien $\text{flot}(T)$ chemins qui passent par l'arc-tâche associé. Les contraintes (7.2) garantissent qu'à chaque ligne de roulement (engin) est affecté un et un seul chemin. Les contraintes (7.3) permettent de s'assurer que pour toute ligne ℓ , la gare d'arrivée du chemin affecté à la ligne ℓ est la même que la gare de départ du chemin affecté à la ligne suivante ($\ell+1$ si $\ell \leq N-1$, 1 si $\ell = N$). De la même façon, les contraintes (7.4) garantissent que pour toute ligne ℓ , l'intervalle de temps entre la date d'arrivée du chemin affecté à ℓ et la date de départ du chemin affecté à la ligne suivante est supérieur à la marge minimum obligatoire M (cf. section 1.1.3).

7.3 Méthode de résolution

Pour résoudre P_{CGC} , il faut connaître tous les chemins de la source au puits dans le graphe décrit à la section 7.2.1, on obtient ainsi un nombre exponentiel de variables. Afin de pouvoir traiter des instances de taille réelle, on choisit ici de considérer uniquement un sous-ensemble de chemins permettant à la fois de trouver une solution admissible (respect des flots et de la cyclicité) et d'obtenir des résultats intéressants en termes de robustesse (critère *invret*). Dans cette optique, nous décidons d'utiliser la technique de génération de colonnes combinée à une méthode par arrondi pour résoudre le problème de façon heuristique.

7.3.1 Principe : étapes de résolution

Procédure de génération de colonnes

Afin de résoudre P_{CGC} (présenté à la section 7.2.4), on cherche d'abord à résoudre sa relaxation continue à l'aide de la technique de génération de colonnes. Dans ce cadre, le problème initial est décomposé en sous-problèmes :

- le problème maître correspond à la relaxation continue de P_{CGC} , restreinte aux colonnes d'une solution initiale à définir ;
- le sous-problème consiste à trouver des colonnes de coût réduit négatif, afin d'améliorer l'objectif du problème maître.

Les colonnes ici sont les variables $X_{c,\ell}$. Une colonne concerne donc une paire associant un chemin c et une ligne de roulement (ou engin) ℓ . Soit \mathcal{C}_{PRESTO} l'ensemble des chemins construits arbitrairement lors de la Phase 2 de PRESTO (cf. section 1.2.3). L'ensemble initial de colonnes est défini de la façon suivante :

$$\{X_{c,\ell} | \forall c \in \mathcal{C}_{PRESTO}, \forall \ell \in \llbracket 1, N \rrbracket\}$$

Par la suite, on notera \mathcal{C} l'ensemble des chemins considérés dans le problème maître, c'est-à-dire l'ensemble des chemins c tels qu'il existe au moins une colonne $X_{c,\ell}$ dans le problème maître. À toute itération, \mathcal{C} est tel que $\mathcal{C}_{PRESTO} \subset \mathcal{C} \subset \mathcal{C}_{max}$.

La génération de colonnes consiste alors en une procédure itérative, dont chaque itération est décomposée en plusieurs étapes :

1. résolution du problème maître (avec son nombre restreint de colonnes) ;
2. récupération des valeurs duales associées ;
3. résolution du sous-problème : recherche d'une colonne X_{c^*,ℓ^*} de plus faible coût réduit ;

4. si le coût réduit calculé de X_{c^*, ℓ^*} est négatif : ajout des colonnes $X_{c^*, \ell}$ ($\forall \ell \in \llbracket 1, N \rrbracket$) au problème maître (et retour en 1.), sinon arrêt de la procédure.

Au cours de la procédure, N colonnes sont ajoutées au problème maître à chaque itération. Aucune colonne n'est supprimée.

Lorsque la procédure est terminée, on obtient un PL restreint à un sous-ensemble de variables permettant d'obtenir une solution optimale de la relaxation continue du PLNE initial décrit à la section 7.2.4.

Résolution entière

Lorsque le plus faible coût réduit est positif ou nul, on considère qu'on est à l'optimum de la relaxation continue à la racine. On cherche alors une solution entière.

D'après des tests préliminaires, nous avons pu faire quelques observations :

- la résolution directe du PLNE restreint aux colonnes générées ne garantit pas de trouver une solution entière (alors qu'il en existe une) ;
- le calcul d'une solution entière par résolution directe peut être très long et engendrer des problèmes de mémoire ;
- la solution de la relaxation continue obtenue est très fractionnaire.

Nous avons décidé d'utiliser une méthode heuristique par arrondi en fixant progressivement les variables. À chaque itération, on sélectionne un chemin c qu'on affecte à une ligne ℓ (variable $X_{c, \ell}$) :

1. la variable $X_{c, \ell}$ dont la valeur dans la solution continue est la plus proche de 1 est fixée à 1 ;
2. la procédure de génération de colonnes est relancée sur le problème restreint aux colonnes générées et pour lequel on a fixé une variable ;
3. si la solution obtenue en fin de génération de colonnes est entière, on a terminé, sinon on fixe une nouvelle variable (retour en 1.).

Lorsqu'on a terminé cette méthode par arrondi, on a trouvé une solution entière, qui correspond à un roulement d'engins cyclique sans maintenance.

Le schéma de la Figure 7.2 reprend les principales étapes de la méthode de résolution mise en œuvre remplaçant les Phases 2 et 3 de PRESTO.

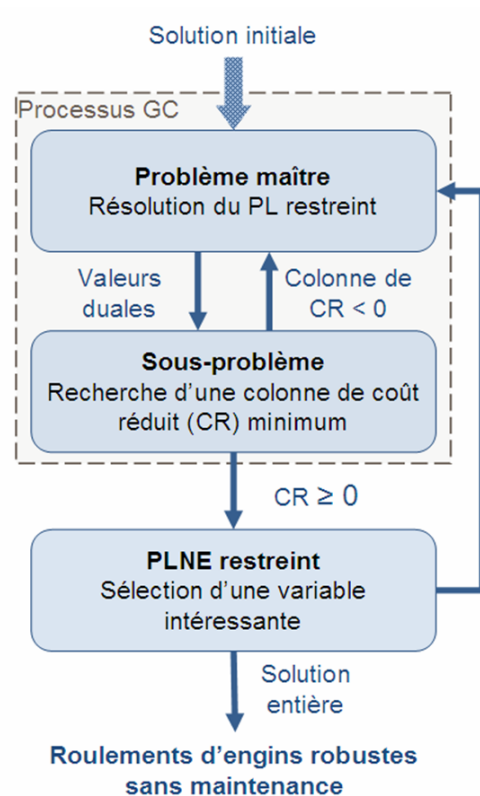


FIGURE 7.2 – Fonctionnement de l’approche (CGC) remplaçant les Phases 2 et 3 de PRESTO.

La méthode de résolution est résumée ci-dessous :

Algorithme 3 Méthode de résolution de l’approche (CGC)

```

tant que on n’a pas de solution entière faire
  // Processus de génération de colonnes
  tant que il existe des colonnes de coût réduit négatif faire
    Résolution du problème maître
    Récupération des valeurs duales
    Résolution du sous-problème : recherche d’une colonne  $X_{c^*,\ell^*}$  de coût réduit minimal
    si Le coût réduit est négatif alors
      Ajout des colonnes  $X_{c^*,\ell}$  ( $\forall \ell = 1, \dots, N$ ) au problème maître
    sinon
      Il n’existe plus de colonnes de coût réduit négatif
    fin si
  fin tant que
  Sélection de la variable de plus haute valeur (fixée à 1)
fin tant que

```

Existence d’une solution entière admissible

Puisqu’on fixe certaines variables pour résoudre le problème en entiers, il faut s’assurer qu’on obtient toujours une solution entière admissible en fin de procédure. Une solution entière est admissible si elle consiste en N chemins de S à P dans le graphe couvrant les tâches, c’est-à-dire respectant la solution de flots et les contraintes de cyclicité (c’est-à-dire avec le même nombre de

chemins partant et arrivant en chaque gare).

Soit $G = (V, E)$ le graphe issu de PRESTO présenté à la section 1.2.1.

On sait que

$$\forall v \in V - \{S, P\}, \sum_{a \in In(v)} flot(a) = \sum_{a \in Out(v)} flot(a).$$

Fixer une variable $X_{c,\ell}$ à 1 revient à modifier le graphe G pour obtenir un nouveau graphe $G' = (V', E')$ dans lequel le flot de chaque arc appartenant au chemin c est diminué de 1. On note $v' \in V'$ le sommet de G' correspondant à un sommet $v \in V$ et $a' \in E'$ l'arc de G' correspondant à un arc $a \in E$. Alors

$$\forall a \in c, flot(a') = flot(a) - 1.$$

On en déduit l'égalité suivante :

$$\forall v' \in V' - \{S', P'\}, \sum_{a \in In(v')} flot(a) = \sum_{a \in Out(v')} flot(a).$$

Lorsqu'on fixe des variables, la propriété de conservation des flots en tout sommet hors S et P est toujours respectée. Alors tant qu'il existe des arcs de flot non nul, il existe au moins un chemin de la source au puits dans le graphe.

De plus, la solution de flots est telle que la somme des flots sur les arcs-sources est égale au nombre d'engins calculé, soit

$$\sum_{a \in Out(S)} flot(a) = N$$

Lorsqu'on a fixé N variables, on a donc diminué cette somme de N , il n'y a plus d'arc-source de flot non nul. À cause de la propriété de conservation des flots dans le graphe, il est alors impossible qu'il existe d'autres arcs non couverts (de flot non nul).

Fixer N variables revient donc à sélectionner N chemins de S à P respectant les flots de la solution de flots de PRESTO.

On sait aussi qu'en respectant cette solution de flots, les chemins sont tels qu'en chaque gare, il existe le même nombre d'engins en début et en fin d'horizon temporel.

Cependant, en fixant certaines variables, on remet en cause l'hypothèse de connexité du graphe Gr_{AR}^p et on risque de ne plus garantir de trouver un roulement droit. Pour trouver une solution entière dans tous les cas, les contraintes (7.3) et (7.4) d'enchaînement des lignes (cf. P_{CGC} à la section 7.2.4) sont relâchées par l'ajout de variables fortement pénalisées dans l'objectif. Si ces variables sont nulles dans la solution entière, on obtient un roulement droit cyclique et on peut passer à la Phase 4 de PRESTO pour ajouter la maintenance. Sinon, la Phase 3 de PRESTO est utilisée pour l'enchaînement des lignes.

7.3.2 Expression du sous-problème

Le sous-problème consiste à trouver une colonne X_{c^*,ℓ^*} de plus faible coût réduit, afin d'ajouter le sous-ensemble de colonnes $\{X_{c^*,\ell}, \forall \ell = 1, \dots, N\}$ au problème maître. Dans ce but, on associe à chaque contrainte de P_{CGC} une variable duale. On remarque que comme il existe un nombre polynomial de contraintes dans P_{CGC} (indexées sur les lignes d'indice ℓ ou les tâches d'indice T données en entrée), il existe un nombre polynomial de variables duales :

- la variable duale y_T^1 correspond à la contrainte (7.1), $\forall T = 1, \dots, |\mathcal{T}|$;
- la variable duale y_ℓ^2 correspond à la contrainte (7.2), $\forall \ell = 1, \dots, N$;
- la variable duale y_ℓ^3 correspond à la contrainte (7.3), $\forall \ell = 1, \dots, N$;

– la variable duale y_ℓ^4 correspond à la contrainte (7.4), $\forall \ell = 1, \dots, N$.

Soit X une solution de base optimale du problème maître et y la solution duale correspondante. Alors pour tout $c \in \{1, \dots, \mathcal{C}\}$ et pour tout $\ell \in \{1, \dots, N\}$, le coût réduit $\Delta_{c,\ell}$ de la variable $X_{c,\ell}$ est le suivant :

$$\begin{aligned} \Delta_{c,\ell} &= \text{invret}(c) - \left(\sum_{T \in \mathcal{T}(c)} y_T^1 \right) - y_\ell^2 \\ &+ G_c^{dep} \cdot y_{\ell-1}^3 - G_c^{arr} \cdot y_\ell^3 + D_c^{fin} \cdot y_\ell^4 - D_c^{deb} \cdot y_{\ell-1}^4 \end{aligned}$$

L'indice $\ell-1$ correspond à l'indice de la ligne précédant la ligne ℓ . Pour $\ell = 1$, la ligne précédente est la ligne N . Dans l'expression de $\Delta_{c,1}$, il faut donc remplacer $\ell-1$ par N , soit :

$$\begin{aligned} \Delta_{c,1} &= \text{invret}(c) - \left(\sum_{T \in \mathcal{T}(c)} y_T^1 \right) - y_1^2 \\ &+ G_c^{dep} \cdot y_N^3 - G_c^{arr} \cdot y_1^3 + D_c^{fin} \cdot y_1^4 - D_c^{deb} \cdot y_N^4 \end{aligned}$$

Le problème consiste alors à trouver un couple (c^*, ℓ^*) de coût minimum :

$$\Delta_{c^*, \ell^*} = \min_{c \in \mathcal{C}, \ell \in [1, N]} (\Delta_{c,\ell})$$

7.3.3 Modélisation du sous-problème

Principe

On duplique le graphe Gr^p (décrit à la section 2.1 et dont un exemple est donné en Figure 2.1) de façon à obtenir N graphes identiques correspondant aux N lignes de roulement, ce qui permet de faire correspondre à chaque variable $X_{c,\ell}$ un chemin c dans un graphe ℓ . Ainsi, il y a bijection entre l'ensemble des variables de décision et l'ensemble des chemins dans les N graphes.

On affecte des poids aux arcs des graphes de façon à ce que le poids d'un chemin c dans un graphe ℓ corresponde au coût réduit de la variable $X_{c,\ell}$. La recherche d'une variable de coût réduit minimal est alors équivalente à la recherche d'un chemin optimal (de poids minimal) dans ces graphes. L'affectation de poids aux arcs est détaillée ci-après.

Poids d'un chemin dans un graphe

Affectation de poids aux arcs Afin de résoudre le sous-problème, on cherche à affecter des poids aux arcs des graphes, de façon à retrouver le coût $\Delta_{c,\ell}$ en sommant les poids sur les arcs du chemin c dans le graphe ℓ . Les différents arcs du graphe sont définis à la section 1.2.1.

À chaque arc est affecté un poids fixe p_f tel que :

- pour tout arc-tâche T , $p_f(T) = -y_T^1$;
- pour tout arc-source $a_s = (S, G^{dep})$:

$$p_f(a_s) \begin{cases} = -y_\ell^2 + G^{dep} \cdot y_N^3 & \text{si } \ell = 1. \\ = -y_\ell^2 + G^{dep} \cdot y_{\ell-1}^3 & \text{sinon.} \end{cases}$$

- pour tout arc-puits $a_p = (G^{arr}, P)$, $p_f(a_p) = -G^{arr} \cdot y_\ell^3$;
- pour tout arc-attente a , $p_f(a) = 0$.

Par ailleurs, sur chaque arc-tâche T_i , on ajoute deux poids p_1 et p_2 optionnels qui ne sont pris en compte que si T_i est la première ou la dernière tâche d'un chemin :

- $p_1(T_i) = -D_i^{deb} \cdot y_{\ell-1}^4$ ($= -D_i^{deb} \cdot y_N^4$ si $\ell = 1$) correspond à la date de début de T_i et est ajouté au poids d'un chemin si T_i est le premier arc-tâche du chemin ;
- $p_2(T_i) = D_i^{fin} \cdot y_\ell^4$ correspond à la date de fin de T_i et est ajouté au poids d'un chemin si T_i est le dernier arc-tâche du chemin.

Enfin, sur chaque arc-attente et sur chaque arc-connexion $a_i = ((G_i^{dep}, D_i^{deb}), (G_i^{arr}, D_i^{fin}))$ est ajouté un poids $p_t(a) = D_i^{fin} - D_i^{deb}$ correspondant à la durée de l'arc a_i .

Le poids d'un chemin c dans un graphe ℓ devant correspondre au coût réduit de la variable $X_{c,\ell}$, il faut pouvoir calculer $invret(c)$ en parcourant le chemin. Or il s'agit de la somme des inverses des temps de retournement du chemin. Le temps de retournement entre deux tâches qui se suivent est la somme des durées des arcs-attentes et des arcs-connexions entre ces tâches, c'est-à-dire la somme des poids p_t de ces arcs.

Soient deux tâches T_i et T_j se suivant sur le chemin c et (a_1, a_2, \dots, a_n) le sous-chemin entre les deux tâches. Le temps de retournement entre T_i et T_j s'écrit $t_{ij} = \sum_{r=1}^n p_t(a_r)$. Il faut alors comptabiliser son inverse $\frac{1}{t_{ij}}$ dans le poids du chemin c .

La Figure 7.3 reprend tous les poids affectés aux différents arcs d'un graphe G_ℓ . L'exemple présenté consiste en trois tâches T_1, T_2 et T_3 allant de la gare G_1 à la gare G_2 ou de la gare G_2 à la gare G_1 . Pour des questions de lisibilité, les arcs-connexions, qui se comportent comme des arcs-attentes, ne sont pas représentés sur le graphe, et les poids $p_t(a)$ sont notés p_{t1} à p_{t4} .

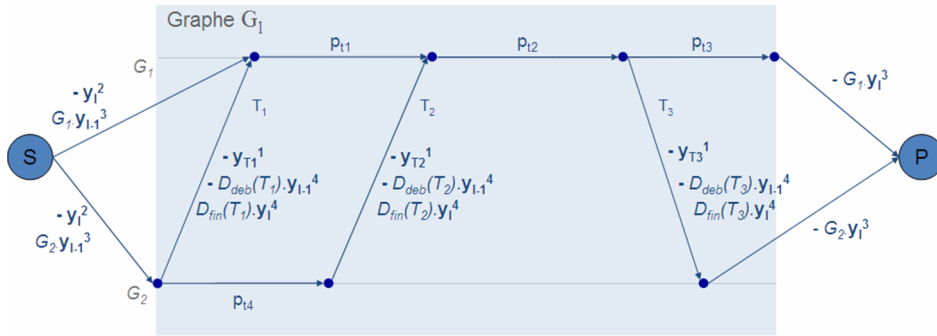


FIGURE 7.3 – Répartition des poids sur le graphe associé à la ligne de roulement ℓ .

Calcul du poids d'un chemin Soit un chemin c de S à P dans un graphe ℓ , de première tâche T_p et de dernière tâche T_d . Le coût total du chemin est alors le suivant :

$$invret(c) + p_1(T_p) + p_2(T_d) + \sum_{a \in c} p_f(a)$$

La Figure 7.4 représente un chemin c dans le graphe de la Figure 7.3. Le poids de c est alors le suivant :

$$\frac{1}{p_{t1} + p_{t2}} + G_2 \cdot y_{\ell-1}^3 - y_{\ell}^2 - D^{deb}(T_1) \cdot y_{\ell-1}^4 + D^{fin}(T_2) \cdot y_{\ell}^4 - G_2 \cdot y_{\ell}^3$$

avec $\frac{1}{p_{t1} + p_{t2}} = invret(c)$. En effet, les tâches T_1 et T_3 se suivent sur le chemin c . Il faut donc comptabiliser l'inverse du temps de retournement entre les deux tâches. Le temps de retournement est la somme des durées des arcs-attentes et des arcs-connexions entre T_1 et T_2 , soit $p_{t1} + p_{t2}$.

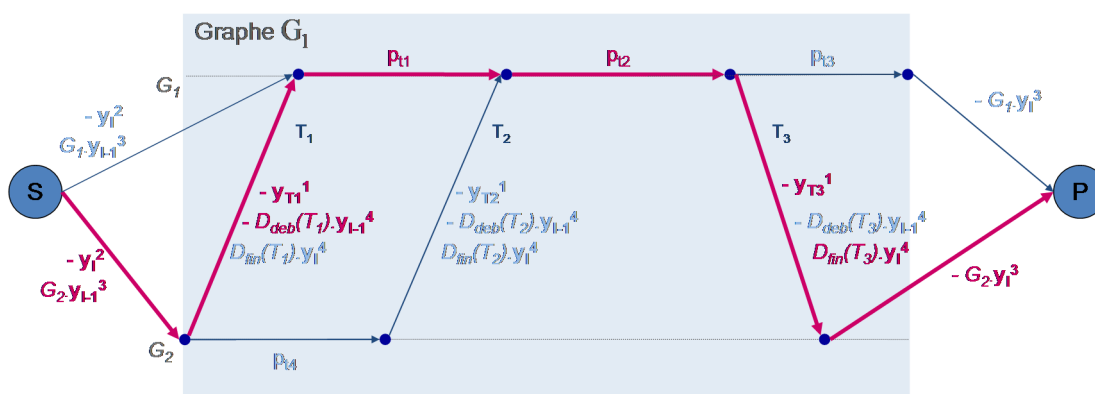


FIGURE 7.4 – Poids d'un chemin c dans le graphe associé à la ligne de roulement ℓ .

On a bien correspondance entre le poids d'un chemin c dans un graphe ℓ et le coût réduit de la variable $X_{c,\ell}$. Afin de trouver une variable $X_{c,\ell}$ de coût réduit $\Delta_{c,\ell}$ minimal, on cherche donc un chemin optimal, c'est-à-dire de coût minimal, de S à P dans un des N graphes.

7.3.4 Résolution du sous-problème

Particularités de la recherche de chemin optimal

Le poids d'un chemin c dans un graphe G_ℓ , correspondant au coût réduit $\Delta_{c,\ell}$ de la variable $X_{c,\ell}$ peut être décomposé en trois catégories de poids différentes :

- les poids fixes : lorsqu'on ajoute un arc a_i au chemin c , on ajoute $p_f(a_i)$ au poids du chemin ;
- les poids optionnels : lorsqu'on ajoute un arc-tâche T_i au chemin c , on ajoute au poids du chemin $p_1(T_i)$ si T_i est son premier arc-tâche (0 sinon) et $p_2(T_i)$ si T_i est son dernier arc-tâche (0 sinon) ;
- les poids "quasi cumulatifs" : lorsqu'on ajoute un arc a_i au chemin c , on n'ajoute pas directement $\frac{1}{p_r(a_i)}$ au poids du chemin. On mémorise la dernière tâche T_j effectuée. Lorsqu'on circule sur la tâche suivante T_k , on ajoute l'inverse du temps de retournement entre T_j et T_k . Si a_i est le seul arc entre les deux tâches, le temps de retournement correspond à sa durée (on ajoute alors $\frac{1}{p_r(a_i)}$ à la valeur de $invret$). Sinon, le temps de retournement est la somme des durées des arcs entre les deux tâches (dont a_i fait partie).

En présence de poids optionnels et quasi cumulatifs, la recherche d'un chemin optimal ne s'apparente pas à la recherche d'un plus court chemin classique. L'utilisation d'un algorithme de recherche

de plus court chemin pour résoudre le sous-problème risque alors d'être sous-optimale.

Exemple

On considère le graphe \mathcal{G}_ℓ de la Figure 7.5, dans lequel on cherche un chemin optimal. Tous les poids non renseignés sont nuls.

On applique un algorithme de recherche de plus court chemin classique entre S et P . Au sommet v_1 , il existe deux chemins possibles de S à v_1 :

- un chemin de poids -5 passant par T_1 ;
- un chemin de poids nul passant par a_1 .

Comme on cherche un plus court chemin, on conserve le chemin passant par T_1 . En parcourant le reste du graphe jusqu'au puits P , le plus court chemin est le chemin c passant par T_2 et de poids

$$p_1(T_1) + \frac{1}{p_t(a_2)+p_t(a_3)} + p_2(T_2) = -5,9$$

($p_1(T_2)$ n'est pas pris en compte car T_2 n'est pas le premier arc-tâche du chemin).

Or il existe un chemin c' passant par a_1 et T_2 de poids $p_1(T_2) + p_2(T_2) = -7$ (comme il n'y a qu'une tâche, il n'y a pas de temps de retournement). Le poids de c' est inférieur au poids du chemin c . L'algorithme de plus court chemin est donc sous-optimal.

Par ailleurs, on considère un chemin de S à v_2 en passant par T_1 . Au sommet v_2 , on ignore quel sera le temps de retournement. On ne peut donc pas encore considérer le poids $p_t(a_2)$. On peut compter le temps de retournement uniquement après avoir emprunté un nouvel arc-tâche (au sommet v_3 dans l'exemple présenté) ou lorsqu'on a atteint le puits.

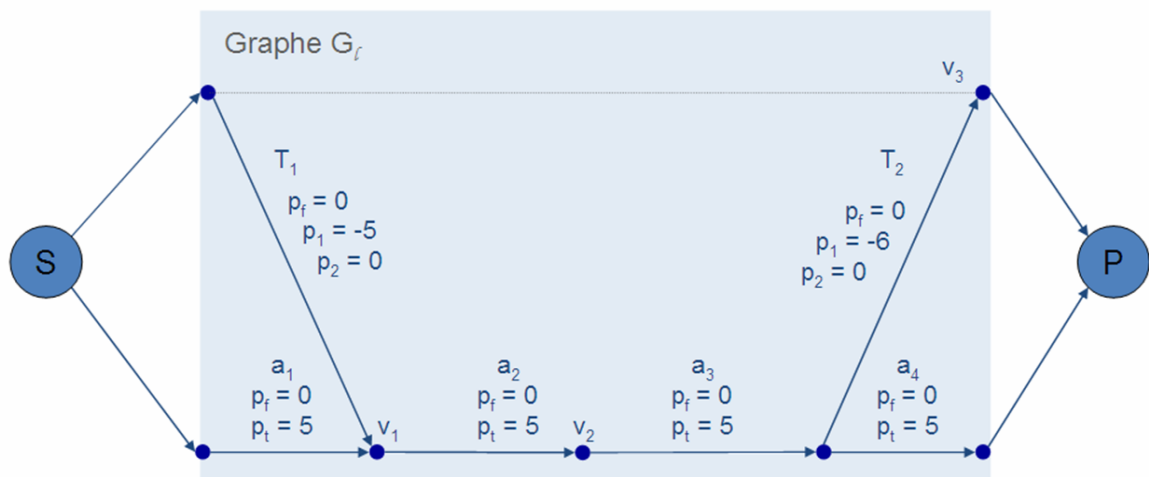


FIGURE 7.5 – Exemple de recherche de chemin sous-optimale avec un algorithme classique de plus court chemin.

Mise en place d'un système d'étiquettes

On met en place un système d'étiquettes pour palier les difficultés liées aux poids optionnels et quasi cumulatifs.

En chaque sommet v , on conserve trois listes d'étiquettes correspondant à trois types de chemin :

- la liste $L_0(v)$ des étiquettes e_0^i correspondant à des chemins de S à v sans arc-tâche (aucun poids $p_1(T)$ ou $p_2(T)$ n'a été pris en compte);
- la liste $L_1(v)$ des étiquettes e_1^i correspondant à des chemins de S à v avec au moins un arc-tâche, mais en considérant qu'il y aura d'autres arcs-tâches entre v et P (le poids $p_1(T)$ a été pris en compte pour le premier arc-tâche T);
- la liste $L_2(v)$ des étiquettes e_2^i correspondant à des chemins de S à v en considérant que tous les arcs-tâches sont déjà dans le chemin partiel, c'est-à-dire qu'il n'y aura aucun arc-tâche de v à P (les poids $p_1(T_p)$ et $p_2(T_d)$ ont été pris en compte pour le premier arc-tâche T_p et le dernier T_d).

De plus, on considère la somme des inverses des temps de retournement $invret(c)$ comme une contrainte de ressource à minimiser. Lors du parcours du graphe, le poids $poids$ correspondant aux poids fixes et optionnels et la valeur de $invret$ sont calculés indépendamment, et ne sont agrégés qu'au sommet puits P . On a en chaque sommet v des étiquettes correspondant à des chemins de S à v .

On définit alors une étiquette e^i en un sommet $v \in V$ comme un couple de réels $(poids^i, invret^i)$. Elle correspond à un chemin de S à v dont $poids^i$ est la somme des poids fixes et optionnels et $invret^i$ la valeur du critère $invret$ du chemin.

Dans la suite, on considère N graphes $G_\ell = (V, E)$ de source S et de puits P pour toute ligne de roulement ℓ allant de 1 à N .

Algorithme de résolution du sous-problème

On cherche un chemin optimal pour chaque graphe G_ℓ (ℓ allant de 1 à N).

Pour chaque graphe, les sommets sont traités dans l'ordre chronologique, de façon à ce que tous les prédécesseurs du sommet courant aient déjà été traités. Le premier sommet traité est donc le nœud source S , pour lequel il n'existe qu'une étiquette e_0 initialisée à $(0, 0)$ correspondant à un chemin optimal sans tâche (de S à S).

On parcourt ensuite tous les sommets v tels que $v \in V \setminus \{P\}$. Les étiquettes en un sommet sont calculées d'après les étiquettes des prédécesseurs. En un sommet v , chaque étiquette de chaque prédécesseur v' permet de créer une voire plusieurs étiquettes : par exemple, si on passe par une tâche, chaque étiquette de la liste $L_1(v')$ étendue à v donnera lieu à une étiquette dans $L_1(v)$ (chemin partiel de S à v contenant au moins une tâche) et à une étiquette dans $L_2(v)$ (chemin partiel de S à v contenant la dernière tâche du futur chemin complet de S à P).

Afin d'éviter une croissance exponentielle du nombre d'étiquettes en chaque sommet, on applique des règles de dominance permettant de supprimer à l'avance les étiquettes qui ne correspondront pas à un chemin optimal. On ne compare des étiquettes que si elles appartiennent à la même liste.

En fin de parcours, le sommet puits est traité : on construit uniquement la liste $L_2(P)$ puisque les chemins sont tous complets. On choisit ensuite parmi toutes les étiquettes $e_2^i \in L_2(P)$ celle qui minimise la quantité $poids^i + invret^i$ correspondant à $\Delta_{c_i, \ell}$, soit :

$$\min_{e_i^i \in L_2(P)} (poids^i + invret^i)$$

S'il s'agit du premier graphe traité, le chemin c_i (correspondant à l'étiquette retenue) est construit en parcourant les sommets du puits à la source, et conservé comme solution courante.

S'il y a déjà une solution courante de coût $\Delta_{c_j, \ell}$, on ne construit le chemin c_i que si $\Delta_{c_i, \ell} < \Delta_{c_j, \ell}$. Sinon, la solution courante est conservée.

En fin d'algorithme, on obtient un chemin c optimal de coût $\Delta_{c_j, \ell}$.

Les étapes de l'algorithme utilisé pour la résolution du sous-problème sont rappelées ci-dessous :

Algorithme 4 Résolution du sous-problème

```
// Initialisation de la meilleure solution courante
 $\Delta_{c^*, \ell^*} = 0$ 
pour tout  $\ell = 1, \dots, N$  faire
  // Recherche du chemin optimal dans le graphe  $G_\ell = (V, E)$ 

  Tri des sommets dans l'ordre chronologique
  Initialisation du sommet source  $S$  : une seule étiquette  $e = (0, 0) \in L_0$ 

  // Parcours des sommets
  pour tout  $v \in V - \{S, P\}$  faire
    // Pour chaque prédécesseur
    pour tout  $v', (v', v) \in E$  faire
      Extension des étiquettes du sommet  $v'$  au sommet  $v$ 
    fin pour
  fin pour

  // Traitement de  $P$ 
  Pour tout prédécesseur de  $P$ 
  pour tout  $v, (v, P) \in E$  faire
    Extension des étiquettes de la liste  $L_2(v')$  à  $P$ 
  fin pour

  // Calcul du coût réduit le plus faible à partir des étiquettes de  $L_2(P)$ 
   $\Delta_{c^{*'}, \ell} = \min_{e_i^i \in L_2(P)} (poids^i + invret^i)$ 

  // Mise à jour de la solution courante
  si  $\Delta_{c^{*'}, \ell} < \Delta_{c^*, \ell^*}$  alors
    Construction du chemin en parcourant les sommets de  $P$  à  $S$ 
     $\Delta_{c^*, \ell^*} = \Delta_{c^{*'}, \ell}$ 
     $c^* = c^{*'}$ 
     $\ell^* = \ell$ 
  fin si
fin pour
```

Optimalité du chemin calculé

Dans le sous-problème, il n'est pas utile de trouver un chemin optimal, une colonne de coût réduit négatif est suffisante pour continuer la procédure de génération de colonnes. En revanche, l'optimalité du chemin calculé est importante en fin de procédure, pour garantir qu'il n'existe plus de chemins de poids négatif, c'est-à-dire de colonnes de coût réduit négatif. On cherche ici à s'assurer que le chemin calculé par l'algorithme de programmation dynamique précédent est bien le

chemin optimal (de poids minimal). Dans ce but, les règles de dominance appliquées ne permettent de supprimer que les étiquettes qui ne pourront pas correspondre à un chemin optimal.

7.4 Premiers résultats et conclusions

Dans cette partie, on cherche à valider le modèle et la méthode de résolution basée sur une technique de génération de colonnes. On va donc tester la méthode sur les instances cibles présentées à la section 5.6 et comparer les solutions obtenues avec l'existant.

7.4.1 Tests sur les instances cibles

Avant d'appliquer l'approche (CGC), on cherche à résoudre P_{CGC} directement pour des instances fictives de petite taille. Ces premiers tests révèlent la nécessité d'une méthode de résolution plus efficace. Nous ajustons alors la méthode (CGC) à partir des premières expérimentations, puis nous présentons les résultats sur les instances cibles.

Volumétrie de P_{CGC}

Une méthode dite explicite générant l'ensemble des variables $X_{c,\ell}$ a été mise en œuvre sur des instances fictives de petite taille : de 3 à 6 engins pour 8 à 45 tâches couvertes. La plus grande instance pour laquelle nous obtenons une solution est une instance à 8 engins couvrant 28 tâches. La résolution directe de P_{CGC} pour les instances plus grandes sature la mémoire de nos moteurs de calcul.

Des tests complémentaires ont été réalisés afin de résoudre l'instance fictive à 28 tâches par la méthode (CGC). Des éléments de volumétrie (après l'étape de presolve du solveur IBM ILOG CPLEX), de calcul et de résultats sont renseignés dans la Table 7.1 :

- nombre de chemins considérés dans le problème (pour (CGC), il s'agit du nombre de chemins en fin de procédure) ;
- nombre de variables (pour (CGC), il s'agit du nombre de colonnes en fin de procédure) ;
- nombre de contraintes ;
- temps de calcul (incluant les phases de PRESTO) ;
- valeur du critère *invret* ;
- gain en robustesse (sur *invret* par rapport à PRESTO).

Méthode	PRESTO	Méthode explicite	CGC
Nombre de chemins	-	698	60
Nombre de variables	-	5584	473
Nombre de contraintes	-	51	51
Temps d'exécution total (en secondes)	1	25	2
Valeur de <i>invret</i>	0,3841	0,2021	0,2036
Gain sur <i>invret</i>	-	47,4%	47,0%

TABLE 7.1 – Comparaison de la méthode explicite résolvant P_{CGC} via IBM ILOG CPLEX et de la méthode (CGC) sur une instance fictive à 28 tâches

Le nombre de variables de P_{CGC} est exponentiel et à partir d'un certain nombre de tâches à couvrir, la résolution du problème sature la mémoire de nos moteurs de calcul. En comparaison, la

méthode (CGC) permet de générer un nombre beaucoup plus faible de variables : dans l'exemple à 28 tâches présenté dans la Table 7.1, P_{CGC} contient plus de 5500 variables, tandis que la méthode (CGC) génère moins de 500 colonnes. De plus, la méthode fournit une solution proche de l'optimal en un temps beaucoup plus court.

La méthode de résolution utilisant la génération de colonnes paraît donc intéressante pour l'approche (CGC).

Ajustements de la méthode

Les premiers essais de l'approche (CGC) sur des instances réelles ont été peu concluants : le calcul, en plus d'être très long (parfois plusieurs jours sans résultats), saturait la mémoire des moteurs de calcul utilisés. Après analyse des solutions, nous avons constaté qu'après un certain nombre d'itérations, le plus faible coût réduit calculé était parfois très proche de 0 mais restait négatif, ajoutant alors indéfiniment la même variable.

Le critère *invret* dans l'objectif prend des valeurs fractionnaires arrondies différemment dans l'outil et par IBM ILOG CPLEX (la précision par défaut n'est pas la même), générant des erreurs de calcul. Nous avons alors défini une pénalité 10^6 au critère et tronqué les valeurs afin de limiter le nombre de décimales. De plus, une précision a été introduite lors de l'évaluation des coûts réduits. Ainsi, le critère d'arrêt de la procédure de génération de colonnes est devenu : coûts réduits < -0.0001 .

Ensuite, de nouveaux tests montrant un temps de calcul encore très long, nous avons défini un nouveau critère d'arrêt : si après 10000 itérations il existe encore des variables de coût réduit négatif, la procédure de génération de colonnes prend fin. Sinon, elle prend fin avant lorsque le plus faible coût réduit calculé est positif ou nul.

Enfin, les colonnes de la solution initiale ne permettant pas toujours de construire un roulement droit, nous avons dû relâcher les contraintes d'enchaînement des lignes par l'ajout de variables de relâchement pénalisées dans l'objectif. L'étape d'enchaînement des lignes de PRESTO (Phase 3) est alors utilisée afin de gérer les cas où on n'obtient pas un roulement droit en sortie de (CGC).

Résultats de l'approche (CGC) sur les instances cibles

On compare ici les résultats obtenus par PRESTO et par la méthode (CGC) sur les instances cibles présentées à la section 5.6. La maintenance n'est pas prise en compte dans ces tests. Pour chaque test, on renseigne les éléments suivants dans la Table 7.2 :

- le temps de calcul total (incluant toutes les phases de résolution) ;
- la valeur du critère *invret* ;
- le gain sur *invret* par rapport à PRESTO ;
- le gap entre l'objectif de la solution continue obtenue lors de la première procédure de génération de colonnes et l'objectif entier de la solution entière obtenue.

Instance	Instance 1		Instance 2		Instance 3	
	PRESTO	CGC	PRESTO	CGC	PRESTO	CGC
Temps d'exécution total (en secondes)	300	720	60	4980	60	360
Valeur de <i>invret</i>	9,18	7,92	10,52	7,28	117,60	117,10
Gain sur <i>invret</i>	-	13,7%	-	30,8%	-	0,4%
Gap	-	0%	-	0%	-	0%

TABLE 7.2 – Premiers résultats de PRESTO et de (CGC) sur les instances cibles

Le gain sur *invret* varie entre près de 0% et plus de 30%.

Le gap entre l'objectif de la solution continue obtenue par résolution de la relaxation continue de P_{CGC} par génération de colonnes¹ et l'objectif entier en fin de méthode est nul. Après étude des solutions continues, on constate des valeurs des variables $X_{c,\ell}$ très fractionnaires.

Le temps d'exécution de (CGC) est variable en fonction des instances, puisque pour l'instance 1 la méthode (CGC) nécessite 2 minutes de plus que PRESTO, tandis que pour l'instance 2, il faut plus d'une heure.

7.4.2 Premières conclusions

De même que pour l'approche (HGL) décrite au Chapitre 6, on traite ici un problème très contraint, permettant d'obtenir une solution à coûts de production optimaux. On rappelle les données imposées issues de la résolution du P_{PRESTO} de PRESTO :

- le nombre d'engins utilisés ;
- le nombre d'engins circulant sur chaque tâche (flots) ;
- les sillons à vide (dates, OD et nombre d'engins de chaque type circulant).

L'utilisation de la génération de colonnes pour résoudre P_{CGC} fournit des solutions intéressantes sur le critère *invret* par rapport à PRESTO en un temps raisonnable pour des instances réelles de plusieurs centaines de tâches, tandis qu'une résolution directe sature la mémoire pour de petites instances fictives.

Les solutions des tests présentés correspondent à des roulements droits sans maintenance. L'ajout de la maintenance par la Phase 4 de PRESTO ne garantit plus un roulement droit et dégrade les critères de robustesse (cf. section 6.4.1).

7.4.3 Industrialisation et perspectives

De la même façon que l'approche (HGL), l'approche (CGC) consiste en un module complètement intégré à PRESTO, présentant ainsi certains avantages en cas d'industrialisation.

Par ailleurs, aucun paramétrage préalable par l'utilisateur n'est nécessaire, ce qui facilite l'utilisation du module.

Mais le temps d'exécution et les ressources nécessaires en termes de mémoire ne permettent pas une industrialisation en l'état. Les performances devraient donc être améliorées avant une utilisation industrielle.

Plusieurs axes d'amélioration de la méthode sont envisageables : l'amélioration de la méthode de résolution, l'ajout de critères de robustesse et l'intégration de la maintenance.

Dans le but d'améliorer la méthode de résolution, on peut s'intéresser aux paramètres de IBM ILOG CPLEX, à l'algorithme de recherche d'une colonne de coût réduit négatif (résolution du sous-problème) ou à l'heuristique de résolution entière. Par exemple, l'algorithme de résolution du sous-problème peut être décomposé en une heuristique permettant de trouver rapidement une bonne solution et un algorithme exact en cas de coût réduit positif.

Plusieurs démarches peuvent être adoptées pour intégrer la maintenance au problème : intégration au modèle de manière exacte ou approchée, génération de chemins respectant certaines conditions uniquement, ajout d'arcs de maintenance dans le graphe, etc.

Cependant, l'ajout de la maintenance ou de critères de robustesse non pris en compte dans la méthode complexifierait davantage le modèle déjà difficile à résoudre efficacement.

1. Les ajustements de la méthode (précision, troncature, critère d'arrêt lié au nombre d'itérations) ne permettent pas de garantir une solution continue optimale.

Par ailleurs, l'approche pourrait permettre de redistribuer les flots, au risque de compliquer davantage le modèle, mais n'est pas adaptée dans le cas de sillons à vide non fixés.

Chapitre 8

Approche par programmation mathématique et résolution exacte

L'approche (MPT) répond à un problème moins restreint que les approches (HGL) et (CGC). En effet, on récupère les informations issues de la résolution de P_{PRESTO} , mais on se permet une marge de manœuvre par rapport à la solution de flots : plutôt que de se contraindre aux flots calculés par PRESTO, on exploite directement les demandes et les capacités sur chaque tâche. On résout un second PLNE P_{MPT} permettant de construire une solution en contrôlant les coûts de production, en respectant au mieux la maintenance et en intégrant les critères de robustesse. L'articulation de cette méthode avec PRESTO est illustrée en Figure 8.1.

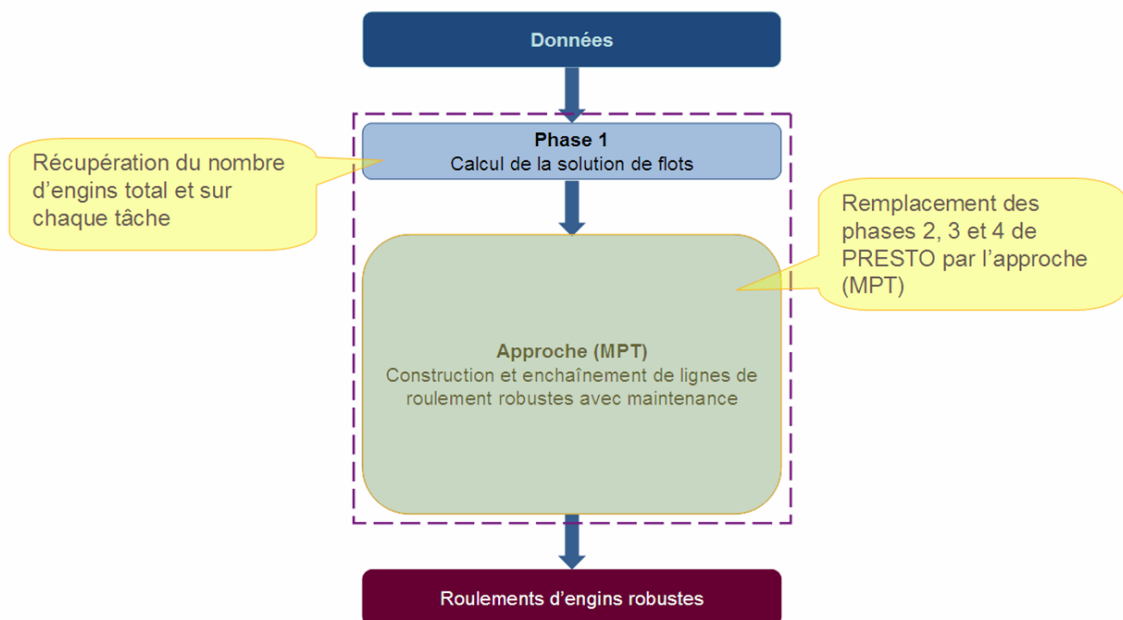


FIGURE 8.1 – Fonctionnement de la méthode par rapport à PRESTO.

8.1 Spécifications du problème considéré

Comme détaillé en section 5.1, l'approche (MPT) intervient après résolution de P_{PRESTO} . On récupère alors le nombre d'engins à utiliser et les tâches à couvrir. Cependant, on n'impose pas le respect des flots calculés et on autorise la création de sillons à vide différents. Le problème n'étant alors pas restreint, on a une plus grande marge de manœuvre pour gagner en robustesse et respecter les contraintes de maintenance.

8.1.1 Données d'entrée

En entrée, on connaît :

- N le nombre d'engins à utiliser ;
- les tâches à couvrir et leurs caractéristiques : gares et dates de départ et d'arrivée, demande et capacité exprimées en nombre d'engins ;
- les couples origine-destination (OD) utilisés par PRESTO pour la création de sillons à vide auxquels sont associés une distance en kilomètres et une durée en minutes ;
- les données nécessaires pour le respect des contraintes de maintenance : pas de maintenance, durée minimale d'immobilisation, technicentre et évolutions vers ce technicentre.

On remarque que la demande n'est plus exprimée en passagers à transporter mais par le nombre d'engins minimum requis, puisqu'à cette étape on sait quel type d'engin couvre chaque tâche.

Pour des raisons de combinatoire élevée, on ne retient que les OD utilisées par PRESTO pour la création de sillons à vide.

8.1.2 Sortie

En sortie, on obtient un roulement de N engins cyclique et robuste avec maintenance.

8.1.3 Problème de base

Pour répondre au problème de planification des roulements d'engins, on cherche à construire un roulement d'engins cyclique minimisant les coûts de production.

Puisque le nombre d'engins et les déplacements en actif sont fixés, les coûts de production à minimiser dans l'objectif sont les coûts liés aux sillons à vide et aux déplacements en passif. Le coût total d'un trajet (sillon à vide ou déplacement en passif) est lié au nombre de kilomètres parcourus pondéré. On connaît donc le coût de chaque trajet.

Les contraintes à respecter sont la couverture des tâches couvertes par PRESTO, le respect des capacités sur ces tâches, la cohérence spatio-temporelle, la cyclicité et le respect impératif du nombre d'engins utilisés par PRESTO.

8.1.4 Problème complet

On cherche maintenant à construire un roulement d'engins robuste avec maintenance.

On intègre alors les critères de robustesse suivants, détaillés à la section 4.2 :

- homogénéisation des temps de retournement (*invret*) ;
- minimisation du nombre de recompositions (*recomp*) ;

- pénalisation des temps de retournement intermédiaires (*tretInt*).

Par ailleurs, des contraintes de maintenance doivent être respectées.

8.2 Modélisation du problème

On modélise le problème spécifié par un programme linéaire en nombres entiers de base (PLNE) présenté à la section 8.2.3) s'appuyant sur un graphe propre à l'approche (MPT) présenté à la section 8.2.1. En effet, si le graphe issu de PRESTO représente les tâches par des arcs, l'approche (MPT) est basée sur une représentation des tâches par des sommets. Le PLNE, décrit ensuite à la section 8.2.7, gère à la fois les coûts de production, la robustesse et les contraintes de maintenance, en tenant compte de priorités.

Cacchiani et al. (2010), en s'intéressant au problème de planification des roulements d'engins, se sont penchés sur une modélisation comparable. Mais dès la spécification de leur problème on trouve des différences :

- plusieurs types d'engins sont considérés simultanément ;
- les coûts de production consistent en une somme des coûts forfaitaires des engins (différents selon le type d'engins) ;
- les sillons à vide également autorisés ne sont pas pénalisés dans l'objectif ;
- l'horizon temporel correspond à une journée (modifiant en particulier la vision de la maintenance) ;
- la robustesse n'est pas considérée.

Ainsi, les problèmes diffèrent. De plus, les méthodes de résolution employées ne sont pas les mêmes puisque Cacchiani et al. (2010) utilisent une heuristique en trois étapes basée sur la technique de génération de colonnes.

8.2.1 Modélisation des données sous forme de graphe

Les données sont d'abord partiellement modélisées par un graphe dont les sommets correspondent aux tâches du problème (\mathcal{T}) et les arcs à des enchaînements de tâches possibles.

Les sommets

Les tâches à couvrir sont numérotées de 1 à $|\mathcal{T}|$.

Deux tâches fictives 0 (tâche-source) et $|\mathcal{T}|+1$ (tâche-puits) sont créées.

On crée également $2 \cdot |\mathcal{G}|$ tâches fictives : pour toute gare G , gd_G correspond au début de l'horizon temporel et gf_G correspondant à la fin de l'horizon temporel en la gare G . On note EGd et EGf les ensembles de ces tâches : $EGd = \bigcup_{G \in \mathcal{G}} gd_G$ et $EGf = \bigcup_{G \in \mathcal{G}} gf_G$.

Les arcs

Un arc est créé entre deux sommets lorsque les tâches correspondantes (réelles ou fictives) peuvent être enchaînées. À chaque arc est associé un poids qui équivaut au coût de production lié à son utilisation. Les arcs sont créés selon les règles suivantes :

- on crée un arc de poids nul de i vers j pour toute paire de tâches réelles $i, j \in \mathcal{T}$ telle que j peut suivre i directement (si $D_j^{deb} \geq D_i^{fin} + tc_{G_i^{arr}}$ et $G_j^{dep} = G_i^{arr}$, cf. définition du temps de crochet à la section 1.1.3);
- on crée un arc de poids nul de la tâche-source 0 vers gd_G pour toute tâche fictive $gd_G \in EGd$ et un arc de gf_G vers la tâche-puits $|\mathcal{T}| + 1$ pour toute tâche fictive $gf_G \in EGd$;
- de toute tâche fictive correspondant à une gare G en début d’horizon temporel, on crée un arc de poids nul vers toute tâche réelle partant de cette gare (de $gd_G \in EGd$ vers $j \in \mathcal{T}$ telle que $G_j^{dep} = gd_G$);
- de toute tâche réelle finissant en gare G , on crée un arc de poids nul vers la tâche fictive correspondant à la gare G en fin d’horizon temporel, (de $j \in \mathcal{T}$ vers $gf_G \in EGf$, avec $G_j^{arr} = gf_G$).

La Figure 8.2 permet de visualiser le graphe correspondant aux données présentées par la grille horaire en Figure 1.1 du Chapitre 1. Les poids sur les arcs, tous nuls, n’apparaissent pas dans cet exemple.

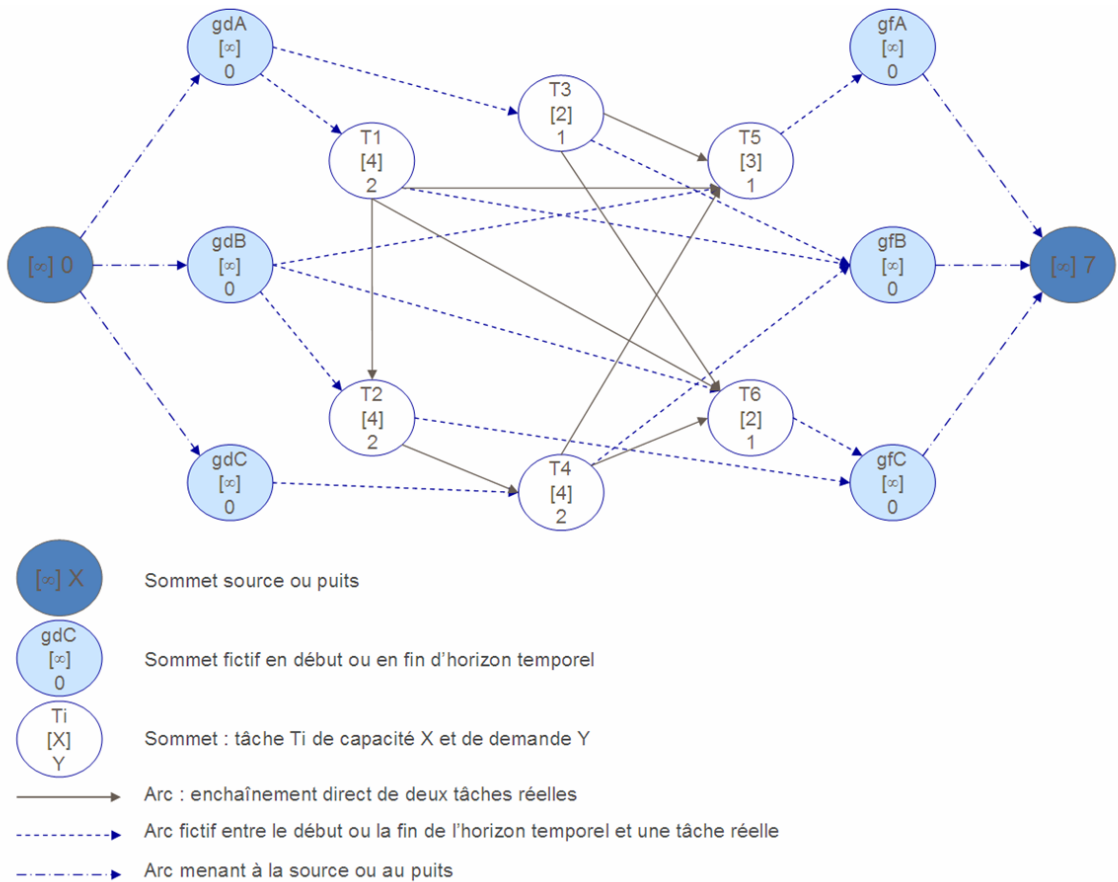


FIGURE 8.2 – Modélisation des données de l’exemple de la Figure 1.1 sous forme d’un graphe propre à l’approche (MPT) pour lequel la demande est exprimée par un nombre d’engins.

Les arcs à vide

De plus, des arcs de poids non nuls sont ajoutés pour prendre en compte la possibilité d’insérer un sillon à vide (W) entre deux tâches. Deux tâches i et j réelles ou fictives peuvent être enchaînées

moyennant l'utilisation d'un sillon à vide si :

- il y a cohérence entre les gares : l'OD (G_i^{arr}, G_j^{dep}) est autorisée dans les données (si elle est utilisée dans la solution de flots) ;
- le temps de retournement entre les deux tâches est suffisant pour insérer un sillon à vide de durée $Dw_{i,j}$ tout en respectant le temps de crochet : $D_j^{deb} \geq D_i^{fin} + tc(G_i^{arr} + Dw_{i,j})$.

Le poids d'un arc correspond au coût du sillon à vide. Plusieurs ensembles d'arcs sont ainsi créés :

- on crée un arc de poids $Cw_{i,j}$ de i vers j pour toute paire de tâches réelles $i, j \in \mathcal{T}$ pouvant être enchaînées moyennant l'utilisation d'un sillon à vide W de coût $Cw_{i,j}$ et de durée $Dw_{i,j}$;
- on crée un arc de poids $Cw_{i,j}$ de $gd_G \in EGd$ vers $j \in \mathcal{T}$ s'il est possible d'emprunter un sillon à vide de poids $Cw_{i,j}$ de G vers G_j^{dep} entre le début de l'horizon temporel et D_j^{deb} ;
- on crée un arc de poids $Cw_{i,j}$ de $j \in \mathcal{T}$ vers $gf_G \in EGf$ s'il est possible d'emprunter un sillon à vide de poids $Cw_{i,j}$ de G_j^{arr} vers G entre D_j^{fin} et la fin de l'horizon temporel.

Dans l'exemple de la Figure 8.2, on considère qu'il est possible de créer des sillons à vide d'une distance de 32 kilomètres et d'une durée de 27 minutes de B vers C . Il faut alors ajouter :

- un arc de poids 32 de $T1$ vers $T4$ correspondant à l'enchaînement possible de $T1$ et $T4$ moyennant l'utilisation d'un sillon à vide ;¹
- un arc de poids 32 de gdB vers $T4$ correspondant à un sillon à vide en début d'horizon temporel avant la tâche $T4$;
- un arc de poids 32 de toute tâche finissant en gare B ($T1, T3$ et $T4$) vers gdC correspondant à un sillon à vide en fin d'horizon temporel.

En revanche, il n'y a pas d'arc supplémentaire entre $T3$ et $T4$ car le temps de retournement de 30 minutes est insuffisant pour ajouter un sillon à vide de 27 minutes et respecter le demi temps de crochet de 4 minutes.

On obtient le graphe exhaustif présenté en Figure 8.3. Pour plus de lisibilité, les arcs déjà représentés sur la Figure 8.2 sont en filigrane, laissant apparaître plus clairement les arcs-à-vide ajoutés.

1. Le coût d'un sillon à vide correspond à sa distance en kilomètres, il est ensuite pénalisé dans l'objectif du PLNE (pénalité Pw).

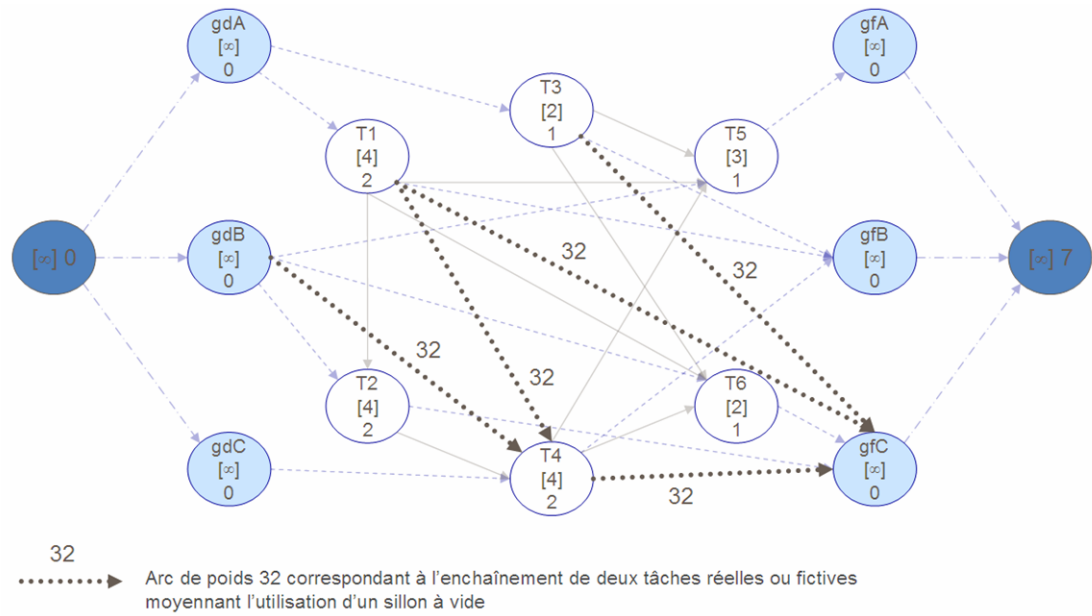


FIGURE 8.3 – Ajout des arcs à vide dans le graphe propre à l'approche (MPT) donné en Figure 8.2.

Chaque chemin de la tâche-source à la tâche-puits dans ce graphe correspond à une ligne de roulement potentielle.

8.2.2 Notations

Le graphe décrit à la section 8.2.1 engendre les notations suivantes :

- 0 Tâche fictive source.
- $|\mathcal{T}| + 1$ Tâche fictive puits.
- gd_G Tâche fictive en début de ligne correspondant à la gare G .
- gf_G Tâche fictive en fin de ligne correspondant à la gare G .
- EGd Ensemble des tâches fictives en début de ligne (une par gare).
- EGf Ensemble des tâches fictives en fin de ligne (une par gare).
- $\mathcal{T}c$ Ensemble des tâches complet (réelles et fictives).
- Ad Ensemble des arcs correspondant à une succession directe de deux tâches réelles.
- Aw Ensemble des arcs correspondant à une succession de deux tâches réelles via un sillon à vide.
- Ar Ensemble des arcs correspondant à une succession de deux tâches réelles : $Ar = Ad \cup Aw$.
- Ac Ensemble complet des arcs : $Ac = Ar \cup Af$.

En complément de ces notations et des notations définies à la section 5.1, on énumère ici

quelques paramètres supplémentaires propres à l'approche (MPT) :

Δ_{cout}	Valeur maximale autorisée des coûts de production.
$Cw_{i,j}$	Coût du sillon à vide pour enchaîner les tâches i et j ou coût d'une évolution de la gare i à la gare j .
$Dw_{i,j}$	Durée du sillon à vide pour enchaîner les tâches i et j .
Pw	Pénalité des sillons à vide dans l'objectif.
$Pevo$	Pénalité des évolutions dans l'objectif.
$Cpassif$	Coût d'un déplacement en passif.
$Pinvret$	Pénalité de la somme des inverses des temps de retournement dans l'objectif.
$Precomp$	Pénalité des recompositions dans l'objectif.
$C_{i,j}$	Nombre d'engins maximum pouvant circuler sur l'arc (i, j) : $C_{i,j} = \min(cap(i), cap(j))$.

Par ailleurs, on définit de nouvelles notations correspondant aux dates de début et de fin des lignes :

$D_{gdG,j}^{deb}$	Date de début d'une ligne de gare de départ G et de première tâche j .
$D_{i,gfG}^{fin}$	Date de fin d'une ligne de dernière tâche i et de gare d'arrivée G .

Ces notations sont utiles car les sillons à vide ne sont pas considérés comme des tâches ici. Or la date de début d'une ligne est la date de début de sa première tâche ou de son premier sillon à vide. En connaissant la première tâche d'une ligne j , on n'a donc pas assez d'informations pour en déduire la date de début de la ligne. En effet, il peut y avoir un sillon à vide avant j .

Dans l'exemple de la Figure 8.3, on considère une ligne ℓ commençant par la tâche fictive gdB , c'est-à-dire de gare de départ B . Si la première tâche de la ligne est $T2$, il n'y a pas de sillon à vide en début d'horizon temporel, la date de début de ℓ est celle de $T2$. En revanche, si $T4$ est la première tâche de ℓ , il y a un sillon à vide en début d'horizon temporel et la date de début de ℓ est celle du sillon à vide.

Par défaut, les dates de début et de fin d'un sillon à vide en début d'horizon temporel sont les dates au plus tard. Elles sont calculées en fonction de la tâche qui suit le sillon à vide. On rappelle que l'intervalle minimum de temps entre un sillon à vide et une tâche est un demi temps de crochet.

Soit une ligne de gare de départ G et de première tâche j . Sa date de début est calculée comme suit :

$$D_{gdG,j}^{deb} \begin{cases} = D_j^{deb} & \text{si } G_j^{dep} = G \\ = D_j^{deb} - Dw_{gdG,j} - \frac{1}{2.tc(G)} & \text{sinon.} \end{cases}$$

De la même façon, il faut pouvoir repérer l'existence d'un sillon à vide en fin d'horizon temporel pour calculer la date de fin d'une ligne. Les dates d'un tel sillon à vide sont les dates au plus tôt, calculées d'après la tâche qui le précède.

Soit une ligne de dernière tâche i et de gare d'arrivée G . Sa date de fin est calculée comme suit :

$$D_{i,gfG}^{fin} \begin{cases} = D_i^{fin} & \text{si } G_i^{arr} = G \\ = D_i^{fin} + Dw_{i,gfG} + \frac{1}{2.tc(G)} & \text{sinon.} \end{cases}$$

8.2.3 Modélisation du problème de base par un PLNE

On modélise ensuite le problème sous forme d'un programme linéaire en nombres entiers correspondant à la recherche d'un roulement d'engins cyclique optimisé sans maintenance et sans robustesse. Deux types de variables de décision sont utilisés : pour tout engin k et pour tout couple de tâches (i, j) , $Y_{k,i,j}$ indique l'enchaînement de i et j par k .

$$\forall k = 1..N, (i, j) \in Ac,$$

$$Y_{k,i,j} \begin{cases} = 1 & \text{si l'engin } k \text{ enchaîne les tâches } i \text{ et } j \\ = 0 & \text{sinon.} \end{cases}$$

Ces variables sont suffisantes pour exprimer tous les critères et les contraintes du modèle, mais on définit également des variables de travail qui facilitent le traitement des résultats sans influencer sur la résolution (temps de calcul, solution) : pour tout engin k et pour toute tâche i , $X_{k,i}$ renseigne la couverture de i par k .

$$\forall k = 1..N, i = 1..|\mathcal{T}|,$$

$$X_{k,i} \begin{cases} = 1 & \text{si l'engin } k \text{ passe par la tâche } i. \\ = 0 & \text{sinon.} \end{cases}$$

La variable $X_{k,i}$ sera non nulle s'il existe une tâche j telle que l'engin k enchaîne j et i . On peut donc définir cette variable par les variables $Y_{k,j,i}$ (prédécesseurs de i possibles) ou par les variables $Y_{k,i,j}$ (successeurs de i possibles) de la façon suivante :

$$X_{k,i} = \sum_{j \in \mathcal{T}_c} Y_{k,j,i} = \sum_{j \in \mathcal{T}_c} Y_{k,i,j}$$

La tâche 0 n'ayant pas de prédécesseur, on la définit uniquement par ses successeurs possibles :

$$X_{k,0} = \sum_{j \in EGd} Y_{k,0,j}$$

On peut alors écrire un PLNE correspondant au problème de planification des roulements d'engins :

$$P_{MPT_base} : \min C_{passif} \cdot \left(\sum_{i \in \mathcal{T}} \left(\sum_{k=1}^N X_{k,i} \right) - dem(i) \right) + Pw \cdot \left(\sum_{k=1}^N \sum_{(i,j) \in Aw} Cw_{i,j} \cdot Y_{k,i,j} \right)$$

$$X_{k,0} = 1 \quad k = 1, \dots, N \quad (8.1)$$

$$\sum_{\substack{j \in \mathcal{T}c \\ (j,i) \in Ac}} Y_{k,j,i} = \sum_{\substack{j \in \mathcal{T}c \\ (i,j) \in Ac}} Y_{k,i,j} \quad k = 1, \dots, N, i \in \mathcal{T} \cup EG \quad (8.2)$$

$$\sum_{k=1}^N X_{k,i} \geq dem(i) \quad i \in \mathcal{T} \quad (8.3)$$

$$\sum_{k=1}^N X_{k,i} \leq cap(i) \quad i \in \mathcal{T} \quad (8.4)$$

$$X_{k,gf_G} = X_{(k \bmod N)+1,gd_G} \quad k = 1, \dots, N, G \in \mathcal{G} \quad (8.5)$$

$$\sum_{\substack{i \in EGd \\ (i,j) \in Ac}} D_{j,i}^{deb} \cdot Y_{(k \bmod N)+1,i,j} + L - \sum_{\substack{j \in EGf \\ (i,j) \in Ac}} D_{i,j}^{fin} \cdot Y_{k,i,j} \geq M, \quad k = 1, \dots, N \quad (8.6)$$

$$X_{k,i} = \sum_{\substack{(j,i) \in Ac \\ j \in \mathcal{T}c}} Y_{k,j,i} \quad k = 1, \dots, N, i \in \mathcal{T}c - \{0\} \quad (8.7)$$

$$X_{k,0} = \sum_{\substack{j \in EGd \\ (0,j) \in Ac}} Y_{k,0,j} \quad k = 1, \dots, N \quad (8.8)$$

$$Y_{k,j,i} \in \{0, 1\} \quad k = 1, \dots, N, i \in \mathcal{T}c, j \in \mathcal{T}c \quad (8.9)$$

$$X_{k,i} \in \{0, 1\} \quad k = 1, \dots, N, i \in \mathcal{T}c \quad (8.10)$$

Dans l'objectif, on cherche à minimiser les coûts de production :

- les coûts en passif (différence entre le nombre d'engins total sur une tâche et sa demande) sont remis en cause, on les minimise avec une pénalité C_{passif} ;
- les coûts liés aux sillons à vide sont remis en cause, chaque sillon à vide entre deux tâches i et j a une pénalité propre $Cw_{i,j}$ (qui peut dépendre de sa distance par exemple), multipliée par un coefficient commun Pw .

Les contraintes (8.1) garantissent l'utilisation de tous les engins : chaque engin k enchaîne la tâche fictive 0 et une tâche réelle (éventuellement en utilisant un sillon à vide). Les contraintes (8.2) assurent le principe de conservation des flots : pour tout sommet i , tout engin k circulant sur un arc entrant doit aussi emprunter un arc sortant.

Une tâche i est couverte si au moins $dem(i)$ engins y sont affectés, mais seuls $cap(i)$ engins au maximum peuvent être affectés à i . Pour chaque sommet i , les contraintes (8.3) assurent qu'on a plus de $dem(i)$ engins incidents intérieurement et les contraintes (8.4) assurent qu'on en a moins de $cap(i)$.

Les contraintes (8.5) et (8.6) sont des contraintes de cyclicité. Les contraintes (8.5) permettent de s'assurer que pour tout k allant de 1 à N , la gare d'arrivée de l'engin k est la même que la gare de départ de l'engin $k+1$ (de l'engin 1 si $k=N$). De la même façon, les contraintes (8.6) garantissent que pour tout engin ℓ allant de 1 à $N-1$, l'intervalle de temps entre la date d'arrivée de l'engin k et la date de départ de l'engin $k+1$ (de l'engin 1 si $k=N$) est supérieur à la marge minimum obligatoire M (par exemple le temps de crochet).

Enfin, les contraintes (8.7) et (8.8) permettent de définir les variables $X_{k,i}$ en fonction des

variables $Y_{k,i,j}$.

8.2.4 Contrôle des coûts de production

Les coûts de production critiques sont les coûts d'utilisation des engins, très élevés par rapport aux autres coûts, et les coûts en actif, liés aux kilomètres parcourus par les engins en actif. Dans l'approche (MPT), le nombre d'engins calculé par PRESTO et les tâches couvertes sont conservés. Les coûts d'utilisation des engins et des coûts en actif des solutions obtenues par (MPT) sont donc identiques à ceux des solutions de PRESTO.

La redistribution des sillons à vide et des flots peut modifier les coûts en passif et à vide liés aux kilomètres parcourus en passif et à vide. Ces coûts étant moins critiques que les coûts d'utilisation et en actif, on ne cherche pas à respecter l'optimum calculé par PRESTO. Cependant, afin de limiter leur dégradation, on les pénalise dans l'objectif.

8.2.5 Modélisation des critères de robustesse

Temps de retournement

On rappelle que le temps de retournement entre deux tâches i et j qui s'enchaînent est défini par $t_{i,j} = D_j^{deb} - D_i^{arr}$.

Dans un premier temps, on cherche à répartir de façon équilibrée les temps de retournement dans le roulement. Dans ce but, on optimise le critère *invret* (détaillé à la section 4.2). Par défaut, tout temps de retournement inférieur à une minute sera pris en compte comme un temps de retournement d'une minute.

Le temps de retournement entre deux tâches i et j réelles qui s'enchaînent directement ($(i, j) \in Ad$) est l'intervalle entre la date de fin de i et la date de début de j . Si cet intervalle est inférieur à 1, on prend la valeur 1 :

$$\frac{1}{\max(1, D_j^{deb} - D_i^{arr})} \cdot Y_{k,i,j}$$

Entre deux tâches i et j enchaînées par un sillon à vide ($(i, j) \in Aw$), il y a en fait deux temps de retournement : le temps de retournement entre la fin de i et le début du sillon à vide et le temps de retournement entre la fin du sillon à vide et le début de j . Par défaut, le sillon à vide est au milieu des deux tâches, donc les deux temps de retournement ont la même durée :

$$\max(1, \frac{D_j^{deb} - D_i^{arr} - Dw_{i,j}}{2})$$

Dans un second temps, on cherche à éviter les temps de retournement intermédiaires : à la fois trop longs pour qu'un engin reste en gare et trop courts pour qu'un engin soit remisé. Ce critère (*tretInt*) est détaillé à la section 4.2. Dans ce but, on définit un intervalle d'exclusion $[tinf, tsup]$ et on pénalise dans l'objectif tout temps de retournement appartenant à cet intervalle.

Nombre de recompositions

On considère aussi le critère *recomp* (cf. section 4.2) : on cherche à minimiser le nombre de recompositions en enchaînant plusieurs tâches avec une même UM quand c'est possible.

Soit T_i une tâche couverte par n engins ($n > 1$). Si tous les engins viennent d'effectuer des tâches différentes en US, il y aura $n - 1$ recompositions pour coupler les engins avant T_i . Si certains engins sont déjà couplés de façon à former p ensembles d'engins, il faudra coupler les p ensembles, donc il y aura $p - 1$ recompositions (cf. section 4.2.1). Le nombre d'opérations de couplage dans

le roulement est donc la somme des recompositions ainsi calculées pour toute tâche en UM. Pour minimiser le nombre de couplages, on peut chercher à minimiser le nombre d'ensembles d'engins à coupler avant chaque tâche.

Dans le graphe présenté à la section 8.2.1, on considère un sommet correspondant à la tâche T_i . Un arc entrant de flot non nul correspond à un ensemble d'engins (US ou UM déjà couplée). Le nombre d'arcs entrants de flot non nul correspond donc au nombre d'ensembles d'engins à coupler avant T_i . En minimisant le nombre d'arcs entrants en chaque sommet, on minimise donc le nombre de couplages.

D'après le même raisonnement concernant les opérations de découplage en fin de tâche et les arcs sortants, minimiser le nombre d'arcs de flot non nul dans le graphe revient à minimiser le nombre de recompositions dans le roulement (couplages et découplages).

Pour intégrer le critère *recomp* dans le modèle, on ajoute alors une nouvelle variable booléenne :

$$e_{i,j} \begin{cases} = 1 & \text{si l'arc } (i,j) \text{ est utilisé, c'est-à-dire s'il existe au moins un engin circulant sur cet arc.} \\ = 0 & \text{sinon.} \end{cases}$$

On peut définir ces variables grâce aux variables de décision $Y_{k,i,j}$. En effet, un arc (i,j) est utilisé dès qu'un engin circule dessus, ce qui se traduit par : $e_{i,j} = 1$ s'il existe au moins un engin k pour lequel $Y_{k,i,j} = 1$, soit :

$$C_{i,j} \cdot e_{i,j} \geq \sum_{k=1}^N Y_{k,i,j}$$

$C_{i,j}$ est une constante supérieure ou égale à la somme des $Y_{k,i,j}$. $\sum_{k=1}^N Y_{k,i,j}$ représente le nombre d'engins enchaînant les tâches i et j . Cette somme sera donc inférieure à la capacité de chaque tâche. On peut donc choisir la constante telle que :

$$C_{i,j} = \min(\text{cap}(i), \text{cap}(j)), (i,j) \in Ar$$

Une autre formulation avait été proposée pour définir les variables $e_{i,j}$:

$$e_{i,j} \geq Y_{k,i,j}, k = 1..N, (i,j) \in Ar$$

Lors des tests de résolution de P_{MPT} , nous avons pu constater que cette formulation était moins efficace.

Pour minimiser le nombre de recompositions, il suffit alors de minimiser la somme pénalisée des variables $e_{i,j}$, soit :

$$\min \text{Precomp} \cdot \left(\sum_{(i,j) \in \mathcal{T}} e_{i,j} \right)$$

avec *Precomp* une pénalité à définir.

8.2.6 Modélisation de la maintenance

Pour chaque série de matériel p , on connaît le pas d'entrée en maintenance pm_p en jours, la durée minimale d'immobilisation dm_p en minutes et le technicentre (ou dépôt, considéré comme unique pour un type d'engin dans le problème) correspondant $depot_p$ (cf. section 1.1.3). Puisque nous traitons chaque type d'engin indépendamment, nous pouvons nous affranchir de l'indice p dans les notations.

Connaître le site de maintenance revient à connaître ses horaires d'ouverture et les évolutions possibles vers ce site de n'importe quelle gare. Une *évolution* est un trajet à vide similaire aux sillons à vide, mais qui a pour origine ou pour destination un technicentre. Lorsqu'un engin stationne dans une gare G , on sait donc s'il peut aller au technicentre $depot$ et on connaît la durée $d_{G,depot}$ et le coût $Cw_{G,depot}$ du trajet entre la gare et le technicentre. Ici, la durée $d_{G,depot}$ inclut le demi temps de crochet obligatoire en gare G .

Pour permettre aux engins d'effectuer des créneaux de maintenance, on définit de nouvelles variables de décision. Pour toute tâche i réelle ou fictive en début de ligne ($\forall i \in \mathcal{T} \cup EGD$), on envisage un créneau de maintenance m_i commençant au plus tôt après la tâche i . Pour chaque engin k , on crée alors une variable $Z_{k,i}$ correspondant au créneau m_i telle que :

$$Z_{k,i} \begin{cases} = 1 & \text{si l'engin } k \text{ entre en maintenance après la tâche } i. \\ = 0 & \text{sinon.} \end{cases}$$

Si un engin entre en maintenance après une tâche i , on calcule les dates au plus tôt de début Dm_i^{deb} et de fin Dm_i^{fin} du créneau de maintenance m_i induit. L'intervalle de temps entre la date de fin de i et la date de début de m_i doit être au moins égal à la durée d'une évolution de la gare d'arrivée de i au dépôt, augmenté d'un demi temps de crochet, soit $d_{G_i^{arr},depot}$. Lorsque la date de début du créneau m_i a été définie, on peut en déduire sa date de fin au plus tôt, qui respecte la durée minimale d'immobilisation en gare (dm). On définit alors :

$$\begin{cases} Dm_i^{deb} = D_i^{fin} + d_{G_i^{arr},depot} \\ Dm_i^{fin} = Dm_i^{deb} + dm \end{cases}$$

La Figure 8.4 permet de visualiser le lien entre les différentes dates et durées évoquées sur l'exemple d'un créneau suivant une tâche T_i au plus tôt. On note que la définition des dates d'un créneau de maintenance m_i n'est pas liée à la tâche suivante (T_j sur le schéma).

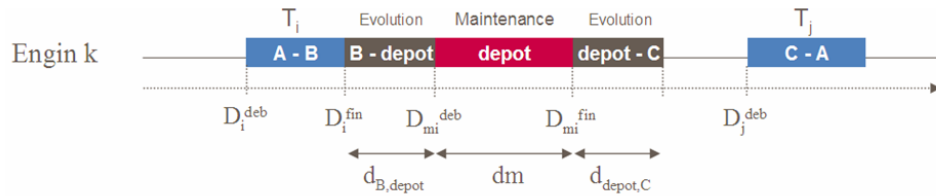


FIGURE 8.4 – Dates de début et de fin au plus tôt d'un créneau de maintenance suivant une tâche T_i .

La contrainte de maintenance peut alors s'exprimer de la façon suivante : chaque engin de la série doit passer au site de maintenance $depot$ tous les pm jours et y rester pendant au moins dm minutes. Pour ce faire, certaines contraintes doivent être vérifiées. D'une part, un engin peut aller en maintenance entre deux tâches i et j si l'engin a le temps d'aller au technicentre, d'effectuer les opérations de maintenance et de revenir en gare G_j^{dep} entre l'arrivée de i et le départ de j : $d_{G_i^{arr},depot} + dm + d_{depot,G_j^{dep}} \leq D_j^{deb} - D_i^{fin}$. Le site de maintenance doit être ouvert entre les horaires de début et de fin du créneau de maintenance, sinon le créneau ne pourra pas être utilisé.

Dans notre modèle, la contrainte de maintenance n'impose pas pour chaque engin d'aller en maintenance tous les pm jours exactement, c'est plutôt une indication (date optimale) : une telle

contrainte serait trop restrictive et ne correspondrait pas à la réalité du métier. Nous imposons en fait un intervalle minimum \overline{pm} inférieur à pm entre deux créneaux de maintenance et un nombre de créneaux minimum par ligne noté $ncmin$. Il faut éviter d'ajouter trop de créneaux, ce qui serait trop cher et inutile. Nous bornons alors le nombre de créneaux de maintenance par engin par un nombre maximal $ncmax$. Par ailleurs, les coûts liés aux évolutions, comparables aux coûts des sillons à vide classiques, sont pénalisés dans l'objectif pour favoriser les départs en maintenance dans des gares proches du technicentre : on cherche à minimiser le nombre de kilomètres parcourus.

On définit les ensembles suivants pour chaque tâche i :

- \overline{E}_i l'ensemble des créneaux trop proches du créneau m_i , c'est-à-dire dans un intervalle inférieur à \overline{pm} suivant m_i , soit $\overline{E}_i = \{j \in \mathcal{T}c : (i, j) \in Ac, Dm_i^{fin} \leq Dm_j^{deb} \leq Dm_i^{fin} + \overline{pm}\}$;
- F_i l'ensemble des tâches j telles qu'un engin k peut enchaîner i et j sauf s'il emprunte le créneau m_i , soit $F_i = \{j \in \mathcal{T}c : (i, j) \in Ac, D_j^{deb} < Dm_i^{fin} + d_{G_j^{dep}, depot}\}$.

Sur l'exemple de la Figure 8.5, on a défini cinq créneaux de maintenance $M1$ à $M5$ pouvant être empruntés. Soit un engin k passant par $M1$. Alors k ne peut pas passer en maintenance pendant une durée minimum \overline{pm} . Or $M2$, $M3$ et $M4$ commencent avant la fin de la durée minimale. k ne peut donc pas emprunter un de ces créneaux, soit $M2, M3, M4 \in \overline{E}_1$. En revanche, il peut passer par $M5$ ($M5 \notin \overline{E}_1$).

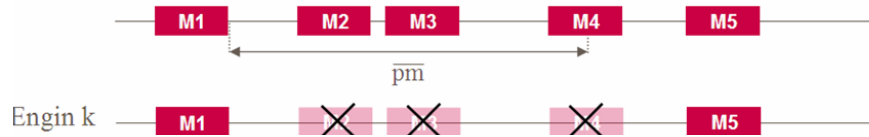


FIGURE 8.5 – Plusieurs créneaux de maintenance appartenant à \overline{E}_1 .

Sur l'exemple de la Figure 8.6, l'engin k peut enchaîner les tâches T_i et T_j (ligne de roulement du haut). En revanche, s'il passe en maintenance après T_i (ligne du bas), il ne peut plus couvrir T_j . Alors $T_j \in F_i$.

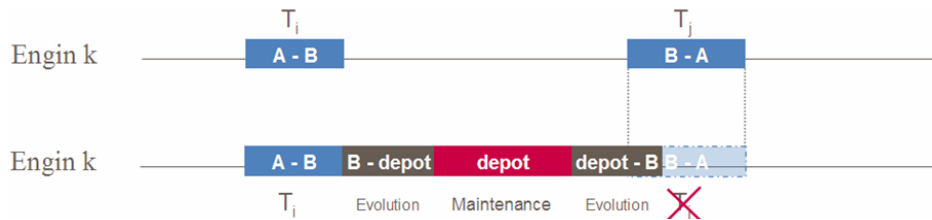


FIGURE 8.6 – Exemple de tâche ne pouvant plus être enchaînée après un créneau de maintenance ($T_j \in F_i$).

Il est à noter qu'un créneau qui n'est pas complètement inclus dans une plage d'ouverture de maintenance du technicentre ne pourra pas être utilisé. On peut garantir cette contrainte en posant simplement $Z_{k,i} = 0$ pour tout créneau i non utilisable. Il s'agit d'une approximation : plutôt que d'interdire un créneau dont les dates ne correspondent pas à une plage d'ouverture du technicentre, on pourrait chercher à redéfinir la date de début du créneau comme le maximum entre la date au plus tôt après une tâche i et la date d'ouverture du technicentre.

Enfin, on exprime les contraintes de maintenance de la façon suivante :

$$Z_{k,i} \leq X_{k,i} \quad k = 1, \dots, N, i \in \mathcal{T} \cup EGd \quad (8.11)$$

$$\sum_{i \in \mathcal{T} \cup EGd} Z_{k,i} \geq n_{cmin} \quad k = 1, \dots, N \quad (8.12)$$

$$\sum_{i \in \mathcal{T} \cup EGd} Z_{k,i} \leq n_{cmax} \quad k = 1, \dots, N \quad (8.13)$$

$$X_{k,j} \leq 1 - Z_{k,i} \quad k = 1, \dots, N, i \in \mathcal{T}, j \in F_i \cup EGd \quad (8.14)$$

$$\sum_{j \in \overline{E_i}} Z_{k,j} \leq 1 - Z_{k,i} \quad k = 1, \dots, N, i \in \mathcal{T} \cup EGd \quad (8.15)$$

Les contraintes (8.11) garantissent qu'un engin effectuant le créneau m_i passe par la tâche i . Les contraintes (8.12) et (8.13) imposent des bornes inférieure et supérieure au nombre de créneaux par ligne. Lorsqu'un engin emprunte un créneau m_i , les contraintes (8.14) assurent qu'il ne pourra pas emprunter des tâches qui commencent plus tôt que le retour de maintenance, et les contraintes (8.15) interdisent l'emprunt d'un créneau dans l'intervalle minimum suivant m_i .

8.2.7 Modélisation du problème complet par un PLNE

On modélise le problème de planification robuste des roulements d'engins (avec maintenance) avec contrôle des coûts par un PLNE plus complet.

On rappelle les différents types de variables de décision utilisés :

$$Y_{k,i,j} \begin{cases} = 1 & \text{si l'engin } k \text{ circule sur l'arc } (i, j). \\ = 0 & \text{sinon.} \end{cases}$$

$$X_{k,i} \begin{cases} = 1 & \text{si l'engin } k \text{ effectue la tâche } i. \\ = 0 & \text{sinon.} \end{cases}$$

$$e_{i,j} \begin{cases} = 1 & \text{si l'arc } (i, j) \text{ est utilisé} \\ = 0 & \text{sinon.} \end{cases}$$

$$Z_{k,i} \begin{cases} = 1 & \text{si l'engin } k \text{ effectue le créneau } i. \\ = 0 & \text{sinon.} \end{cases}$$

Le PLNE complet est alors le suivant :

$$\begin{aligned}
P_{MPT} : \min & Cpassif. \left(\sum_{i \in \mathcal{T}} \left(\sum_{k=1}^N X_{k,i} \right) - dem(i) \right) \\
& + Pw. \left(\sum_{k=1}^N \sum_{(i,j) \in Aw} Cw_{i,j} \cdot Y_{k,i,j} \right) + Pevo. \left(\sum_{k=1}^N \sum_{i \in \mathcal{T} \cup EGd} 2 \cdot Cw_{G_i^{arr}, depot} \cdot Z_{k,i} \right) \\
& + Pinvret. \left(\sum_{k=1}^N \sum_{(i,j) \in Ad} \frac{1}{\max(1, D_j^{deb} - D_i^{arr})} \cdot Y_{k,i,j} \right) + Pinvret. \left(\sum_{k=1}^N \sum_{(i,j) \in Aw} \frac{2}{\max(1, \frac{D_j^{deb} - D_i^{arr} - Dw_{i,j}}{2})} \cdot Y_{k,i,j} \right) \\
& + \sum_{t_{i,j} \in [tinf, tsup]} Ptret + Precomp. \left(\sum_{(i,j) \in \mathcal{T}} e_{i,j} \right)
\end{aligned}$$

$$X_{k,0} = 1 \quad k = 1, \dots, N \quad (8.1)$$

$$\sum_{\substack{j \in \mathcal{T}^c \\ (j,i) \in Ac}} Y_{k,j,i} = \sum_{\substack{j \in \mathcal{T}^c \\ (i,j) \in Ac}} Y_{k,i,j} \quad k = 1, \dots, N, i \in \mathcal{T} \cup EG \quad (8.2)$$

$$\sum_{k=1}^N X_{k,i} \geq dem(i) \quad i \in \mathcal{T} \quad (8.3)$$

$$\sum_{k=1}^N X_{k,i} \leq cap(i) \quad i \in \mathcal{T} \quad (8.4)$$

$$X_{k,gf_G} = X_{(k \bmod N)+1, gd_G} \quad k = 1, \dots, N, G \in \mathcal{G} \quad (8.5)$$

$$\begin{aligned}
& \sum_{\substack{i \in EGd \\ (i,j) \in Ac}} D_{j,i}^{deb} \cdot Y_{(k \bmod N)+1, i, j} + L \\
& - \sum_{\substack{j \in EGf \\ (i,j) \in Ac}} D_{i,j}^{fin} \cdot Y_{k, i, j} \geq M, \quad k = 1, \dots, N \quad (8.6)
\end{aligned}$$

$$X_{k,i} = \sum_{\substack{(j,i) \in Ac \\ j \in \mathcal{T}^c}} Y_{k,j,i} \quad k = 1, \dots, N, i \in \mathcal{T}^c - \{0\} \quad (8.7)$$

$$X_{k,0} = \sum_{\substack{j \in EGd \\ (0,j) \in Ac}} Y_{k,0,j} \quad k = 1, \dots, N \quad (8.8)$$

$$C_{i,j} \cdot e_{i,j} \geq \sum_{k=1}^N Y_{k,i,j} \quad (i,j) \in Ar \quad (8.9)$$

$$Z_{k,i} \leq X_{k,i} \quad k = 1, \dots, N, i \in \mathcal{T} \cup EGd \quad (8.10)$$

$$\sum_{i \in \mathcal{T} \cap EGd} Z_{k,i} \geq nmin \quad k = 1, \dots, N \quad (8.11)$$

$$\sum_{i \in \mathcal{T} \cap EGd} Z_{k,i} \leq nmax \quad k = 1, \dots, N \quad (8.12)$$

$$X_{k,j} \leq 1 - Z_{k,i} \quad k = 1, \dots, N, i \in \mathcal{T} \cup EGd, j \in F_i \quad (8.13)$$

$$\sum_{j \in \overline{E_i}} Z_{k,j} \leq 1 - Z_{k,i} \quad k = 1, \dots, N, i \in \mathcal{T} \cup EGd \quad (8.14)$$

$$Y_{k,i,j} \in \{0, 1\}$$

$$X_{k,i} \in \{0, 1\}$$

$$e_{i,j} \in \{0, 1\}$$

$$Z_{k,i} \in \{0, 1\}$$

L'objectif du PLNE est multicritère. En effet, on optimise :

- les coûts de production : les déplacements en passif et les sillons à vide sont pénalisés ;
- les coûts des évolutions : le nombre de kilomètres liés aux déplacements à vide vers le technicentre sont minimisés ;
- les temps de retournement : d'une part, tous les temps de retournement sont uniformisés par minimisation de la somme des inverses, d'autre part, les temps de retournement déconseillés sont fortement pénalisés ;
- les recompositions : le nombre d'opérations de recompositions est limité par la minimisation du nombre d'arcs utilisés.

Les contraintes (8.1) à (8.8) correspondent aux contraintes de base décrites à la section 8.2.3. Les contraintes (8.9) permettent de définir les variables $e_{i,j}$ correspondant à l'utilisation de l'arc (i, j) . Les contraintes (8.11) à (8.15) correspondent aux contraintes de maintenance détaillées à la section 8.2.6.

8.3 Méthode de résolution

Le modèle complet P_{MPT} présenté à la section 8.2.7 peut être résolu directement en entiers par un solveur (IBM ILOG CPLEX pour les tests présentés par la suite) : quelques premiers résultats sont présentés à la section 8.4.

8.4 Premiers résultats et conclusions

Dans cette section, nous présentons quelques résultats concernant les tests de l'approche (MPT) sur les instances cibles (cf. section 5.6). Ces premiers résultats, en plus de valider le bon fonctionnement de la méthode proposée, seront comparés à ceux de PRESTO afin d'évaluer l'intérêt de l'approche en termes de robustesse.

8.4.1 Choix des paramètres

Un plan d'expérimentation préliminaire de la méthode sur les instances cibles en faisant varier les valeurs des paramètres nous a permis de fixer le modèle définitif. Les valeurs des paramètres ont été choisies selon plusieurs critères d'efficacité : temps de résolution, robustesse, dégradation des coûts.

On définit alors les valeurs par défaut suivantes :

- la pénalité des kilomètres à vide $Pw = 1500$;
- la pénalité des inverses des temps de retournement $P_{invret} = 300$;
- la pénalité des temps de retournement intermédiaires $P_{tret} = 100$;
- l'intervalle des temps de retournement intermédiaires $[20, 40]$;
- la valeur de l'écart minimum entre deux créneaux de maintenance $\overline{pm} = 0,5 \cdot pm$;²
- le nombre minimum de créneaux de maintenance par ligne $ncmin = \lfloor \frac{L}{pm} \rfloor$;
- le nombre maximum de créneaux de maintenance par ligne $ncmax = ncmin + 1$.³

2. Sur un exemple avec un horizon temporel d'une semaine, pour $pm = 1$ jour, il y aura au minimum 12 heures entre deux créneaux de maintenance, tandis que pour $pm = 3$ jours, il y aura au minimum 36 heures (1,5 jour) entre deux créneaux.

3. Par exemple, pour $pm = 1$ jour, il y aura au minimum 7 créneaux par ligne et au maximum 8, tandis que pour $pm = 3$ jours, il y aura au minimum 2 créneaux par ligne et au maximum 3.

Les paramètres non listés ci-dessus sont alors nuls par défaut. En effet, d’après les tests réalisés, leur ajout rend la résolution du modèle beaucoup plus complexe : le temps de calcul peut atteindre plusieurs jours.

- la pénalité des déplacements en passif $C_{passif} = 0$;
- la pénalité des kilomètres en évolutions $P_{evo} = 0$;
- la pénalité des recompositions $P_{recomp} = 0$.

Par ailleurs, d’autres ajustements du modèle dont on a pu constater l’efficacité en termes de temps de résolution sont utilisés :

- les coefficients dans l’objectif sont tronqués de façon à ne garder que leur partie entière ;
- les temps de retournement au-delà d’une valeur $trretMax$ (par défaut 60 minutes) ne sont pas pénalisés dans l’objectif ;
- la contrainte limitant la dégradation des coûts par rapport à PRESTO est désactivée.

8.4.2 Tests sur les instances cibles

Nous présentons ici les premiers résultats de la comparaison des méthodes PRESTO et (MPT) par défaut sur les instances cibles (cf. Table 5.1).

La Table 8.1 donne les valeurs de plusieurs critères permettant de comparer les solutions de PRESTO et de (MPT) pour les instances cibles 1 et 2 :

- le temps d’exécution total : pour l’approche (MPT), il correspond au temps d’exécution de PRESTO auquel on ajoute le temps de résolution de P_{MPT} ;
- le temps de résolution continue : pour l’approche (MPT) uniquement, il s’agit du temps de calcul de la relaxation continue à la racine ;
- le temps de résolution entière : pour l’approche (MPT) uniquement, il s’agit du temps de calcul pour trouver la solution entière optimale ;
- le gap entre l’objectif continu à la racine et l’objectif entier : uniquement pour l’approche (MPT) ;
- la robustesse : mesurée ici par *invret*, on présente aussi le gain en robustesse de l’approche (MPT) par rapport à PRESTO ;
- les coûts de production⁴ : les coûts à vide, qui regroupent les sillons à vide et les évolutions (déplacements vers le dépôt), et les coûts en passif, en kilomètres parcourus.

4. les coûts d’utilisation (coût forfaitaire par engin) et les coûts des déplacements en actif ne sont pas mentionnés puisqu’ils sont identiques dans les deux méthodes

Instance	Instance 1		Instance 2	
	PRESTO	MPT	PRESTO	MPT
Temps d'exécution total approximatif	300" (5')	420" (7')	60" (1')	1860" (31')
Temps de résolution continue	-	5"	-	39"
Temps de résolution entière	-	34"	-	1863"
Gap objectif	-	0%	-	0%
Valeur de <i>invret</i>	9,18	3,96	10,52	5,80
Gain sur <i>invret</i> par rapport à PRESTO	-	57%	-	45%
Coûts actifs (en km)	17542	17542	24870	24870
Coûts passifs (en km)	164	368	0	0
Coûts à vide (en km)	10642	10269	11135	13373

TABLE 8.1 – Premiers résultats de PRESTO et de (MPT) sur les instances cibles 1 et 2

Pour les deux instances présentées dans la Table 8.1, les solutions obtenues par PRESTO et par (MPT) respectent les contraintes de maintenance.

Pour les deux solutions obtenues par l'approche (MPT), on observe un gain important sur le critère de robustesse mesuré (*invret*) par rapport à la solution de PRESTO.

Ce résultat n'est pas à coûts de production constants (par rapport à la solution de PRESTO). Les coûts en actif et les coûts d'utilisation des engins sont identiques pour une même instance, mais les coûts en passif et à vide sont volontairement relâchés dans l'approche (MPT).

Pour l'instance 1 à 208 tâches, les coûts en passif ont été plus que doublés, ce qui s'explique par une pénalité nulle dans l'objectif ($C_{passif} = 0$) : on préfère parcourir plus de kilomètres en passif afin de minimiser les déplacements à vide. En effet, les coûts à vide sont plus faibles dans la solution par (MPT). Les déplacements à vide étant généralement plus coûteux que les déplacements en passif, on a donc une solution intéressante en termes de coûts et de robustesse.

En revanche, pour la seconde instance à 314 tâches, l'approche (MPT) dégrade les coûts en passif et à vide⁵ de 20%. Ce résultat trouve une justification dans le fait que pour les tests présentés, les déplacements à vide pour aller au dépôt (évolutions) ne sont pas pénalisés ($P_{evo} = 0$).

La solution reste intéressante puisque le gain en robustesse est important et les coûts de production critiques (les coûts d'utilisation non compensatoires) sont inchangés par rapport à l'optimum calculé par PRESTO. On peut cependant envisager de pénaliser ou d'interdire des évolutions trop coûteuses.

Le temps de calcul des solutions obtenues par l'approche (MPT) est très différent pour les deux tests présentés. Il dépend de l'instance et des paramètres choisis. S'il est difficile d'en tirer des conclusions sur ces deux tests, il paraît raisonnable, mais plus élevé que PRESTO.

Cependant, en décomposant le temps de calcul, on s'aperçoit que le temps de résolution continue est assez faible (moins d'une minute dans les deux cas). Or le gap entre l'objectif de la relaxation continue à la racine et l'objectif entier est nul, mais la solution continue optimale est très fractionnaire. On peut envisager un gain de temps en tentant de trouver une solution entière par une autre

5. Les coûts en passif et à vide ne sont qu'une partie des coûts de production. Il faut y ajouter les coûts en actif (24870 kilomètres) et les coûts d'utilisation d'un engin. En prenant en compte les coûts en actif pour cette instance, les coûts sont dégradés d'à peine 8%.

méthode.

Les deux instances précédentes permettent le respect des contraintes de maintenance avec le nombre minimum d'engins calculé par PRESTO. En revanche, pour l'instance 3, elle n'est pas respectée car non réalisable en l'état (à cause des données ou de la solution de flots).

Par l'approche (MPT), on n'obtient pas de solution admissible, nous avons donc reconfiguré les paramètres pour relâcher davantage la maintenance, sans succès. Nous l'avons donc désactivée dans les deux méthodes PRESTO et (MPT). Les résultats obtenus sont présentés dans la Table 8.2.

Instance	Instance 3	
	PRESTO (sans main- tenance)	MPT (sans mainte- nance)
Temps d'exécution total approximatif	60" (1')	3420" (57')
Temps de résolution continue	-	877"
Temps de résolution entière	-	2887"
Gap objectif	-	0%
Valeur de <i>invret</i>	117,60	92,50
Gain sur <i>invret</i> par rapport à PRESTO	-	21%
Coûts actifs (en km)	45646	45646
Coûts passifs (en km)	1229	2599
Coûts à vide (en km)	11865	11665

TABLE 8.2 – Premiers résultats de PRESTO et de (MPT) sans maintenance sur l'instance cible 3

Les premières observations sont similaires aux précédentes concernant les instances 1 et 2.

Le gain en robustesse est intéressant puisqu'on réduit de 21% la valeur de *invret*.

Les coûts des déplacements à vide sont moins élevés que dans la solution de PRESTO, mais les coûts en passif sont dégradés. Si on somme les kilomètres en passif et à vide comme s'ils étaient équivalents⁶, l'approche (MPT) dégrade les coûts de moins de 9%.

Par ailleurs, en considérant les coûts en actif, la dégradation atteint 2% des coûts de la solution de PRESTO.

Le temps de calcul d'une heure à peine reste raisonnable mais pourrait poser problème en cas d'industrialisation. On remarque cependant que la résolution de la relaxation continue à la racine prend moins de 15 minutes. Or, de même que pour les instances 1 et 2, le gap est nul et la solution continue optimale est très fractionnaire. On peut également envisager une autre méthode de résolution en entiers à partir de la solution continue.

8.4.3 Premières conclusions

L'approche (MPT) permet un gain significatif en termes de robustesse pour le critère considéré (*invret*).

6. En réalité, les déplacements à vide sont souvent plus coûteux : dans PRESTO, les kilomètres à vide sont deux fois plus chers que les kilomètres en passif. En prenant en compte cette équivalence, l'approche (MPT) dégrade les coûts (en passif et à vide) de moins de 4%.

Les coûts de production sont globalement intéressants, puisque les coûts d'utilisation et les coûts en actif sont constants, et les coûts des déplacements à vide sont contrôlés.

Mais l'absence de pénalisation des évolutions peut être critiquée puisqu'elle permet l'augmentation des coûts à vide (cf. instance 2 dans la Table 8.1) : la pénalisation est possible en théorie mais difficile à gérer en pratique (temps de calcul beaucoup plus longs).

De même, les coûts en passif ne sont pas contrôlés et peuvent ainsi doubler par rapport aux solutions de PRESTO. Leur pénalisation dans l'objectif complique également la résolution.

L'approche (MPT) permet l'intégration de la maintenance lors de la construction du roulement et donne ainsi des solutions avec des créneaux réguliers. En revanche, elle ne propose pas de solution si les contraintes exprimées ne peuvent pas être respectées, dans ce cas les paramètres doivent être reconfigurés.

Le temps de calcul dépend des tests (instance et paramètres choisis) mais reste raisonnable. Le gap entre les solutions continue (à la racine) et entière (optimale) très faible voire nul laisse envisager l'utilisation d'une méthode plus rapide pour trouver une solution entière proche de l'optimum.

8.4.4 Industrialisation et perspectives

De même que les approches précédentes, l'approche (MPT) est intégrée à PRESTO et ne demande pas de modification des formats d'entrée ou de sortie.

L'approche (MPT) répond au problème de planification robuste des roulements d'engins en remettant en cause les déplacements à vide et passifs proposés par PRESTO. L'approche est globale, permettant de traiter simultanément la construction d'un roulement, les critères de robustesse et les contraintes de maintenance.

Puisqu'il s'agit d'un prototype dédié à des expérimentations, quelques améliorations seraient utiles pour rendre le module plus ergonomique (choix de certains paramètres \overline{pm} , critères de robustesse en sortie).

La principale faiblesse de l'approche utilisée est le temps de résolution.

Par ailleurs, on constate que pour l'instance cible 3, il n'est pas possible de respecter la maintenance avec le nombre d'engins imposé. Dans ce cas, la recherche d'une solution admissible avec maintenance requiert une procédure faisant varier les paramètres en cause (\overline{pm} , $ncmin$, $ncmax$, voire le nombre d'engins) non automatisée. On note que PRESTO ne permet pas non plus d'obtenir une solution avec maintenance.

Plusieurs pistes sont envisageables pour améliorer le modèle ou sa résolution.

Dans l'optique d'une industrialisation, il faudrait probablement mettre en place une résolution entière (exacte ou approchée) plus rapide à partir de la solution continue.

Plutôt que de pénaliser les évolutions selon le nombre de kilomètres, on peut interdire celles qui sont trop coûteuses (distance supérieure à un seuil en kilomètres). On peut cependant se trouver confronté à un problème de réalisabilité, en particulier dans le cas d'un technicentre unique.

Le modèle présenté à la section 8.2.7 prend en compte un unique technicentre. Dans le cas de plusieurs technicentres pour un même type d'engin, on peut considérer qu'après une tâche i , un créneau de maintenance se fait dans le technicentre le plus proche de G_i^{arr} .

Enfin, il faudrait étudier la possibilité de prendre en compte exactement les contraintes couplées concernant tâches et sillons à vide couverts par des engins de type différent : capacités

(nombre d'engins maximum) et recompositions. En effet, le problème est divisé en plusieurs sous-problèmes (un par type d'engin) traités indépendamment. Or on considère une tâche pouvant être couverte par une UM composée d'engins de type 1 et de type 2. Dans le sous-problème de type 1, le nombre d'engins de type 2 couvrant la tâche n'est pas connu. On ne peut donc pas connaître le nombre d'engins maximum de type 1 pouvant couvrir la tâche. De même, les compositions ou décompositions d'un engin de type 1 et d'un engin de type 2 ne seront pas prises en compte. Il s'agit bien de contraintes couplantes devant être gérées.

Troisième partie

Expérimentations

Dans la partie précédente (Partie II), nous avons proposé trois approches répondant à trois problèmes de planification robuste de roulements d'engins distincts. Elles ont été soumises à des tests fonctionnels et validées indépendamment les unes des autres. Dans cette partie, il s'agit de comparer les solutions obtenues par ces différentes approches en termes de coûts de production et de robustesse. On utilise dans ce but un outil indépendant simulant le comportement d'un roulement d'engins soumis à des retards et permettant de calculer tous les indicateurs d'évaluation et de construction énumérés à la section 4.2. Par cette démarche, les solutions sont comparées dans des conditions identiques (mêmes scénarios de retards, mêmes indicateurs), et on peut évaluer l'intérêt de chaque approche en termes de robustesse.

Chapitre 9

Evaluation de la robustesse de roulements d'engins par simulation

Les différentes méthodes présentées en Partie II permettent de calculer des roulements d'engins en répondant à des problématiques un peu différentes : les critères pris en compte dans l'objectif et les données imposées (sillons à vide et flots) ne sont pas les mêmes d'une méthode à l'autre. Par ailleurs, il existe d'autres solutions, en particulier celle de PRESTO ou toute autre solution issue du métier directement. Il n'est pas évident de comparer toutes ces solutions, on cherche donc à simuler le comportement de chacune lors de l'apparition d'aléas.

9.1 Principe

L'objectif de l'outil de simulation est de pouvoir évaluer la robustesse de roulements d'engins existants. Dans ce but, on génère d'abord des retards sources sur certains trajets afin de modéliser les incidents (cf. section 9.2). On simule ensuite le comportement d'une solution en situation perturbée, il s'agit principalement de propager le retard généré. Puis on calcule tous les indicateurs d'évaluation, y compris les indicateurs ne pouvant être calculés qu'a posteriori, en distinguant les valeurs théoriques et "réalisées" de certains indicateurs. Ce système permet de comparer plusieurs solutions existantes en termes de robustesse : des roulements d'engins construits par des concepteurs, par PRESTO ou par une méthode avec robustesse présentée en Partie II.

9.2 Génération de scénarios de retards

Afin de générer des scénarios relativement réalistes, on exploite les données de la base BREHAT du système ferroviaire français, qui recense tous les horaires théoriques et réalisés des trains. Pour analyser cette base de données, on utilise une méthode de stratification permettant de classer les trains en catégories selon leur retard (pas de retard, retard inférieur à deux minutes, retard compris entre deux et cinq minutes, etc.), leur période (par exemple période de pointe), et leur zone d'appartenance. À partir des strates ainsi créées, on peut générer des scénarios de retards par une méthode de sondage stratifié. Un scénario consiste en un ensemble de retards sources, chaque retard étant modélisé par un poids (nombre de minutes) affecté à une tâche. Nous avons testé deux méthodes de sondage (Chandesris, 2005) :

- l'allocation proportionnelle, qui permet la génération de scénarios représentatifs ;

- une allocation pseudo-optimale, afin de couvrir au mieux les différentes configurations d’incidents possibles.

Base de données BREHAT

La base de données BREHAT recense des observations. Chaque observation correspond au passage d’un train (ou circulation) en un point remarquable (par exemple une gare) du réseau. Les informations qui nous intéressent sont les suivantes :

- numéro de la circulation : il permet d’identifier le train ;
- date et heure ;
- type horaire : origine, terminus (destination), départs et arrivées (qui concernent les dessertes, c’est-à-dire les gares intermédiaires d’un train) ;
- écart horaire : le nombre de minutes d’avance ou de retard (il peut être négatif pour exprimer une avance) ;
- codes gares : codes permettant d’identifier une gare.

Ces données sont ensuite triées par numéro de circulation croissant pour regrouper toutes les observations concernant un même train, puis dans l’ordre chronologique (par date et heure). On a donc pour chaque train l’ordre des gares dans lesquelles il s’est arrêté et pour chaque arrêt la date ainsi que l’heure d’arrivée et de départ. On récupère alors les informations sous forme d’un tableau. L’exemple du tableau 9.1 correspond au train représenté en Figure 9.1.

Numéro de circulation	Date et heure	Type horaire	Écart horaire	Codes gares
550	01/01/2008 00 :01	Arrivée	-2	001 G0A
550	01/01/2008 00 :02	Départ	0	001 G0A
550	01/01/2008 00 :17	Arrivée	0	002 G0B
550	01/01/2008 00 :18	Départ	3	002 G0B
550	01/01/2008 00 :17	Terminus	10	003 G0C

TABLE 9.1 – Exemple de données récupérées de BREHAT

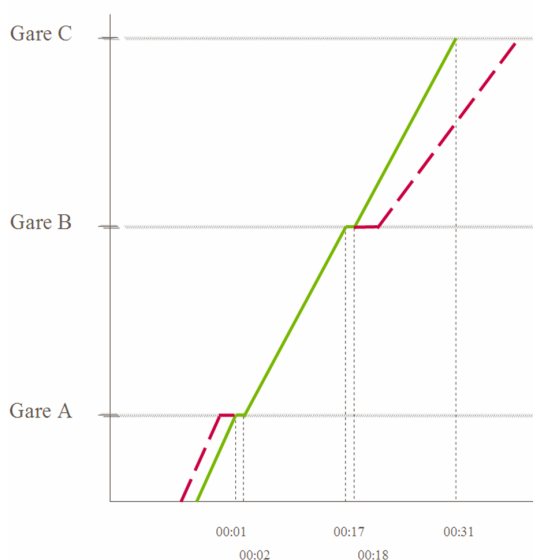


FIGURE 9.1 – Exemple : un extrait du train 550, en vert le sillonn théorique, en rose le réalisé.

Stratification

À partir des données récupérées triées, on cherche à construire des strates reflétant la fréquence d'apparition de chaque type de retard.

Dans le tableau précédent, chaque paire (origine/départ, arrivée/terminus) de deux lignes qui se suivent correspond à une tâche et à chaque tâche correspond un retard dit indépendant ou retard source. Un retard indépendant est la conséquence directe d'un incident et non la propagation d'un retard sur un train précédent, par opposition au retard propagé. Pour simplifier, on considère que le retard d'un train à l'arrivée est la somme du retard propagé (issu d'un retard précédent) et du retard indépendant (pris pendant le trajet). Puisque le retard indépendant est pris durant le trajet, le retard au départ correspond au retard propagé. Pour calculer le retard indépendant sur une tâche, on soustrait son retard au départ à son retard à l'arrivée.

Dans l'exemple du tableau 9.1, les deux dernières lignes correspondent à une tâche de la gare 002 G0B à 003 G0C ayant 3 minutes de retard au départ et 10 à l'arrivée. Le retard indépendant sur cette tâche est donc de 7 minutes. De plus, la tâche est associée à une zone d'après ses gares de départ et d'arrivée, et à une période d'après la date et l'heure (dans cet exemple on peut imaginer que la tâche pourrait être affectée à une période creuse).

On obtient donc un nouveau tableau à trois colonnes recensant tous les retards indépendants et pour chacun la zone et la période d'appartenance.

On classe alors les retards en plusieurs catégories : négatif ou nul, inférieur à deux minutes, entre deux et cinq minutes, etc. En pratique, les retards négatifs seront ramenés à zéro. La dernière catégorie concerne les retards supérieurs à un seuil fixé (15 minutes dans l'exemple du tableau 9.2) à partir duquel on considère que la solution devra être modifiée (on ne cherche donc pas à y être robuste). Et pour chaque paire (zone, période), on construit des strates en calculant la fréquence d'apparition des retards de chaque catégorie. La forme de ces strates pour une zone et pour une période données est illustrée par le tableau 9.2.

Retard indépendant (min)	Fréquence d'apparition (%)
0	70
]0, 2]	18
]2, 5]	5
]5, 10]	2
]10, 15]	3
> 15	2

TABLE 9.2 – Exemple de strates pour une zone et pour une période fixées

Fréquences efficaces

Les strates ainsi construites représentent l'historique réalisé. Ces fréquences représentatives sont utilisées dans le cas d'une allocation proportionnelle. Elles permettent de générer des scénarios réalistes. Or pour construire des solutions plus robustes ou pour mettre à l'épreuve des solutions existantes, il peut être plus intéressant d'utiliser des fréquences dites efficaces, qui permettent de tester plus efficacement les différentes configurations de retards possibles. Plusieurs méthodes d'allocation existent, comme l'allocation optimale de Neymann (cf. Chandesris (2005)). Nous choisissons de modifier les strates manuellement selon le besoin. Pour l'exemple 9.2, nous proposons les strates du tableau 9.3, qui "gonflent" un peu les retards mais restent relativement réalistes. Dans l'idée d'analyser un "pire cas", les strates de la Table 9.4 peuvent être utilisées pour générer beaucoup de retard.

Retard	Fréquence
0	60
]0, 2]	18
]2, 5]	8
]5, 10]	4
]10, 15]	6
> 15	4

TABLE 9.3 – Exemple 1 de strates efficaces

Retard	Fréquence
0	50
]0, 2]	10
]2, 5]	10
]5, 10]	10
]10, 15]	10
> 15	10

TABLE 9.4 – Exemple 2 de strates efficaces

Tirage de retards

Une fois les strates construites, nous pouvons générer des scénarios de retards. Le principe est de tirer des retards de façon à respecter les strates choisies.

Pour chaque tâche T_i , on détermine la zone et la période d'appartenance afin de se référer aux strates correspondantes. On tire alors aléatoirement un nombre entre 0 et 100, qui correspond à une strate. Par exemple, un nombre entre 0 et 70 correspond à la première strate de la Table 9.2, c'est-à-dire à un retard nul, tandis qu'un nombre entre 96 et 98 correspond à un retard de 10 à 15 minutes. On tire ensuite aléatoirement un retard dans l'intervalle correspondant. Ce retard est affecté à la tâche T_i : il s'agit du retard indépendant $Rind_i$ pris par les engins lors de la réalisation de la tâche. Il sera pris en compte à l'arrivée.

On obtient ainsi un scénario de retards pour la simulation.

9.3 Simulation de la propagation des retards

Principe

En utilisant le simulateur sur un roulement d'engins pour un scénario donné, on cherche à répondre à la question suivante : "Comment pourrait se comporter ce roulement d'engins en situation perturbée s'il n'était pas modifié?". En effet, aucune intervention humaine, aucune décision au niveau opérationnel (en temps réel) n'est prise en compte ici. On met à l'épreuve un roulement tel qu'il a été conçu afin d'en tester la robustesse en termes de résistance et d'autorégulation.

Dans cette optique, on propage simplement le retard généré par tirage stratifié sur les tâches. Soit un engin k circulant sur une tâche T_i arrivant en gare G_i^{arr} avec un retard Ra_i . Il doit ensuite couvrir la tâche T_j . Plusieurs cas de figure peuvent se présenter :

- k est en US (un seul engin circulant) sur T_j et le retard est inférieur au temps de stationnement en gare (diminué du temps de crochet obligatoire), soit $Ra_i \leq t_{i,j} - tc(G_i^{arr})$: le retard est absorbé lors du stationnement en gare et l'engin k effectue la tâche T_j sans retard au départ (plan de transport résistant) ;
- k est en US sur T_j et le retard est supérieur au temps de stationnement en gare, soit $Ra_i > t_{i,j} - tc(G_i^{arr})$: le retard n'est que partiellement absorbé et l'engin k effectue la tâche T_j avec un retard au départ (ou retard propagé) $Rp_j = Ra_i - (t_{i,j} - tc(G_i^{arr}))$;
- k est en UM sur T_j : le retard au départ de T_j est le maximum des retards propagés de tous les engins en UM.

Afin d'obtenir une formule générale pour le retard propagé, on note $\delta^-(T_j)$ l'ensemble des tâches T_i telles qu'il existe un engin enchaînant T_i et T_j . T_j appartient alors à l'ensemble $\delta^+(T_i)$ des successeurs de T_i . On peut écrire :

$$Rp_j = \max_{T_i \in \delta^-(T_j)} (Ra_i - (t_{i,j} - tc(G_j^{dep})))$$

Le retard propagé ainsi calculé est le retard au départ de T_j . Le retard à l'arrivée de la tâche T_j est alors ce retard au départ (lié aux tâches précédentes) augmenté du retard indépendant (tiré aléatoirement) pris lors de sa réalisation, soit :

$$Ra_j = Rp_j + Rind_j$$

En suivant ce principe, on propage alors les retards générés par tirage stratifié. Sur l'exemple de la Figure 9.2, deux engins 1 et 2 couvrent trois tâches T_1, T_2 et T_3 auxquelles ont été affectés des retards (entre parenthèses sur les schémas). Le schéma de gauche représente les parcours prévus des deux engins, et celui de droite les parcours réalisés (en cas de réalisation du scénarios de retards indiqué).

L'engin 1 prend 12 minutes de retard sur la tâche T_2 . En gare, le temps de retournement théorique est de 13 minutes, et l'engin doit stationner au moins 7 minutes (temps de crochet). Le retard n'est donc que partiellement absorbé et l'engin 1 effectue alors la tâche T_3 avec un retard de 6 minutes au départ. L'engin 2 ne prend pas de retard sur T_1 , il pourrait donc effectuer T_3 sans retard au départ. Mais T_3 est couverte par les engins 1 et 2 en UM, l'engin 2 prend donc le même retard au départ que l'engin 1 (6 minutes). Pendant le parcours, les engins prennent encore 2 minutes de retard supplémentaires. Ils arrivent donc avec un retard de 8 minutes.

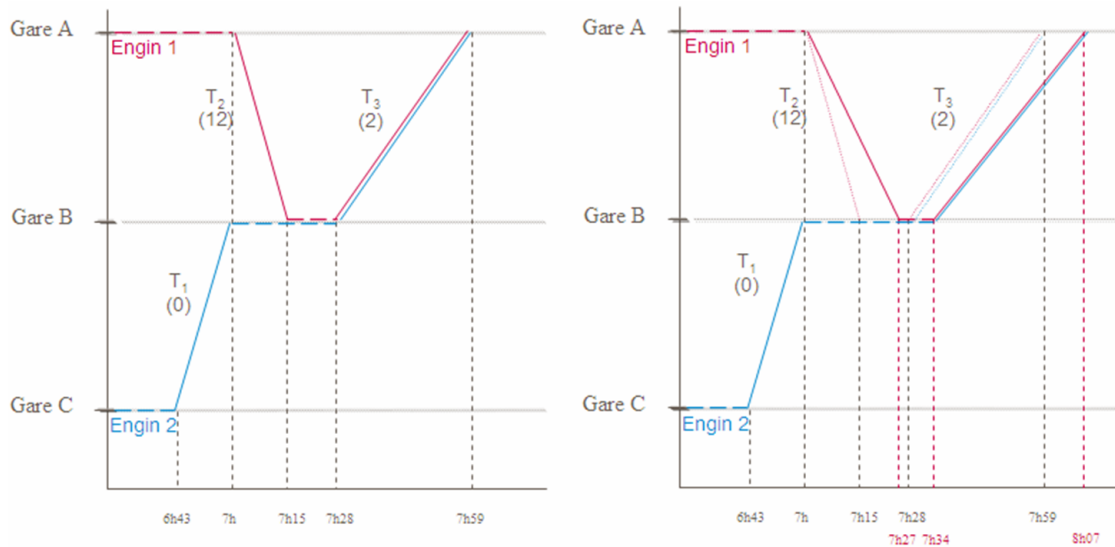


FIGURE 9.2 – Propagation de retards : parcours théoriques (à gauche) et réalisés (à droite)

Variantes

En pratique, lorsqu'il y a un retard au départ d'une tâche, ce retard a souvent une influence sur le temps de parcours. D'une part, l'agent de conduite peut chercher à rattraper tout ou partie du retard. D'autre part, un train en retard n'est en général pas prioritaire et peut avoir à laisser passer d'autres trains, aggravant son retard initial. Pour prendre en compte ce phénomène, deux variantes du système de propagation du retard sont mises en place.

Une première variante rattrape systématiquement une partie du retard exprimée sous la forme d'un pourcentage du temps de parcours fixé à l'avance. Par exemple, on considère que tous les

trains peuvent rattraper leur retard à hauteur de 10% du temps de parcours théorique. Dans le cas de la Figure 9.2, T_3 dure 31 minutes. 3 minutes environ peuvent donc être rattrapées pendant le trajet. À l'arrivée, les engins n'auraient plus que 5 minutes de retard au lieu de 8.

Au contraire, la seconde variante aggrave systématiquement tous les retards à hauteur d'un pourcentage du temps de parcours prévu. Cette variante peut être considérée comme un "pire cas" pour la réalisation d'un scénario donné.

9.4 Calcul des indicateurs

L'évaluation de la robustesse d'une solution se fait par le calcul des différents indicateurs de robustesse sélectionnés.

Certains indicateurs sont liés à la structure d'une solution. Par exemple, le nombre de recompositions ou les temps de retournement sont immuables pour une solution donnée. En revanche, d'autres indicateurs liés aux retards (régularité, retard propagé) ne peuvent pas être calculés a priori. Simuler le comportement d'une solution par propagation du retard permet le calcul de tels indicateurs. On rappelle ici les indicateurs en deux catégories :

- indicateurs pouvant être calculés a priori :
 - homogénéité des temps de retournement (*invret*) ;
 - répartition de la charge de travail entre les engins (durée de l'ensemble des tâches, *chge*) ;
 - temps de retournement minimum (*tmin*) ;
 - nombre de recompositions (*recomp*) ;
- indicateurs devant être calculés par simulation :
 - régularité (*reg*) ;
 - retard propagé cumulé (*retprop*) ;
 - retard en fin d'horizon temporel (*retHT*) ;
 - temps de retour à la normale (en pratique temps en situation perturbée, *TRN*).

Ces indicateurs ont été détaillés à la section 4.2.

Les valeurs de certains indicateurs dépendent du scénario qui se réalise : elles peuvent être différentes d'un scénario à l'autre. Pour avoir des valeurs représentatives, il est nécessaire de simuler le comportement d'une solution pour plusieurs scénarios. Une question importante a alors été soulevée : sur combien de scénarios faut-il tester une solution pour obtenir des résultats fiables ? Nous apporterons des éléments de réponse à cette question à la section 9.5.

9.5 Fiabilité des résultats

Le paramètre principal de l'outil de simulation proposé est le nombre de scénarios à tester. En effet, l'ensemble des scénarios testés doit être suffisamment représentatif de l'ensemble complet des scénarios réalisables (selon les strates choisies) pour obtenir des indicateurs fiables. Trop peu de scénarios peut mener à des indicateurs erronés. Trop de scénarios est inutile et allonge le temps d'exécution.

Afin de choisir un nombre de scénarios à tester pertinent, nous avons lancé une simulation pour l'instance cible 1 sur 5000 scénarios. Nous avons alors étudié la moyenne et l'écart-type de différents indicateurs sur des sous-ensembles de scénarios.

Le premier indicateur analysé est la régularité (*reg*). Parmi les 5000 scénarios testés, nous avons sélectionné 10 sous-ensembles de 10 scénarios chacun. Pour chaque sous-ensemble, nous avons calculé la moyenne et l'écart-type de la régularité. Nous avons ensuite reproduit la procédure pour des sous-ensembles de 20, 50, 100, 200 et 500 scénarios. Les valeurs de la moyenne et de l'écart-type sont utilisées comme valeurs de référence pour l'ensemble complet des 5000 scénarios. L'ensemble de ces valeurs est contenu dans la Table A.1 de l'Annexe A.

L'objectif de cette procédure est de déterminer la variabilité des valeurs des indicateurs en fonction du nombre de scénarios considérés. Si plusieurs sous-ensembles de S scénarios donnent des valeurs très différentes pour un même indicateur, cet indicateur est très variable. La valeur donnée par un sous-ensemble de S scénarios n'est donc pas fiable. Au contraire, si la valeur de l'indicateur varie peu d'un sous-ensemble de S scénarios à l'autre, chaque sous-ensemble est suffisamment représentatif de l'ensemble des scénarios réalisables. La Figure 9.3 permet d'observer la variabilité de la régularité.

Le second indicateur présenté ici est le retard propagé cumulé (*retprop*). De même que pour la régularité, on analyse la variabilité des valeurs moyennes (et moyennes augmentées ou diminuées de l'écart-type) en fonction du nombre de scénarios considérés. La Figure 9.4 permet d'observer cette variabilité. L'ensemble des valeurs est contenu dans la Table A.2 de l'Annexe A.

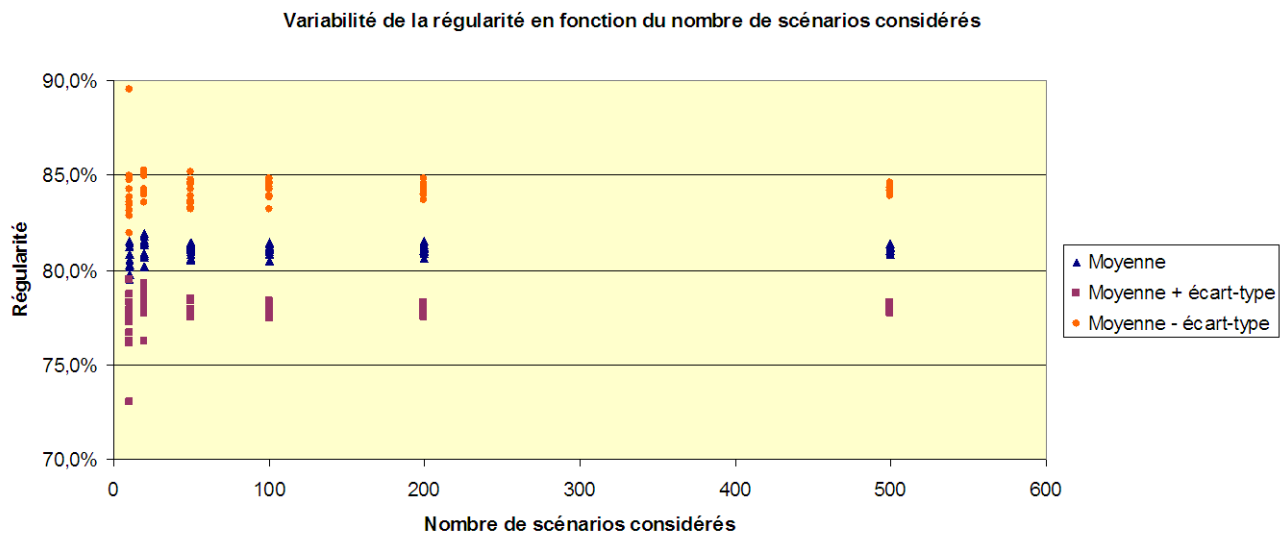


FIGURE 9.3 – Variabilité de la régularité moyenne en fonction du nombre de scénarios considérés.

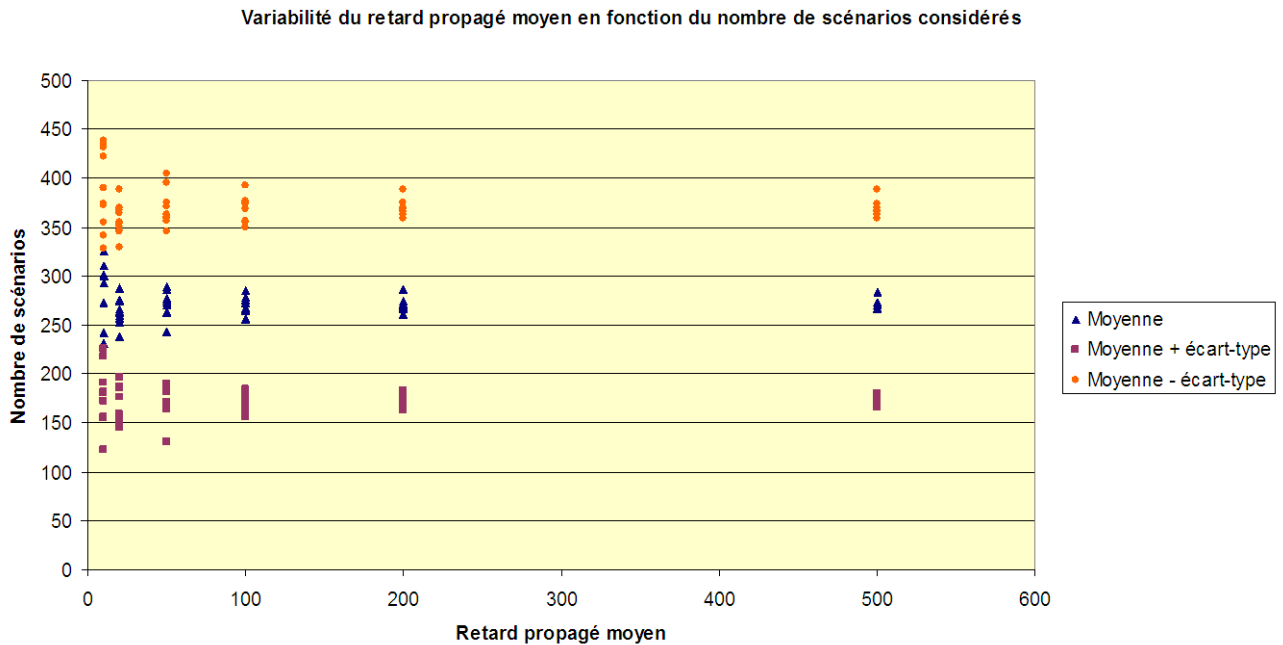


FIGURE 9.4 – Variabilité du retard propagé moyen en fonction du nombre de scénarios considérés.

Pour un nombre de scénarios fixé, on s'intéresse à la variabilité des valeurs : moyenne et moyenne augmentée ou diminuée de l'écart-type. Pour les sous-ensembles de 10 scénarios, la régularité moyenne varie entre 79,5% et 81,5%, tandis que pour des sous-ensembles de 100 scénarios, elle est comprise entre 80,4% et 81,4% ; la moyenne sur les 5000 scénarios testés est de 81,1% (cf. Table A.1). Il semble donc que les valeurs obtenues sur des ensembles de 100 scénarios soient plus fiables que pour des ensembles de 10 scénarios.

Le graphique de la Figure 9.3 qui représente la variabilité de la régularité, permet de confirmer cette impression : plus le nombre de scénarios d'un ensemble est élevé, moins la régularité varie et pour des ensembles de 100 scénarios, les valeurs obtenues paraissent fiables.

On observe la même tendance pour le retard propagé cumulé. D'après le graphique de la Figure 9.4, les valeurs moyennes du retard propagé (cumulé) pour des sous-ensembles de 10 ou 20 scénarios sont variables et donc potentiellement peu représentatives. Par exemple, le retard propagé moyen sur 10 scénarios varie entre 232 et 326 minutes selon le sous-ensemble choisi.

À partir de 50 et surtout 100 scénarios, les valeurs obtenues sont plus fiables : le retard propagé moyen sur 100 scénarios varie entre 255 et 285 minutes, soit une amplitude de 30 minutes, diminuée de 68% par rapport à l'amplitude de 94 minutes pour des sous-ensembles de 10 scénarios. La valeur moyenne de référence (sur les 5000 scénarios testés) est de 271 minutes. Le même constat est possible pour les valeurs moyennes augmentées ou diminuées de l'écart-type.

Les valeurs des indicateurs calculés varient d'un ensemble de 10 ou 20 scénarios à l'autre. À partir de 50 et surtout 100 scénarios, les valeurs des indicateurs sont plus fiables. Pour les tests d'évaluation par simulation effectués au Chapitre 10, on choisit donc de tester chaque solution sur 100 scénarios.

9.6 Limites de la méthode

Le modèle de propagation du retard proposé est très simplifié.

D'une part, la distinction entre le retard propagé et le retard indépendant lors de la phase de stratification n'est pas fidèle à la réalité. En effet, on considère que le retard propagé se limite au retard au départ d'une tâche. Or, comme expliqué précédemment, un train partant en retard risque de rencontrer plus d'imprévus et de prendre plus de retard encore qu'un train parti à l'heure. Le retard supplémentaire à l'arrivée est donc souvent une conséquence du retard au départ et devrait donc faire partie du retard propagé. Il faudrait utiliser des méthodes statistiques pour estimer plus précisément les retards propagé et indépendant.

D'autre part, certains incidents ne sont pas pris en compte dans la façon de calculer les retards. En effet, un retard au départ d'un train n'est pas forcément dû à un retard précédent, mais peut être la conséquence directe d'un incident en gare. En négligeant le retard au départ lors du calcul des retards sur les tâches, on ignore donc tous les incidents survenus en gare. Un moyen de remédier à cette imprécision serait d'affecter des retards aux stationnements en gare comme aux tâches. Il faudrait dans ce cas définir le calcul dudit retard, à quoi ce retard serait affecté (à tous les engins stationnant ou à un seul), etc.

D'autres imprécisions sont introduites dans le modèle par sa simplicité. Mais vouloir les corriger demande un travail important sans garantie de plus de réalisme : on ne sait pas exploiter les données pour faire la différence entre le retard indépendant et le retard lié à des événements précédents, tandis qu'un modèle de propagation simple est déjà intéressant et adapté à nos besoins. On cherche principalement à comparer plusieurs solutions sur un certain nombre de scénarios. Les imperfections introduites par la simplicité du modèle sont donc les mêmes pour toutes les solutions testées.

Chapitre 10

Analyse des résultats

Dans le but de construire des roulements d'engins robustes, nous avons proposé trois méthodes décrites à la Partie II répondant à des problèmes comparables mais distincts. Ce chapitre a alors deux principaux objectifs. D'une part, nous cherchons à mettre en évidence les avantages et les inconvénients de chaque méthode en comparant les solutions obtenues les unes aux autres. D'autre part, nous aimerions évaluer l'intérêt d'une démarche robuste en comparant les résultats obtenus par ces approches aux solutions existantes. Les tests sont effectués sur neuf instances de référence à l'aide de l'outil de simulation indépendant présenté au Chapitre 9.

10.1 Informations préliminaires

10.1.1 Environnement

Les tests présentés ont été effectués sur une machine composée d'un Intel Core i7-2600 CPU, avec 8 Go de RAM, sous Windows XP 64 bits.

10.1.2 Instances de référence

Pour assurer une certaine diversité et ainsi comparer plus justement les différentes approches, nous avons sélectionné neuf instances de référence représentatives de l'échantillon d'instances réelles à notre disposition. Parmi elles, nous retrouvons les trois instances cibles présentées à la section 5.6. Les caractéristiques principales des neuf instances sont renseignées dans le tableau 10.1 :

- l'identifiant ;
- la région ;
- la série d'engin (ou type d'engin) ;
- le nombre de tâches à couvrir ;
- le nombre de tâches couvertes par PRESTO (couverture maximale) ;
- le nombre d'engins utilisés dans la solution de PRESTO (nombre minimum).

<i>Instance</i>	Région	Série d'engin	Nombre de tâches à couvrir	Nombre de tâches couvertes	Nombre d'engins minimum
1	Bretagne	B82500	208	207	7
2	Rhône-Alpes	BB67300	314	314	11
3	Bretagne	Z27500 3C	624	597	15
4	Bretagne	X2100	168	166	11
5	Bretagne	Z27500 3C	307	241	6
6	Rhône-Alpes	Z27500 3C	352	352	7
7	Alsace	B82500	428	372	6
8	Bretagne	Z21500	526	482	15
9	Rhône-Alpes	Z27500 4C	778	763	15

TABLE 10.1 – Instances de référence

On note que pour chaque instance, il existe un roulement droit. En effet, la non connexité du graphe avec arcs-retours issus du graphe de PRESTO (cf. Chapitre 2) est un cas particulier peu fréquent.

10.2 Comparaison des différentes approches sans maintenance selon le critère *invret*

Nous comparons d'abord PRESTO, (HGL), (CGC) et (MPT) sur le critère *invret* optimisé par les trois dernières approches, qui intègrent la robustesse.

L'approche (HGL) est limitée à 5 minutes (cf. section 6.4.1) et l'approche (MPT) est paramétrée par défaut (cf. section 8.4.1).

Les contraintes de maintenance ne sont pas prises en compte pour les tests de cette section.

La Table 10.2 recense les valeurs du critère *invret* des solutions obtenues par chaque méthode sur toutes les instances de référence. La valeur de *invret* ne varie pas d'un scénario à un autre pour une même solution, les valeurs obtenues sont donc issues de la simulation sur un unique scénario sans retards. Le gain relatif par rapport à PRESTO est indiqué pour les trois approches présentées au Chapitre II. Les temps d'exécution correspondants sont présentés dans la Table 10.3¹.

1. Pour l'instance 9, la résolution par la méthode (CGC) a été interrompue après trois jours d'exécution.

Méthode	PRESTO	(HGL)		(CGC)		(MPT)	
Critère / Instance		Valeur de <i>invret</i>	Gain relatif	Valeur de <i>invret</i>	Gain relatif	Valeur de <i>invret</i>	Gain relatif
Instance 1	9,18	7,92	13,7%	7,92	13,7%	4,09	55,4%
Instance 2	10,52	7,27	30,9%	7,28	30,8%	5,85	44,4%
Instance 3	117,60	113,97	3,1%	117,10	0,4%	92,50	21,3%
Instance 4	10,94	9,84	10,1%	9,84	10,1 %	4,37	60,1%
Instance 5	14	13,38	4,4%	13,38	4,4%	7,40	47,1%
Instance 6	23,72	21,34	10,0%	21,40	9,8%	19,61	17,3%
Instance 7	27,42	24,88	9,3%	24,96	9,0%	21,06	23,2%
Instance 8	45,10	42,40	6,0%	45,10	0,0%	28,75	36,3%
Instance 9	92,89	88,48	4,7%	-	-	71,01	23,6%
Gain moyen	-	-	10,2%	-	8,7%	-	36,5%

TABLE 10.2 – Valeurs de *invret* des solutions obtenues par chaque méthode sur toutes les instances de référence sans prise en compte de la maintenance

Méthode / Instance	PRESTO	(HGL)	(CGC)	(MPT)
Instance 1	5	10	12	6
Instance 2	< 1	5	83	6
Instance 3	< 1	5	6	57
Instance 4	< 1	5	2	2
Instance 5	< 1	5	3	1
Instance 6	15	20	27	16
Instance 7	< 1	5	112	5
Instance 8	< 1	5	19	37
Instance 9	< 1	5	-	263

TABLE 10.3 – Temps d'exécution (en minutes) des différentes méthodes pour chaque instance de référence sans prise en compte de la maintenance

Pour plus de lisibilité, on peut se référer à la Figure 10.1 :

- le graphique de gauche représente les valeurs de *invret* pour chaque instance selon la méthode utilisée;
- le graphique de droite représente le gain sur *invret* par rapport à PRESTO pour chaque approche proposée.

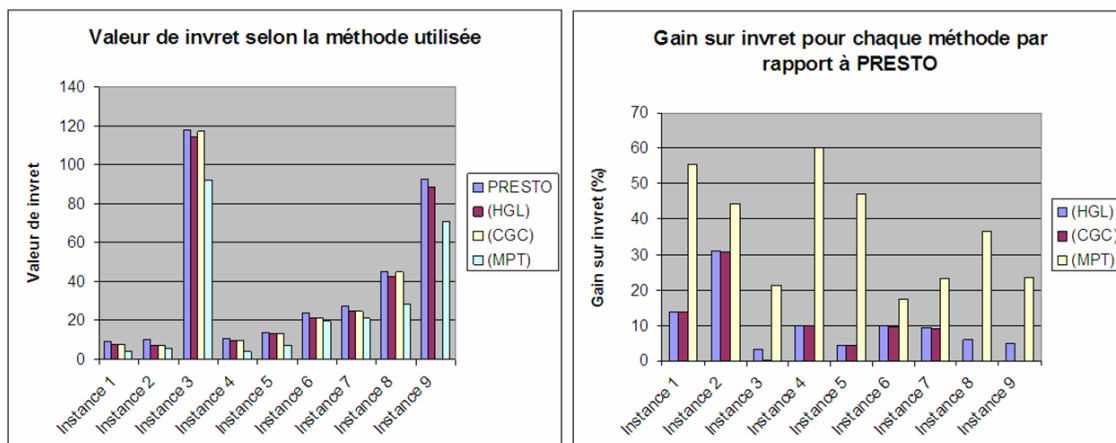


FIGURE 10.1 – Valeur et gain sur *invret* selon la méthode utilisée.

Pour les trois approches proposées (HGL), (CGC), (MPT), on constate un gain significatif par rapport à PRESTO sur le critère *invret* (cf. Table 10.2). (MPT) présente un gain beaucoup plus important que les deux autres approches : 37% en moyenne tandis que (HGL) et (CGC) avoisinent les 10%. C'est donc l'approche qui semble la plus intéressante. Au contraire, le gain obtenu par l'approche (CGC) est toujours inférieur (ou égal) à celui de (HGL).

Ces gains sont obtenus à coûts constants par rapport aux solutions de PRESTO. En effet, les approches (HGL) et (CGC) sont bridées pour respecter les coûts de PRESTO. L'approche (MPT) n'intégrant pas la maintenance dans ces tests, les coûts à vide sont minimisés, approchant ceux calculés par PRESTO (il existe souvent une différence de quelques kilomètres, on trouve en général quelques kilomètres en passif supplémentaires mais moins à vide).

En observant la Table 10.3, on remarque que l'approche (HGL) est rapide puisqu'en 5 minutes elle permet d'obtenir des solutions intéressantes, c'est-à-dire présentant un gain significatif sur le critère choisi (*invret*). En revanche, les approches (CGC) et (MPT) ont un temps de résolution plus élevé dans la plupart des cas.

L'approche (HGL) permet d'obtenir des solutions intéressantes rapidement.

L'approche (MPT) fournit des solutions présentant le gain le plus important sur le critère *invret*, en un temps qui reste raisonnable.

En revanche, l'approche (CGC) est moins performante en termes de robustesse et peut être très longue. Par ailleurs, seul le critère *invret* est intégré : la maintenance et les autres critères ont été modélisés mais n'ont pas été implémentés. En effet, d'après ces premiers résultats et quelques essais avec le critère *tmin*, nous avons choisi de ne pas poursuivre les développements sur cette méthode : elle n'apparaît plus dans la suite des résultats.

10.3 Corrélation entre les critères optimisés : *invret* et *recomp*

L'approche (HGL) nécessite la définition d'une hiérarchie des critères de robustesse. Nous cherchons ici à montrer l'importance du choix des critères à travers l'exemple de deux critères optimisés : *invret* et *recomp*. La Table 10.4 présente pour chaque instance les valeurs des deux critères pour deux configurations de la hiérarchie des critères :

- *invret* est prioritaire, *recomp* secondaire ;

– *recomp* est prioritaire, *invret* secondaire.

Hiérarchie Critère / Instance	<i>invret</i> prioritaire		<i>recomp</i> prioritaire	
	valeur de <i>invret</i>	valeur de <i>recomp</i>	valeur de <i>invret</i>	valeur de <i>recomp</i>
Instance 1	7,92	8	7,94	8
Instance 2	7,27	0	7,27	0
Instance 3	113,97	44	114,10	44
Instance 4	9,84	70	10,15	64
Instance 5	13,38	16	13,38	16
Instance 6	21,34	16	21,37	16
Instance 7	24,88	42	25,03	38
Instance 8	42,40	134	43,71	112
Instance 9	88,48	208	89,07	192

TABLE 10.4 – Valeurs des critères *invret* et *recomp* selon la hiérarchie de critères utilisée.

D’après les valeurs de la Table 10.4, utiliser *recomp* comme premier critère d’optimisation semble minimiser significativement le nombre de recompositions mais dégrade partiellement la valeur de *invret*. Les deux critères sont donc corrélés négativement : l’optimisation de l’un a tendance à dégrader l’autre.

Cependant, pour la seconde hiérarchie (*recomp* prioritaire), la dégradation de *invret* paraît faible comparée au gain sur *recomp*. Par exemple, pour l’instance 4, on passe de 70 à 64 recompositions (gain de plus de 8%), tandis que la valeur de *invret* est passée de 9,84 à 10,15 (perte de 3%).

On observe en Figure 10.2 le gain sur le critère *recomp* et la perte sur le critère *invret* pour chaque instance lorsqu’on choisit de passer *recomp* en critère prioritaire.

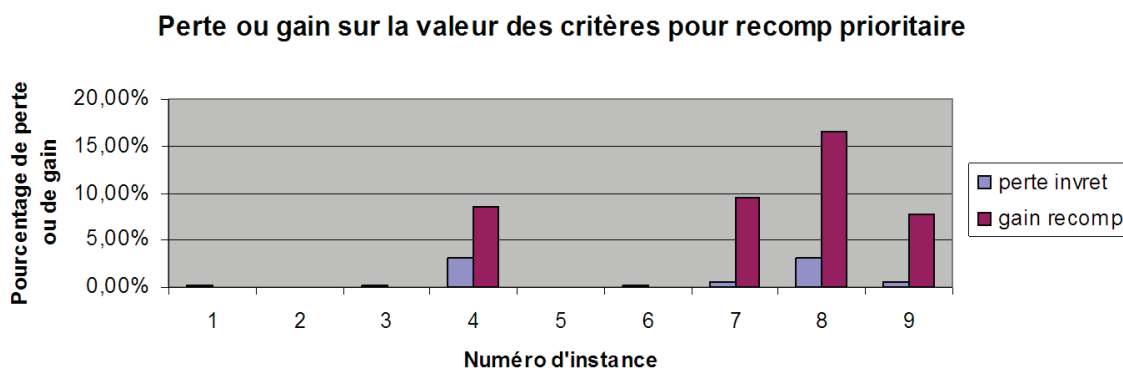


FIGURE 10.2 – Perte ou gain sur la valeur des critères lorsque *recomp* est prioritaire.

D’après le graphique de la Figure 10.2, la perte sur la valeur de *invret* est négligeable par rapport au gain sur le nombre de recompositions.

10.4 Estimation des critères d’évaluation

À la section 4.2, nous avons défini deux catégories de critères : les critères de construction, optimisés dans les approches présentées à la Partie II, et les critères d’évaluation, estimés a posteriori

(par simulation ou en post-opérationnel). Nous cherchons donc à vérifier la pertinence des critères optimisés (et donc des approches proposées) en estimant les critères d'évaluation des solutions obtenues. Dans ce but, on utilise l'outil de simulation décrit au Chapitre 9 : chaque solution est testée sur 100 scénarios de retards. Sauf mention contraire, les valeurs des indicateurs présentées correspondent alors à une moyenne des valeurs sur les 100 scénarios.

10.4.1 Corrélation entre *invret* et la charge de travail

Le critère lié à la charge de travail (durée de circulation des engins, cf. section 4.2), consiste en l'homogénéisation des charges des engins. Or on a supposé que l'homogénéisation des temps de retournement dans le roulement (*invret*) permettait également celle des charges de travail. On cherche ici à observer l'impact de l'optimisation de *invret* sur la charge de travail par engin.

Dans ce cadre, nous avons étudié la charge de travail par engin dans les solutions obtenues par PRESTO et (MPT) pour les instances 1 et 2. La Figure 10.3 permet de visualiser la charge de travail par engin en minutes des solutions calculées par PRESTO et par (MPT) pour les instances cibles 1 et 2. La charge de travail est liée à la structure d'une solution : la charge de travail par engin reste la même pour tout scénario. On simule ici les solutions sur un unique scénario sans retards.

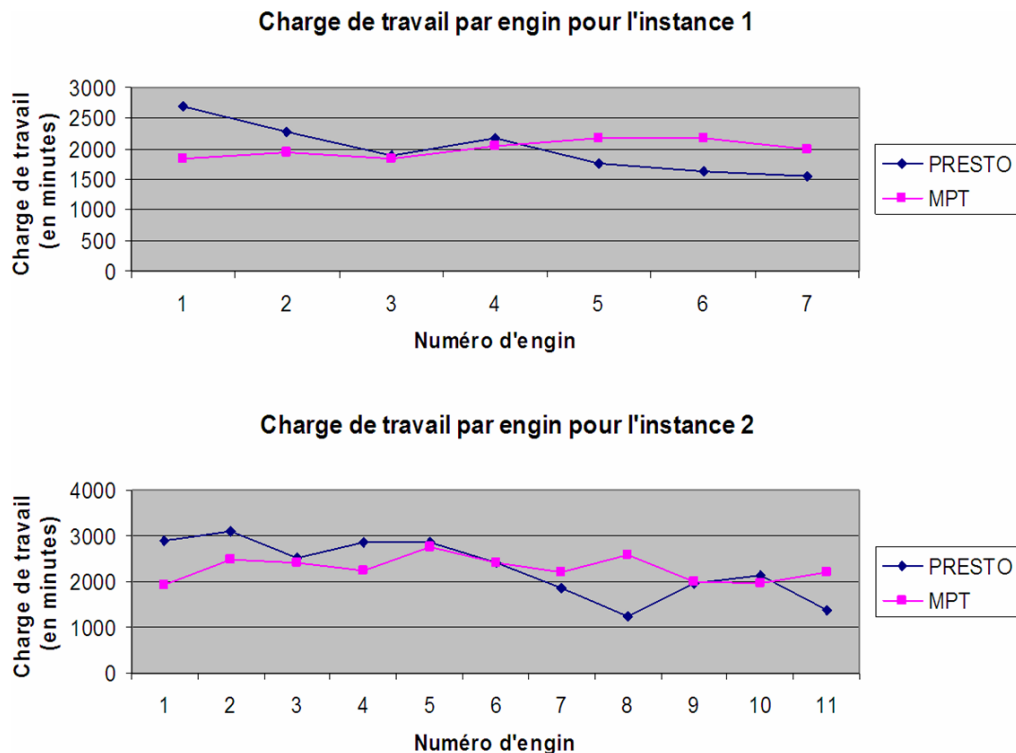


FIGURE 10.3 – Charge de travail par engin des instances 1 et 2 pour les deux méthodes utilisées.

Sur chaque graphique de la Figure 10.3, on constate un lissage de la charge pour la solution calculée par (MPT) par rapport à celle de PRESTO. Par exemple pour l'instance 1 traitée par PRESTO, l'engin 1 a une charge de plus de 2500 minutes sur la semaine, tandis que celle de l'engin 7 dépasse à peine 1500 minutes. En revanche, la charge de travail de chaque engin de la solution par (MPT) est comprise entre 1800 et 2200 minutes.

Afin d'analyser plus finement la variabilité de la charge de travail par engin selon la méthode utilisée, on calcule la moyenne, l'écart-type, le minimum et le maximum des charges de travail des engins (en minutes) pour les instances 1 et 2 (cf. Table 10.5). De plus, on représente cette variabilité par des boîtes à moustaches, qui utilisent la moyenne, la médiane, les quartiles 1 et 3, le minimum et le maximum des charges de travail des engins (en minutes) (cf. Figure 10.4).

Instance	Instance 1 (7 engins)		Instance 2 (11 engins)	
	PRESTO	(MPT)	PRESTO	(MPT)
Moyenne	1997	1997	2295	2293 ²
Écart-type	403	140	412	264
Minimum	1559	1829	1241	1922
Maximum	2681	2164	3087	2759

TABLE 10.5 – Analyse de la charge de travail par engin des solutions obtenues par PRESTO et par (MPT) pour deux instances cibles.

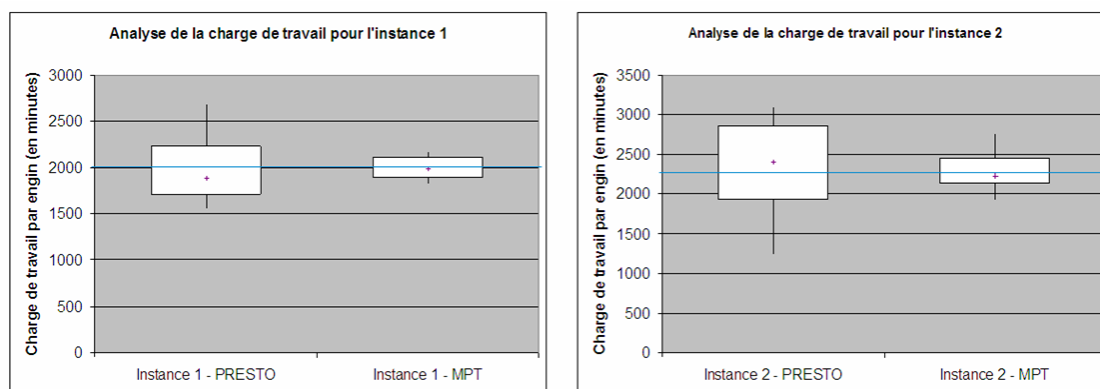


FIGURE 10.4 – Variabilité de la charge de travail par engin sur les instances 1 et 2 selon la méthode utilisée.

Les chiffres donnés dans la Table 10.5 et les boîtes à moustaches présentées en Figure 10.4 donnent lieu à un certain nombre d'observations :

- l'écart-type est beaucoup plus faible pour les solutions obtenues par (MPT) (il passe de 403 minutes pour la solution obtenue par PRESTO à 140 pour l'instance 1) ;
- la médiane pour la solution obtenue par (MPT) est plus proche de la moyenne que celle de PRESTO, (elles sont même quasiment confondues : 1988 et 1997 minutes pour l'instance 1) ;
- la boîte à moustaches est beaucoup plus étendue pour la solution obtenue par PRESTO (l'amplitude est de 335 minutes alors qu'elle est de 1122 pour PRESTO pour l'instance 1).

Les mêmes observations peuvent être faites concernant l'instance 2. Elles conduisent toutes aux mêmes conclusions : la dispersion de la charge de travail est beaucoup plus faible pour des solutions obtenues par l'approche (MPT). L'optimisation de *invret* conduit donc bien à l'amélioration du critère lié à la charge de travail.

Corrélation entre *invret* et les autres critères d'évaluation

Un critère de robustesse important est la régularité (cf. section 4.2). Elle est mesurée et suivie en permanence en opérationnel et en post-opérationnel, et est utilisée comme un indicateur clé de la qualité du service fourni.

Afin d'estimer la régularité³ pour les solutions obtenues par PRESTO et par une approche intégrant la robustesse, nous avons effectué des simulations sur les instances de référence. Les solutions sont éprouvées sur 100 scénarios (identiques pour toutes les solutions). La Table 10.6 permet de comparer la régularité moyenne sur les 100 scénarios pour les solutions obtenues par PRESTO et par (HGL). Le gain relatif par rapport à PRESTO est également indiqué.

Instance	Solution obtenue par PRESTO	Solution obtenue par (HGL)	Gain par rapport à PRESTO
Instance 1	75,9%	76,6%	0,8%
Instance 2	75,8%	77,4%	2,0%
Instance 3	66,8%	68,2%	2,2%
Instance 4	56,1%	70,3%	25,4%
Instance 5	54,5%	53,4%	-2,1%
Instance 6	68,3%	69,0%	1,0%
Instance 7	68,2%	71,2%	4,4%
Instance 8	42,9%	46,0%	7,3%
Instance 9	65,0%	63,6%	2,2%
Moyenne	63,6%	66,3%	4,8%

TABLE 10.6 – Régularité moyenne sur 100 scénarios pour les solutions obtenues par PRESTO et (HGL).

On constate une tendance à gagner en régularité par l'optimisation de *invret* puisque sur huit instances parmi les neuf de référence, on observe un gain moyen de 5,6% en régularité (4,8% en incluant l'instance 5).

D'autres critères d'évaluation avaient été également définis, comme le retard propagé cumulé et le temps de retour à la normale. Les valeurs du gain moyen par rapport à PRESTO sur les instances cibles sont données dans la Table 10.7. On observe alors une tendance plus nette à améliorer ces critères par l'approche (HGL).

Critère / Instance	Retard propagé cumulé	Temps de retour à la normale
Instance 1	18,6%	6,2%
Instance 2	36,1%	12,9%
Instance 3	13,3%	7,1%

TABLE 10.7 – Gain moyen sur les critères *retprop* et *TRN* par rapport à PRESTO pour les instances cibles.

Conclusion intermédiaire

On constate dans cette section que l'optimisation du critère *invret*, intégré dans les trois méthodes (HGL), (CGC) et (MPT), tend à améliorer les critères d'évaluation (définis à la section 4.2) suivants : charge de travail, régularité, retard propagé, temps de retour à la normale. Le choix de ce critère est donc pertinent pour la construction de roulements robustes.

10.5 Analyse de la maintenance

Deux méthodes ont été utilisées pour intégrer la maintenance dans le roulement (cf. section 5.3) :

3. La régularité mesurée ici est le taux de trains arrivant avec moins de 2 minutes de retard.

- PRESTO, (HGL) et (CGC) ajoutent la maintenance lors d'une Phase 4, en fin de processus, alors que le roulement a déjà été construit ;
- (MPT) inclut les contraintes de maintenance lors de la construction du roulement.

Nous comparons ici les deux méthodes. Parmi les neuf instances de référence présentées dans la Table 10.1, six permettent le respect (exact ou approché) des contraintes de maintenance : les instances 1, 2, 4, 5, 6 et 8⁴. Pour chacune de ces six instances, nous avons comparé la maintenance dans les roulements obtenus par PRESTO et par (MPT).

La méthode (MPT) a été paramétrée par défaut, avec $\overline{pm} = 0,5.pm$. Pour les instances de référence 1 et 6, on cherche à passer en maintenance tous les jours ($pm = 1$). L'intervalle minimum entre deux créneaux de maintenance est alors de 12 heures ($\overline{pm} = 0,5$).

On étudie d'abord la maintenance pour l'instance 1. La Figure 10.5 correspond aux roulements des 7 engins de l'instance 1 sur les trois premiers jours (du dimanche au mardi) obtenus par les méthodes PRESTO et (MPT). Les créneaux de maintenance y ont été mis en valeur.

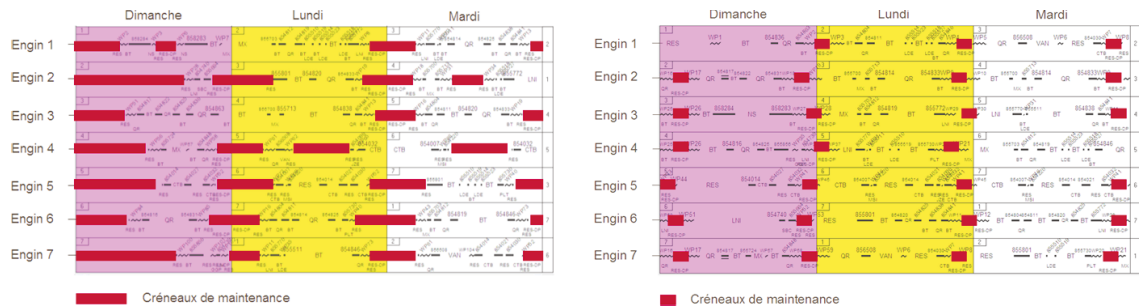


FIGURE 10.5 – Extraits des roulements d'engins de l'instance 1 obtenus par PRESTO (à gauche) et par (MPT) (à droite)

La Figure 10.6 correspond aux mêmes résultats pour l'instance 6 : roulements sur les trois premiers jours obtenus par les méthodes PRESTO et (MPT).

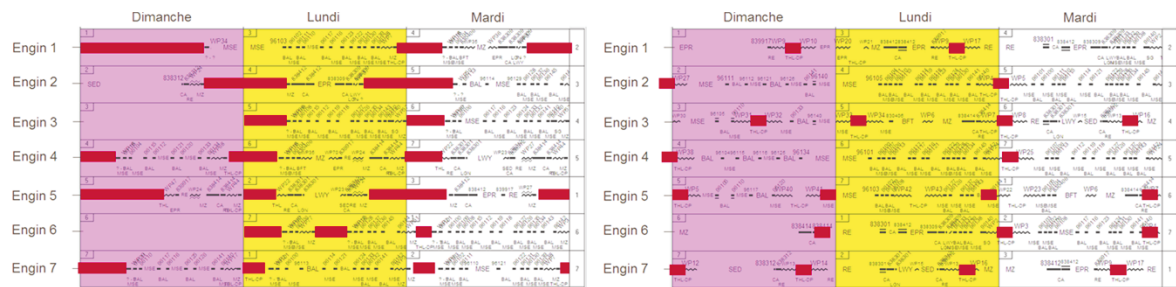


FIGURE 10.6 – Extraits des roulements d'engins de l'instance 6 obtenus par PRESTO (à gauche) et par (MPT) (à droite).

D'après les roulements présentés, les deux méthodes permettent des passages en maintenance réguliers, bien qu'on observe des créneaux parfois proches dans les roulements obtenus par PRESTO.

4. On rappelle que pour les autres instances, les données (par exemple les évolutions possibles) ou la solution de flots (par exemple le nombre d'engins) ne permettent pas le respect des contraintes de maintenance.

Pour les deux méthodes, on constate que les engins passent souvent en maintenance la nuit et tous en même temps. Cela pourrait poser un problème de capacité au niveau du technicentre. Actuellement, aucune donnée de capacité n'est fournie. Si nécessaire, on pourrait simplement introduire une contrainte de capacité.

Pour la méthode (MPT), on pourrait envisager l'intégration de contraintes supplémentaires au modèle 8.2.7, qui pourraient avoir la forme suivante :

$$\sum_{i, m_i \in ph} .z_{ki} < cap(depot), \forall ph$$

avec ph un ensemble de créneaux occupant la même plage horaire et $cap(depot)$ le nombre d'engins maximum pouvant être accueillis simultanément dans le technicentre.

Les figures mettent en évidence une différence entre les méthodes utilisées par PRESTO et (MPT) pour intégrer la maintenance : les durées des créneaux proposés. Dans les solutions obtenues par PRESTO, les créneaux sont de durées très variables et souvent assez longues, laissant ainsi le choix des horaires des opérations de maintenance au technicentre. Au contraire, (MPT) propose des créneaux correspondant à la durée minimale d'immobilisation en maintenance (dm).

Actuellement, nous pouvons constater cette différence, sans savoir ce qui est préférable d'un point de vue métier. Cependant, si les créneaux proposés par (MPT) s'avéraient trop justes, il serait facile d'adapter la méthode en augmentant la durée d'immobilisation dans les données.

Par ailleurs, nous avons noté un inconvénient qui n'apparaît pas directement sur les figures présentées : les évolutions créées (sillons à vide dédiés au technicentre) peuvent être très longues. En effet, certaines évolutions ont une distance largement supérieure à 200km.

Or dans la méthode (MPT), la pénalisation des évolutions en fonction de leur distance a été testée : elle peut être mise en place mais la résolution du PLNE devient très lourde. On peut alors imaginer l'interdiction d'évolutions dépassant un seuil donné (100 ou 150km par exemple).

Enfin, nous rappelons que s'il n'est pas possible de respecter les contraintes de maintenance, la méthode (MPT) ne trouve pas de solution admissible et il faut la relancer avec un paramètre (\overline{pm}) relâché ou sans maintenance, tandis que PRESTO fournit toujours une solution "au mieux".

Nous pouvons conclure, malgré quelques différences, que les deux méthodes utilisées pour intégrer la maintenance dans le roulement permettent d'obtenir des solutions de qualité équivalente en termes de maintenance. Nous nous intéressons maintenant à leur impact sur la robustesse des solutions.

Nous présentons alors la valeur de *invret* des solutions sans et avec maintenance obtenues par PRESTO, par (HGL) et par (MPT) sur les instances cibles 1 et 2 (cf. Table 10.8). Le gain relatif (par rapport à PRESTO) et le temps de calcul sont également renseignés.

Instance	Instance 1			Instance 2		
	PRESTO	(HGL)	(MPT)	PRESTO	(HGL)	(MPT)
Valeur de <i>invret</i> sans maintenance	9,18	7,94	4,09	10,52	7,27	5,80
Gain sur <i>invret</i> sans maintenance	-	13,7%	55,4%	-	30,9%	44,4%
Valeur de <i>invret</i> avec maintenance	15,50	15,09	3,96	21,62	18,83	5,80
Gain sur <i>invret</i> avec maintenance	-	1,6%	74,5%	-	9,5%	73,2%
Temps d'exécution sans maintenance	5'	10'	6'	< 1'	5'	6'
Temps d'exécution avec maintenance	5'	10'	7'	< 1'	5'	31'

TABLE 10.8 – Impact de l’ajout de la maintenance par des méthodes différentes sur la valeur de *invret* pour les instances cibles 1 et 2

Comme démontré à la section 6.4.1, on constate que la valeur de *invret* est dégradée lorsqu’on ajoute la maintenance en fin de processus (ici dans les méthodes PRESTO et (HGL)). Ce n’est pas le cas lorsqu’on intègre robustesse et maintenance simultanément lors de la construction du roulement⁵. Une approche globale (gestion simultanée de la maintenance et des critères) comme l’approche (MPT) permet donc d’obtenir des meilleures valeurs pour *invret*. En revanche, le temps de calcul peut être impacté : dans le cas de l’instance 2, l’ajout de la maintenance fait passer le temps de résolution de 6 à 31 minutes.

10.6 Conclusion

À travers les expérimentations effectuées dans ce chapitre, nous avons pu évaluer les différentes méthodes proposées à la Partie II (HGL), (CGC) et (MPT) selon plusieurs indicateurs : critères de robustesse (de construction et d’évaluation), maintenance, temps d’exécution.

Les trois méthodes présentent toutes un gain significatif par rapport à la méthode existante PRESTO sur le critère *invret*. On a de plus constaté une corrélation entre ce critère et les critères d’évaluation : l’optimisation de *invret* permet l’amélioration de la répartition de la charge de travail, du retard propagé, de la régularité et du temps de retour à la normale. Ces observations valident donc la pertinence de l’approche structurelle utilisée et du choix du critère de robustesse *invret* dans le cadre de cette approche.

Cependant, les trois méthodes proposées ne sont pas équivalentes.

La méthode (HGL) est rapide puisqu’en quelques minutes on obtient une solution intéressante en termes de robustesse (selon les critères de construction *invret* et *recomp*, et les critères d’évaluation). Cependant, l’ajout de la maintenance en fin de processus dégrade la robustesse des solutions. De plus, le problème résolu est très contraint par la solution de PRESTO (flots, sillons à vide). Cette méthode présente donc les qualités nécessaires à une industrialisation. De plus, elle a été développée directement comme un module du logiciel PRESTO déjà industrialisé. Les équipes en

⁵. L’amélioration observée pour l’instance 1 entre les cas sans et avec maintenance peut surprendre. Elle s’explique par l’optimisation des coûts de production (distance des sillons à vide) prioritaire par rapport à *invret*.

charge de l'industrialisation ont donc déjà pris le code en main.

La méthode (CGC) répond au même problème que la méthode (HGL), mais obtient systématiquement un gain moins intéressant sur *invret* pour les instances testées. L'intégration d'autres critères (par exemple *recomp*) ou de la maintenance compliquerait le modèle déjà difficile à résoudre en l'état. Enfin, le temps d'exécution dépend des instances testées et peut être long.

La méthode (MPT) permet de résoudre exactement un problème plus large puisque les flots et les sillons à vide sont redistribués. Elle permet alors un gain beaucoup plus intéressant sur le critère *invret*, tout en pénalisant les temps de retournement intermédiaires. La maintenance est intégrée simultanément, conservant ainsi l'optimalité des critères de robustesse et des coûts de production pris en compte. Le temps de calcul dépend des instances testées mais reste raisonnable. D'autres critères (par exemple *recomp* ou les kilomètres parcourus en évolutions) sont modélisés, mais le temps de calcul peut alors devenir rédhibitoire. La méthode est donc intéressante mais encore trop lourde pour une éventuelle industrialisation.

L'utilisation d'une approche structurale consistant en l'optimisation de critères agissant sur la structure de la solution et les méthodes de résolution utilisées dans les approches (HGL) et (MPT) semblent donc adaptées à la résolution du problème de planification de roulements d'engins robustes.

Conclusion

Nous nous sommes d'abord intéressés au problème de planification des roulements d'engins en milieu ferroviaire. À SNCF, il existe actuellement un logiciel, PRESTO, permettant de répondre à ce problème en proposant des solutions à coûts de production optimaux. Cependant, on observe en opérationnel des retards dus à l'apparition d'aléas tels que des pannes matérielles. L'impact de ces retards sur le plan de transport peut être important, dégradant les coûts de production et la qualité de service proposée. Face à ce constat, nous avons adopté une démarche robuste.

En premier lieu, nous avons proposé une définition de la robustesse en milieu ferroviaire adaptée au transport express régional français (TER) :

La robustesse d'un plan de transport est son aptitude à résister et à s'autoréguler en un temps limité ou à être facilement réparable en présence d'une perturbation.

L'objectif est alors de construire des roulements d'engins robustes au sens de cette définition.

Une des premières contraintes auxquelles nous avons dû faire face concerne les coûts de production : il ne s'agit pas d'utiliser des engins supplémentaires, dont les coûts d'utilisation sont très élevés, pour obtenir une solution robuste. On cherche à rester à coûts de production constants ou très peu dégradés. Dans ce but et après une étude du problème et de l'existant au Chapitre 1, nous avons décidé de réutiliser une partie de PRESTO. En effet, la solution calculée lors de la première phase de PRESTO garantit des coûts de production optimaux.

De plus, nous avons prouvé une condition nécessaire et suffisante à l'existence d'un roulement droit, c'est-à-dire d'un seul roulement cyclique, dans la solution calculée par PRESTO. Nous avons également montré comment déterminer un nombre minimal de roulements cycliques lorsque cette condition n'était pas vérifiée. En effet, dans le cadre de notre démarche robuste nous cherchons à obtenir un nombre minimal de roulements cycliques.

Suite à une analyse de la littérature, les approches classiques de modélisation directe des aléas par scénarios ou par intervalles d'incertitude nous ont semblé peu adaptées à la construction de roulements d'engins robustes dans notre contexte. Nous avons dressé un panorama des approches robustes envisageables et nous avons opté pour une approche structurelle : on agit directement sur la structure d'une solution pour la rendre robuste d'après des critères adaptés au contexte.

Pour notre périmètre, nous définissons des critères de robustesse en deux catégories : les critères de construction sont pris en compte directement lors de la construction de la solution, et les critères d'évaluation sont utilisés a posteriori pour mesurer la qualité de la solution produite en termes de robustesse.

Les critères de construction concernent essentiellement les temps de retournement, un temps de retournement étant l'intervalle de temps passé en gare par un engin entre la date d'arrivée d'une tâche et la date de départ de la suivante. On cherche à maximiser le temps de retournement

minimum, à homogénéiser l'ensemble des temps de retournement dans le roulement, et à pénaliser des temps de retournement intermédiaires à la fois trop longs pour un stationnement en gare et trop courts pour un remisage. Nous avons également pris en compte le nombre de recompositions, c'est-à-dire le nombre d'opérations de couplage et de découplage de plusieurs engins couvrant une même tâche simultanément (UM). Ces recompositions prennent du temps et augmentent le risque de retard, on cherche donc à les limiter.

Les critères d'évaluation sont calculés a posteriori, une fois les roulements construits. On étudie la répartition de la charge de travail entre les engins : la charge de travail d'un engin est le temps passé à effectuer des tâches sur un horizon temporel. Un roulement est plus robuste si tous les engins effectuent un volume horaire équivalent. Plusieurs critères d'évaluation sont liés au retard : on cherche des roulements maximisant la régularité, c'est-à-dire le taux de tâches à l'heure, et minimisant le retard propagé cumulé et le retard en fin d'horizon temporel. De plus, une solution est plus robuste si elle minimise le temps de retour à la normale, c'est-à-dire le temps en situation perturbée.

Afin de répondre à des besoins différents, nous avons mis en œuvre trois approches intégrant la robustesse par l'optimisation des critères de construction ainsi définis. Elles garantissent toutes le respect des coûts de production critiques en se basant sur la Phase 1 de PRESTO : le nombre d'engins et le nombre de kilomètres en actif sont optimaux. Enfin, elles s'appuient sur le résultat du Chapitre 2 afin de produire un nombre de sous-roulements minimum. On obtient ainsi plusieurs sous-problèmes résolus indépendamment, chacun concernant un unique type d'engin.

Tous les développements pour la mise en œuvre des méthodes ont été complètement intégrés à PRESTO. Les approches ont ainsi été testées sur 9 instances réelles de référence correspondant à un échantillon représentatif des données à notre disposition (de 6 à 15 engins pour couvrir 168 à 778 tâches dans trois régions TER différentes).

Les approches présentent également des différences : articulation avec PRESTO, données, prise en compte de la robustesse et de la maintenance.

Nous avons proposé de remplacer la Phase 2 de PRESTO, qui construit arbitrairement les lignes de roulements, par une approche heuristique visant à construire des lignes robustes. Dans cette approche (HGL), la solution de la première phase de PRESTO est respectée : le nombre d'engins, les sillons à vide, les flots (nombre d'engins circulant sur chaque tâche). On garantit ainsi des coûts de production optimaux : les coûts d'utilisation des engins les coûts de déplacement à vide, en actif et en passif sont ceux calculés par PRESTO.

La robustesse est intégrée par la prise en compte des critères de construction cités précédemment (exceptés les temps de retournement intermédiaires) et l'utilisateur peut définir une hiérarchie sur ces critères. La méthode de résolution employée consiste en deux étapes : une heuristique construit une solution initiale de manière gloutonne en optimisant les critères dans l'ordre lexicographique, puis une technique de recherche locale par voisinage permet d'améliorer la solution obtenue.

Lors des tests sur des instances réelles, nous avons pu comparer les solutions obtenues par cette heuristique à l'existant. Les solutions obtenues présentent un gain en robustesse intéressant rapidement : le gain moyen sur le critère principal est de 10% pour une phase robuste limitée à 5 minutes.

Cependant, la maintenance est ajoutée après calcul d'un roulement robuste à coûts de production optimaux, lors de la Phase 4 de PRESTO. La solution est modifiée, dégradant les critères de robustesse et les coûts de production optimisés. La solution obtenue par l'approche (HGL) reste

malgré tout plus robuste que celle obtenue par PRESTO.

L'approche (CGC) s'appuie sur une modélisation par la programmation linéaire en nombres entiers correspondant à une vision par chemins. Chaque chemin de la source au puits dans le graphe spatio-temporel correspond à une ligne de roulement potentielle. Il s'agit alors d'affecter un chemin robuste (d'après le critère optimisé, ici l'homogénéisation des temps de retournement) à chaque ligne de roulement de façon à respecter les flots calculés par PRESTO. De même que pour l'approche (HGL), le respect de ces flots et des sillons à vide utilisés par PRESTO garantit des coûts de production optimaux. L'approche (CGC) permet de remplacer les Phases 2 et 3 de construction et d'enchaînement des lignes de PRESTO.

Le modèle comportant un nombre exponentiel de variables, nous avons cherché à résoudre le problème de façon heuristique en utilisant une technique de génération de colonnes. Le sous-problème associé correspond à une recherche de chemin optimal particulière : la prise en compte des poids sur les arcs dépend du reste du chemin. Un algorithme basé sur des listes d'étiquettes en chaque sommet permet de résoudre ce problème.

Une méthode par arrondi permet ensuite de trouver une solution entière. On constate que le coût de la solution entière est souvent très proche du coût de la solution de la relaxation continue obtenue par génération de colonnes. En revanche, la solution continue est très fractionnaire et ne permet pas de trouver rapidement une solution entière.

Les résultats sur les instances cibles présentés à la section 7.4 montrent un gain variable sur le critère optimisé par rapport aux solutions de PRESTO (entre 0 et 30%). Le temps d'exécution dépend lui aussi de l'instance (de 6 à 80 minutes). Des tests complémentaires sur les instances de référence ont permis d'observer un gain moyen inférieur à 10% pour des temps d'exécution dépassant parfois les 3 jours.

De plus, les contraintes sur l'enchaînement des lignes ont été relâchées afin de garantir l'existence d'une solution admissible. L'enchaînement des lignes se fait donc si besoin par la Phase 3 de PRESTO. La maintenance est ensuite ajoutée par la Phase 4 de PRESTO, dégradant la solution en termes de coûts et de robustesse. On ne peut donc pas garantir un nombre minimum de sous-roulements en sortie.

Le modèle est déjà difficile à résoudre pour un critère et donne des résultats moins intéressants que l'approche (HGL). Nous n'avons donc pas cherché à intégrer les critères de robustesse manquants et la maintenance.

L'approche (MPT) est plus globale que les approches (HGL) et (CGC). D'une part, elle permet de redistribuer les sillons à vide et les flots afin de gagner en robustesse. D'autre part, elle intègre simultanément les critères de robustesse de construction (exceptée la minimisation du temps de retournement minimum) et la maintenance lors de la construction et de l'enchaînement des lignes. On garantit ainsi le respect des contraintes de maintenance lorsque c'est possible, et les critères optimisés ne sont pas dégradés par une phase ultérieure.

Cette approche est fondée sur la modélisation globale du problème par un programme linéaire en nombres entiers dont les variables principales représentent l'enchaînement de deux tâches sur une ligne de roulement. La résolution du problème s'appuie alors sur l'utilisation d'un solveur de programmes mathématiques comme IBM ILOG CPLEX.

Les résultats sur toutes les instances testées présentent un gain plus important en robustesse en comparaison avec les approches (HGL) et (CGC) : sur le critère d'homogénéisation des temps de retournement, on observe un gain moyen de plus de 36% relativement aux solutions de PRESTO.

Un gain équivalent est constaté lorsque les contraintes de maintenance sont vérifiées. Le temps de résolution, pouvant s'élever à 263 minutes, dépend fortement des instances.

Enfin, il est possible que les contraintes de maintenance ne puissent pas être respectées si le nombre d'engins calculé par PRESTO est insuffisant ou si les données sont trop restrictives. Dans ce cas, il est possible de paramétrer l'approche pour relâcher partiellement ou complètement les contraintes de maintenance.

Parallèlement à ces travaux, un outil de simulation a été mis en place afin de pouvoir mesurer la robustesse des solutions obtenues. L'outil permet de simuler le comportement d'un roulement d'engins en situation perturbée. Une situation perturbée est modélisée par un scénario affectant un retard en minutes à chaque tâche. Les scénarios sont construits d'après des données historiques. Chaque solution peut être testée sur un nombre de scénarios suffisant pour obtenir des résultats significatifs. Tous les critères de robustesse de construction et d'évaluation sont calculés et les valeurs moyennes sont fournies en sortie. On peut ainsi comparer les trois approches entre elles et avec l'existant.

Les résultats obtenus confirment les premières conclusions concernant les différentes approches. En particulier, l'approche (HGL) donne des solutions intéressantes rapidement et l'approche (MPT) fournit les solutions les plus robustes d'après les critères calculés.

Les résultats montrent également une corrélation positive entre le critère d'homogénéisation des temps de retournement et les critères d'évaluation liés au retard et à la charge de travail : en effet, l'homogénéisation des temps de retournement permet l'amélioration des critères d'évaluation, validant le choix de ce critère pour la construction de roulements d'engins robustes.

Par ailleurs, étant donné l'avancement des travaux et en particulier des prototypes développés pour tester les trois approches, une phase d'industrialisation a débuté fin 2013.

L'approche (CGC) est moins intéressante que les approches (HGL) et (MPT) en termes de robustesse : seul le critère d'homogénéisation des temps de retournement est optimisé et le gain sur ce critère est plus faible. De plus, le temps de calcul est variable et peut être très long.

L'approche (MPT) présente un réel intérêt industriel : le gain sur le critère d'homogénéisation des temps de retournement est plus élevé que pour les solutions obtenues par les autres approches, et l'approche globale permet le respect des contraintes de maintenance sans dégrader les critères de robustesse. En revanche, le temps de résolution peut être encore trop long pour permettre une industrialisation en l'état.

L'approche (HGL) est rapide et intéressante en termes de robustesse puisqu'elle intègre plusieurs critères et présente un gain significatif sur les critères choisis. De plus, comme les autres méthodes, elle est complètement intégrée à PRESTO. Ces qualités ont permis de valider son industrialisation : l'équipe en charge de PRESTO a pris en main le code et a pu tester la méthode sur de nouveaux cas réels en région Nord-Pas-de-Calais.

Les premiers retours des experts métier confirment l'intérêt de l'approche : les critères optimisés sont pertinents et les solutions observées sont plus robustes que celles obtenues par PRESTO. De plus, les experts ont insisté sur la criticité des contraintes de maintenance non prises en compte dans les tests présentés. L'intégration de la maintenance lors de la construction des lignes comme cela est fait dans l'approche (MPT) serait donc un atout en cas d'utilisation industrielle.

On note que l'approche (MPT) présente des atouts majeurs et ne nécessite plus que l'amélior-

ration du temps de résolution pour envisager une utilisation industrielle. Dans ce but, deux pistes sont envisageables : on peut tenter d'améliorer directement le temps de calcul en affinant encore le modèle ou en agissant sur sa résolution, ou on peut chercher une solution admissible non optimale présentant un gain significatif en termes de robustesse.

Annexe A

Résultats

A.1 Fiabilité des résultats obtenus par simulation en fonction du nombre de scénarios testés

Dans le cadre de l'analyse de fiabilité des résultats par simulation, nous avons effectué une simulation sur 5000 scénarios de l'instance cible 1 (cf. Table 5.6) et étudié les valeurs de la régularité et du retard propagé cumulé.

La Figure A.1 recense les valeurs moyennes de la régularité pour différents sous-ensembles de 10 à 500 scénarios. Il indique également la moyenne augmentée ou diminuée de l'écart-type. Les valeurs pour l'ensemble des 5000 scénarios sont également fournies à titre de référence.

La Figure A.2 donne des informations équivalentes concernant le retard propagé cumulé.

Nombre de scénarios	Moyenne	Moyenne + écart-type	Moyenne - écart-type	Nombre de scénarios	Moyenne	Moyenne + écart-type	Moyenne - écart-type
10	81,2%	73,0%	89,5%	100	81,1%	77,4%	84,8%
10	80,8%	78,3%	83,4%	100	81,1%	78,4%	83,8%
10	80,5%	76,3%	84,7%	100	81,2%	78,1%	84,3%
10	80,2%	76,7%	83,8%	100	81,4%	78,3%	84,6%
10	81,5%	79,5%	83,5%	100	80,8%	77,8%	83,8%
10	80,2%	77,2%	83,1%	100	81,3%	78,1%	84,6%
10	79,5%	76,1%	82,8%	100	80,4%	77,7%	83,2%
10	81,4%	77,9%	84,9%	100	80,9%	77,9%	83,9%
10	79,8%	77,5%	82,0%	100	81,2%	78,0%	84,4%
10	81,5%	78,7%	84,2%	100	81,3%	78,0%	84,5%
20	81,5%	77,7%	85,2%	200	81,2%	78,0%	84,4%
20	81,8%	79,3%	84,3%	200	81,0%	77,9%	84,1%
20	81,6%	78,1%	85,1%	200	80,9%	77,9%	84,0%
20	81,9%	78,9%	84,9%	200	81,4%	78,3%	84,5%
20	80,1%	76,3%	84,0%	200	81,1%	77,8%	84,3%
20	80,8%	78,0%	83,5%	200	81,0%	77,8%	84,2%
20	80,7%	77,9%	83,5%	200	81,5%	78,2%	84,8%
20	80,9%	77,8%	84,0%	200	80,6%	77,5%	83,7%
20	81,9%	78,8%	85,1%	200	81,2%	78,0%	84,4%
20	81,3%	78,4%	84,2%	200	81,3%	78,1%	84,5%
50	81,1%	77,5%	84,6%	500	81,2%	77,8%	84,6%
50	81,0%	78,3%	83,6%	500	81,2%	78,2%	84,2%
50	81,2%	78,5%	83,9%	500	81,0%	78,0%	84,0%
50	81,3%	78,5%	84,2%	500	81,4%	78,3%	84,4%
50	80,6%	77,8%	83,5%	500	80,9%	77,7%	84,0%
50	81,3%	77,9%	84,7%	500	81,1%	77,8%	84,4%
50	80,8%	78,3%	83,3%	500	81,1%	78,0%	84,2%
50	80,5%	77,9%	83,2%	500	80,8%	77,7%	83,9%
50	81,2%	77,8%	84,5%	500	81,1%	77,9%	84,3%
50	81,4%	77,7%	85,1%	500	81,1%	78,0%	84,3%
				5000	81,1%	77,9%	84,2%

FIGURE A.1 – Valeur moyenne de la régularité pour des sous-ensembles de 10 à 500 scénarios et pour l'ensemble global des 5000 scénarios.

Nombre de scénarios	Moyenne	Moyenne + écart-type	Moyenne - écart-type
10	293	154	431
10	273	157	389
10	242	156	328
10	232	122	341
10	326	217	434
10	300	226	374
10	310	182	438
10	301	180	422
10	272	172	373
10	273	190	355
20	260	152	367
20	262	177	347
20	266	186	345
20	257	159	354
20	263	156	369
20	275	196	353
20	275	186	365
20	238	145	330
20	253	155	351
20	288	187	388
50	287	168	405
50	277	182	371
50	263	165	360
50	263	181	345
50	275	190	361
50	244	130	357
50	270	181	360
50	288	182	395
50	263	164	363
50	273	171	375

Nombre de scénarios	Moyenne	Moyenne + écart-type	Moyenne - écart-type
100	272	168	376
100	265	175	355
100	279	185	373
100	256	163	350
100	255	161	349
100	275	174	376
100	265	155	375
100	285	178	392
100	266	176	356
100	267	166	369
200	271	168	375
200	267	166	368
200	266	173	359
200	270	172	368
200	271	174	368
200	261	162	359
200	269	175	363
200	286	183	389
200	272	178	365
200	274	178	369
500	270	167	374
500	266	169	363
500	268	177	358
500	266	169	363
500	273	178	367
500	270	165	374
500	270	174	365
500	284	180	388
500	273	176	370
500	270	174	365
5000	271	173	369

FIGURE A.2 – Valeur moyenne de la régularité pour des sous-ensembles de 10 à 500 scénarios et pour l'ensemble global des 5000 scénarios.

Annexe B

Glossaire

Creux de roulement

Un creux de roulement est un temps de retournement très long (plusieurs heures).

Déplacement actif

Un déplacement actif correspond à un engin circulant sur une tâche afin d'en couvrir la demande (ou une partie de la demande). C'est un déplacement nécessaire.

Déplacement passif

Un déplacement passif est un déplacement non nécessaire à la couverture de la demande : un engin supplémentaire circule sur une tâche dont la demande est déjà couverte.

Dépôt

Le technicentre (ou dépôt) est le site de maintenance auquel un engin doit se rendre.

Durée d'immobilisation

La durée d'immobilisation (en maintenance) est le temps minimum pendant lequel un engin doit rester au dépôt.

Engin

Un engin est un ensemble de rames indivisible.

Grille horaire

Une grille horaire est un graphique représentant l'ensemble des tâches à couvrir sur un horizon temporel (par exemple une semaine), avec pour chaque tâche la demande et le nombre d'engins maximum autorisé.

Ligne

Une ligne est un ensemble de trains qui diffèrent uniquement par leurs horaires de départ et d'arrivée.

Ligne de roulement

Une ligne de roulement est une succession admissible de tâches effectuées sur l'horizon temporel d'étude par un même engin. Sa gare de départ est la gare de départ de sa première tâche, et sa gare d'arrivée est la gare d'arrivée de sa dernière tâche. De même, sa date de début est celle de sa première tâche, et sa date de fin est celle de sa dernière tâche.

Ligne de roulement partielle

Une *ligne de roulement partielle* est une ligne de roulement tronquée, dont les tâches doivent se suivre. On complète une ligne partielle en ne remettant pas en question l'enchaînement des tâches déjà affectées : on peut ajouter des tâches à la suite de la dernière tâche de la ligne partielle uniquement.

Marge de régularité

La marge de régularité est le nombre de minutes de retard que peut prendre une tâche sans être considérée en retard.

Pas de maintenance

Le pas de maintenance (ou pas d'entrée en maintenance) est la fréquence à laquelle un engin doit se rendre en maintenance.

Planification des engins

La planification des engins est la mise en relation d'un parc d'engins avec une charge de travail (tâches à couvrir) sur une période donnée (l'horizon temporel, par exemple une semaine).

Recomposition

Une recomposition est une opération permettant de coupler plusieurs engins entre eux afin d'obtenir une unité multiple ou de les découpler après utilisation.

Roulement d'engins cyclique

Un *roulement de N engins* est un enchaînement de N lignes de roulement sur l'horizon temporel.

Deux lignes ℓ_1 et ℓ_2 peuvent s'enchaîner si la première tâche de ℓ_2 peut suivre la dernière tâche de ℓ_1 , c'est-à-dire si :

- la gare de départ de ℓ_2 est la gare d'arrivée de ℓ_1 ;
- la date de début de ℓ_2 est ultérieure à la date de fin de ℓ_1 .

Un roulement de N engins est *cyclique* si on peut numéroter les lignes de 1 à N , de sorte que :

- pour tout i allant de 1 à $N - 1$, les lignes i et $i + 1$ s'enchaînent ;
- les lignes N et 1 s'enchaînent.

Roulement d'engins partiel

Un *roulement d'engins partiel* est un ensemble de lignes de roulement partielles ou complètes. En plus de ne pas respecter les valeurs des flots, il peut donc aussi violer les contraintes de cyclicité.

Roulement droit

Un roulement droit est un roulement cyclique formé par tous les engins utilisés d'un même type. Cependant, il est possible d'avoir plusieurs roulements cycliques pour un type d'engin (sous-roulements).

Sous-roulement

Un sous-roulement est un roulement formé d'une partie des engins d'un même type uniquement.

Série de matériel

Des engins présentant les mêmes caractéristiques techniques appartiennent à une même série de matériel (ou type d'engin) et sont par conséquent interchangeables.

Sillon à vide

Un sillon à vide (ou W) est une tâche de demande nulle créée afin de déplacer un ou plusieurs engins sans transport de passagers.

Tâche

Une tâche est un parcours direct d'une gare appelée gare de départ vers une autre gare appelée gare d'arrivée, avec un horaire de début et un horaire de fin.

Technicentre

Le technicentre (ou dépôt) est le site de maintenance auquel un engin doit se rendre.

Temps de crochet

Le temps de crochet est le temps minimum de stationnement imposé en gare entre l'arrivée d'un engin et un nouveau départ (en général quelques minutes).

Temps de retournement

Le temps de retournement est le temps de stationnement en gare entre l'arrivée d'un engin et un nouveau départ. Pour deux tâches qui se suivent sur une même ligne, le temps de retournement est l'intervalle entre la date de fin de la première tâche et la date de début de la seconde.

Train

Un train est défini par une origine, une destination, des dessertes, des horaires et un régime, c'est-à-dire les jours de passage dans la semaine.

Un train est composé de trajets unitaires appelés tâches.

Type d'engin

Des engins présentant les mêmes caractéristiques techniques appartiennent à une même série de matériel (ou type d'engin) et sont par conséquent interchangeables.

Unité multiple (UM)

Une unité multiple, notée UM, est un ensemble de plusieurs engins couplés entre eux afin de couvrir une même tâche conjointement.

Unité simple (US)

Une unité simple, notée US, est un engin couvrant une tâche seul.

Bibliographie

- M. A. Aloulou, M. Haouari, F. Zeghal Mansour A model for enhancing robustness of aircraft and passenger connections *Transportation Research* 32C, pages 48-60, 2013.
- A. A. Assad Models for Rail Transportation *Transportation Research* 14B, pages 101-114, 1980.
- A. Ben-Tal et A. Nemirovski Robust convex optimization *Mathematics of Operations Research* 23, pages 769-805, 1998.
- A. Ben-Tal et A. Nemirovski Robust solutions to uncertain programs *Operations Research Letters* 25, pages 1-13, 1999.
- A. Ben-Tal et A. Nemirovski Robust solutions of linear programming problems contaminated with uncertain data *Mathematical Programming* 86, pages 463-473, 2000.
- F. Benhizia *Optimisation du plan de transport par planification intégrée des ressources* Thèse de Doctorat en Génie industriel, Ecole Nationale Supérieure des Mines de Saint-Etienne, SNCF, 2012.
- F. Benhizia, S. Dauzère-Pérès, D. De Almeida, O. Guyon A Lagrangian heuristic for a real-life integrated planning problem of railway transportation resources In *Proceedings of the 5th International Seminar on Railway Operations Modelling and Analysis - IAROR RailCopenhagen*, Copenhagen, Denmark, May 2013.
- D. Bertsimas et M. Sim The Price of Robustness *Operations Research* 52, pages 35-53, 2001.
- D. Bertsimas et M. Sim Robust Discrete Optimization and Network Flows *Mathematical Programming Series B*, pages 49-71, 2002.
- A. Billionnet Using Integer Programming to Solve the Train Platforming Problem *Transportation Science* 37, pages 213-222, 2003.
- J.R. Birge Stochastic programming computation and applications *INFORMS J. on computing*, 9, pages 111-133, 1997.
- M. R. Bussieck, P. Kreuzer, U. T. Zimmermann Optimal Lines for Railway Systems *European Journal of Operational Research* 96, pages 54-63, 1996.
- V. Cacchiani, A. Caprara, P. Toth Solving a real-world train-unit assignment problem Springer and Mathematical Programming Society, 2010.
- A. Caprara, M. Fischetti, and P. Toth Modeling and Solving the Train Timetabling Problem *Operations Research* 50, pages 851-861, 2002.

- A. Caprara, L. Kroon, M. Monaci, M. Peeters, P. Toth Passenger Railway Optimization Technical Report 0035, EU ARRIVAL project, Janvier 2006.
- A. Caprara, P. Toth, D. Vigo, M. Fischetti Modeling and Solving The Crew Rostering Problem *Operations Research* 46, pages 820-830, 1998.
- M. Chandesris Prise en compte de phénomènes stochastiques dans l'optimisation des plans de production. Application au domaine ferroviaire Thèse de Doctorat en sciences mathématiques, Université Paris 6, SNCF, 2005.
- G. B. Dantzig Linear Programming under Uncertainty *Management Science* 1, pages 197-206, 1955.
- M. Fischetti et M. Monaci Light Robustness In R.K. Ahuja et al. (eds.) : *Robust and Online Large-Scale Optimization*, Springer, pages 61-84, 2009.
- J. H. M. Goossens, C. P. M. van Hoesel, L. G. Kroon On Solving Multi-type Railway Line Planning Problems *European Journal of Operational Research* 168, pages 403-424, 2005.
- M. A. Hofman et L. F. Madsen Robustness in train scheduling Master thesis, Informatics and Mathematical Modelling, Technical University of Denmark, Septembre 2005.
- P. Kouvelis et G. Yu Robust discrete optimization and its application Kluwer Academic Publishers, 1997.
- L. Kroon, E. Abbink, M. Vromans, M. Fischetti Reinventing Crew Scheduling at Netherlands Railways *Informatics* Volume 35, Issue 5, 2005.
- L. Kroon, D. Huisman, G. Maróti Optimisation models for railway timetabling, In *Railway Timetable & Traffic*, I.A. Hansen et J. Pachl (eds.), Eurailpress : Hamburg, Germany, pages 135-154, 2008.
- L. G. Kroon et L. W. P. Peeters A Variable Trip Time Model for Cyclic Railway Timetabling *Transportation Science* 37, pages 198-212, 2003.
- A. Le Maitre Optimisation robuste d'une migration de matériel dans un réseau en décroissance de clients Thèse de Doctorat en Informatique et Recherche Opérationnelle, CEDRIC-CNAM, France Telecom, 2008.
- C. Liechen, M. Lübbecke, R. Möhring, S. Stiller The Concept of Recoverable Robustness, Linear Programming Recovery, and Railway Applications In R.K. Ahuja et al. (eds.) : *Robust and Online Large-Scale Optimization*, Springer, 2009.
- C. Liechen, M. Lübbecke, R. Möhring, S. Stiller Recoverable Robustness Technical Report 0066, EU ARRIVAL project, Août 2007.
- N. Marcos. *Modélisation et optimisation de la gestion du matériel roulant à la SNCF*. Thèse de Doctorat en informatique, Université Paris 13, SNCF, 2006.
- L. K. Nielsen, L. Kroon, G. Maroti Absorption Robustness of Railway Resource Schedules Technical Report 0113, EU ARRIVAL project, Decembre 2007.
- E. R. Petersen et R. Merchant Scheduling of trains in a linear Railway system *INFOR* 19, pages 246-258, 1981.

- A. Schrijver. Minimum circulation of railway rolling stock. *CWI-Quarterly* 6, pages 205-217, 1993.
- B. Smith et E. L. Johnson Robust airline fleet assignment : Imposing station purity using station decomposition *Transportation Science* 40, pages 497-516, 2006.
- A. L. Soyster Convex programming with set-inclusive constraints and applications to inexact linear programming *Operations Research* 21, pages 1154-1157, 1973.
- B. Szpigiel Optimal train scheduling on a single track railway In M. Ross (ed.) : *OR'72*, North-Holland, Amsterdam, The Netherlands, pages 343-351, 1973.
- Y. Takeuchi, N. Tomii, C. Hirai Evaluation Method of Robustness for Train Schedules *Quarterly Report of RTRI*, Vol. 48, N°4, 2007.
- S. Tréfond, H. Djellab, A. Billionnet, S. Elloumi Planification robuste du matériel roulant ferroviaire *ROADEF*, 13ème congrès annuel de la Société française de Recherche Opérationnelle et d'Aide à la Décision, Angers, France, pages 544-545, 2012.
- S. Tréfond, H. Djellab, E. Escobar, A. Billionnet, S. Elloumi A robust-planning methodology for railway rolling-stock *COMPRAIL*, 13th International Conference on Design and Operation in Railway Engineering, Lyndhurst, New Forest, UK, pages 349 - 357, 2012.
- S. Tréfond, H. Djellab, A. Billionnet, S. Elloumi Optimisation et simulation pour la planification robuste des roulements d'engins en milieu ferroviaire *ROADEF*, 14ème congrès annuel de la Société française de Recherche Opérationnelle et d'Aide à la Décision, Troyes, France, Vol. 2, page 241, 2013.
- S. Tréfond, H. Djellab, A. Billionnet, S. Elloumi A column generation based method for robust railway rolling-stock planning *TRISTAN*, 8th triennial Symposium on transportation analysis, San Pedro de Atacama, Chili, page 4, 2013.
- S. Vianey Robustesse dans la planification de la production ferroviaire *Rapport technique interne*, référence N°GDA/ARR/7237070418, 2007.
- M. Vromans, R. Dekker, L. Kroon Reliability and heterogeneity of railway services *ERIM Report*, n°ERS-2003-090-LIS, 2003.

Résumé

Cette thèse est consacrée à la planification robuste du matériel roulant ferroviaire dans un contexte de transport de voyageurs régional. Il s'agit de déterminer pour chaque engin une suite de trajets à réaliser sur une période donnée de façon à résister au mieux aux aléas pouvant apparaître en opérationnel. Dans ce but, nous proposons une définition et une caractérisation de la robustesse par des indicateurs propres au contexte d'étude. Nous abordons le problème par une approche structurelle différente des approches robustes classiques qui permet d'agir sur la structure d'une solution en fonction des indicateurs définis. Trois méthodes sont alors mises en œuvre dans le cadre de ce travail. Une résolution approchée par une heuristique gloutonne et une recherche locale permet d'obtenir des solutions rapidement. Une méthode de résolution approchée par génération de colonnes a été développée afin de prendre en compte plus de contraintes de façon intégrée. Une méthode basée sur un programme linéaire en nombres entiers résolu exactement traite un problème plus général. Afin de limiter le coût de la robustesse, ces méthodes sont basées sur un existant à SNCF répondant au problème de planification des ressources matérielles à coûts de production minimaux. Un outil de simulation du comportement des solutions en situation perturbée permet de comparer les solutions entre elles par évaluation des indicateurs de robustesse. Des expérimentations sur des instances réelles ont prouvé la pertinence des approches et ont mené à l'industrialisation d'un prototype.

Mots-clés : Transport ferroviaire, planification, roulements engins, robustesse, optimisation.

Abstract

Robust railway rolling-stock planning

This thesis deals with robust rolling-stock planning problems for passenger regional trains. It consists in building robust rolling-stock schedules to operate trains under technical constraints while anticipating operational disturbances that can occur. First of all, we define indicators to characterize robustness in context. We use these indicators to have an effect on solutions that we build. This structural approach is unusual compared to classical robust optimization approaches. We have implemented three methods to solve the problem. A sequential heuristic method enhanced by a local search gives solutions quickly. A column-generation method calculates approximate solutions. An integer linear program is solved exactly to obtain solutions to a global problem. These methods are based on an existing tool at SNCF that optimizes the rolling-stock planning problem to assure optimal production costs. A simulation tool evaluates robustness indicators to compare solutions. Tests on real instances have proved the relevance of the approaches and have lead to the use of a prototype in production.

Keywords : railways, rolling-stock planning, robustness, optimization.