

Université de Montréal

Approches heuristiques pour le problème d'ordonnement de véhicules

par
Dumitru Silviu Craciunas

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

Dec, 2006

© Dumitru Silviu Craciunas, 2006.



AVIS

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

NOTICE

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal
Faculté des études supérieures

Ce mémoire intitulé:

Approches heuristiques pour le problème d'ordonnement de véhicules

présenté par:

Dumitru Silviu Craciunas

a été évalué par un jury composé des personnes suivantes:

Jacques Ferland,	président-rapporteur
Jean-Yves Potvin,	directeur de recherche
Michel Gendreau,	codirecteur
Bernard Gendron,	membre du jury

Mémoire accepté le: 19 avril 2007

RÉSUMÉ

Le problème consiste à ordonnancer la production de plusieurs types d'automobiles sur une ligne de production, sujet à plusieurs contraintes. Les contraintes sont associées aux différents ateliers de l'usine, soit l'atelier de peinture, l'atelier de montage et l'atelier de tôlerie. Les violations des contraintes molles sont pénalisées dans l'objectif et, selon leur importance, on retrouve différents objectifs hiérarchiques. L'heuristique proposée essaie de satisfaire les contraintes les plus importantes en premier, avant de traiter les autres. Pour faire ceci, une analyse statistique est réalisée afin d'évaluer la difficulté de chaque contrainte. L'analyse est ensuite utilisée pour faire des ajustements au niveau des paramètres des heuristiques. Une construction heuristique est d'abord proposée pour déterminer rapidement une bonne solution de départ. Ensuite, une recherche tabou est appliquée pour améliorer la solution de départ.

Mots clés: Recherche Tabou, heuristiques, ordonnancement de véhicules.

ABSTRACT

In the car sequencing problem, one has to decide in what order the cars must be processed on a production line for one operation day, subject to many constraints. These constraints are associated with three different shops in the factory: body shop, paint shop and assembly line. Violations of soft constraints are penalized in the objective and, depending on their priorities, different hierarchical objectives are obtained. The proposed heuristics try to satisfy higher-priority constraints first. To this end, a statistical analysis is performed to evaluate the difficulty of each constraint and adjust the parameters of the heuristics accordingly. A construction heuristic is first proposed to quickly find a good starting solution. Then, a tabu search is applied to improve this solution.

Keywords: Tabu Search, heuristics, car sequencing.

TABLE DES MATIÈRES

RÉSUMÉ	iii
ABSTRACT	iv
TABLE DES MATIÈRES	v
LISTE DES TABLEAUX	viii
LISTE DES FIGURES	ix
DÉDICACE	x
REMERCIEMENTS	xi
CHAPITRE 1 : INTRODUCTION	1
1.1 Problématique	1
1.2 Description du problème d'ordonnement de véhicules	2
1.2.1 Les ateliers de montage et de tôlerie	2
1.2.2 L'atelier de peinture	3
1.2.3 Rôle de la journée précédente	3
1.2.4 Objectifs	3
1.3 Revue de la littérature	4
1.4 Contributions	5
1.4.1 Pendant le challenge ROADEF	5
1.4.2 Après le challenge ROADEF	6
1.5 Plan du mémoire	6
CHAPITRE 2 : ALGORITHME GÉNÉRAL	7
2.1 Algorithme générique	7
2.2 Difficulté d'une l'instance	8
2.3 Relaxation	8
2.4 Heuristiques de construction	10
2.5 Améliorations de deuxième ou troisième niveau	11
2.6 Recherche tabou	11
CHAPITRE 3 : HEURISTIQUES DE CONSTRUCTION	12
3.1 Traitements des contraintes de ratio	12
3.1.1 Difficulté des contraintes de ratio	13
3.1.2 Heuristiques de construction pour les problèmes <i>EP_ENP</i> et <i>EP_RAF</i>	15

3.1.3	Coût global d'insertion	17
3.1.4	Heuristique de construction dynamique pour les problèmes <i>EP_ENP</i> et <i>EP_RAF</i>	18
3.1.5	Heuristique de construction stochastique pour les problèmes <i>EP_ENP</i> et <i>EP_RAF</i>	19
3.2	Heuristique de construction pour le problème <i>RAF_EP</i>	20
3.2.1	Préserver l'optimalité au niveau du nombre de changements de couleur	20
3.3	Améliorations de 2 ^{ème} ou 3 ^{ème} niveau	23
CHAPITRE 4 : LA RECHERCHE TABOU		25
4.1	Description générale	25
4.1.1	L'espace de recherche	26
4.1.2	Listes tabou	26
4.1.3	Niveaux d'aspiration	26
4.1.4	Critères d'arrêt	26
4.1.5	Intensification et diversification	27
4.2	Implémentation de la recherche tabou pour l'ordonnancement de véhicules	28
4.2.1	Initialisation	28
4.2.2	L'espace de recherche	28
4.2.3	Voisinages	29
4.2.4	Voisinage par échange	29
4.2.5	Voisinage par extraction\insertion	30
4.2.6	Le choix des candidats en vue d'un mouvement	30
4.2.7	Listes tabou	31
4.2.8	Niveau d'aspiration	32
4.2.9	Critère d'arrêt	32
4.2.10	Intensification et diversification	33
4.3	La complexité de la recherche tabou	35
4.4	Techniques avancées	37
4.4.1	Bornes sur les valeurs des échanges	37
4.4.2	Niveaux	37
CHAPITRE 5 : VOISINAGES UTILISÉS DANS LA RECHERCHE TABOU		39
5.1	Introduction	39
5.2	Voisinage par échange	40
5.2.1	Le coût d'un <i>échange(i, j)</i> pour les contraintes de ratio	40
5.2.2	Le coût d'un <i>échange(i, j)</i> pour le nombre de purges de peinture	47
5.3	Voisinage par extraction\insertion	52
5.3.1	Le coût d'une <i>ExtIns(i, j)</i> pour les contraintes de ratio	53
5.3.2	Le coût d'une <i>ExtIns(i, j)</i> pour le nombre de purges de peinture	56

CHAPITRE 6 : RÉSULTATS	61
6.1 Instances <i>EP_ENP</i>	62
6.2 Instances <i>EP_RAF</i>	66
6.3 Instances <i>RAF_EP</i>	69
6.4 Étude de sensibilité des paramètres	72
6.5 Évolution moyenne de la solution dans le temps	74
6.5.1 Graphiques sur l'évolution de la meilleure solution	74
6.5.2 Graphiques sur l'évolution de la solution courante	74
CHAPITRE 7 : CONCLUSIONS	82
BIBLIOGRAPHIE	84

LISTE DES TABLEAUX

6.1	Provenance des instances	62
6.2	Description des instances <i>EP_ENP</i>	64
6.3	Résultats pour les instances <i>EP_ENP</i>	65
6.4	Description des instances <i>EP_RAF</i>	67
6.5	Résultats pour les instances <i>EP_RAF</i>	68
6.6	Description des instances <i>RAF_EP</i>	70
6.7	Résultats pour les instances <i>RAF_EP</i>	71
6.8	Analyse des résultats selon la taille des instances <i>EP_ENP</i>	72
6.9	Analyse des résultats selon la taille des instances <i>EP_RAF</i>	72
6.10	Analyse des résultats selon la taille pour les instances <i>RAF_EP</i>	73
6.11	Analyse des résultats pour les instances <i>EP_ENP</i> selon le nombre de modèles <i>M_{EP}</i>	73
6.12	Analyse des résultats pour les instances <i>EP_RAF</i> selon le nombre de modèles <i>M_{EP}</i>	73
6.13	Analyse des résultats pour les instances <i>RAF_EP</i> selon le nombre de modèles <i>M_{EP}</i>	74

LISTE DES FIGURES

6.1	Évolution moyenne de la meilleure solution rencontrée sur les instances EP_ENP	76
6.2	Évolution moyenne de la solution courante sur les instances EP_ENP	77
6.3	Évolution moyenne de la meilleure solution rencontrée sur les instances EP_RAF	78
6.4	Évolution moyenne de la solution courante sur les instances EP_RAF	79
6.5	Évolution moyenne de la meilleure solution rencontrée sur les instances RAF_EP	80
6.6	Évolution moyenne de la solution courante sur les instances RAF_EP	81

À mes parents

REMERCIEMENTS

Je tiens à remercier tout particulièrement mon directeur M. Jean-Yves Potvin pour toute l'assistance qu'il m'a donnée au long de mon travail. Je désire aussi remercier mon co-directeur M. Michel Gendreau pour son soutien et intérêt dans ce projet.

Je suis également reconnaissant envers le CRSNG et CRT pour l'appui accordé sur cette recherche.

J'ajouterai à mes remerciements mes amis Julio Montecinos et Tudor Tutu pour avoir partagé avec moi leurs propres expériences et un grand merci à ma famille pour leur support moral et leurs conseils.

CHAPITRE 1

INTRODUCTION

1.1 Problématique

Dans les années 1925, la couleur la moins dispendieuse et qui séchait le plus rapidement était le noir. Cela devait mener à la fameuse phrase de Henry Ford “vous pouvez choisir n’importe quelle couleur pourvu qu’elle soit noire”. Cela en disait long sur les lignes de production du début du siècle, où un produit standardisé permettait facilement la production en masse. Les choses ont beaucoup changé depuis dans l’industrie automobile. Une compétition accrue entre les compagnies manufacturières a fait en sorte que les désirs des clients influencent directement la production. Ainsi, une large gamme d’options et accessoires sont maintenant disponibles.

Le premier défi est d’harmoniser la diversité de ces options avec la production de masse. En dépit de toutes les options offertes aux clients, les véhicules sont toujours assemblés sur une seule ligne de production. Plusieurs véhicules consécutifs ayant la même option peuvent sur-solliciter un poste de travail et causer des blocages ailleurs sur la ligne de production. Ainsi, les postes de travail qui suivent seront utilisés en dessous de leur capacité au tout début et seront surchargés par la suite.

Pour les options difficiles à réaliser, telles qu’un toit décapotable, une solution pratique est d’avoir de courtes lignes secondaires le long de la ligne principale. Ceci permet de traiter l’option problématique, sans causer de délais sur la ligne principale. Une fois l’option installée, le véhicule est réinséré sur la ligne principale. Plusieurs types de lignes auxiliaires sont décrites en [13]. Par contre, une ligne auxiliaire entraîne des coûts additionnels et la réinsertion du véhicule sur la ligne principale peut causer des problèmes.

Outre le lissage de la charge du travail, qui permet de distribuer la charge uniformément sur toute la ligne de production, la consommation des composantes qui sont montées systématiquement sur tous les véhicules doit être aussi homogène que possible pour permettre un bon approvisionnement [8]. Ceci entraîne la nécessité d’une planification de la production. D’une façon générale, ce problème est connu sous le nom de **problème d’ordonnement sur une ligne de production avec produits hétérogènes**.

Dans l’industrie automobile s’ajoutent d’autres contraintes techniques liées aux rafales de peinture. Il y a quelques années, la séquence de véhicules pouvait être triée localement à l’entrée des véhicules dans l’atelier de peinture. Ensuite, la séquence de véhicules était triée à nouveau en entrée de l’atelier de montage. Ainsi, on pouvait traiter ces contraintes indépendamment des autres contraintes. Dernièrement, on a toutefois commencé à imposer un respect strict d’un ordonnancement prévisionnel qui interdit les tris locaux [12].

1.2 Description du problème d'ordonnement de véhicules

La description suivante suit la description du problème faite lors du challenge ROADEF'05 [5, 12].

Le problème d'ordonnement de véhicules consiste à planifier la production d'automobiles sur une ligne pour un horizon d'une journée. La ligne est généralement composée de trois ateliers : l'atelier de tôlerie, l'atelier de peinture et l'atelier de montage. Chaque atelier a ses propres contraintes et objectifs, souvent contradictoires avec ceux des autres ateliers.

1.2.1 Les ateliers de montage et de tôlerie

Bien que les ateliers de montage et de tôlerie nécessitent une certaine planification, les besoins de l'atelier de montage sont beaucoup plus importants. C'est pourquoi les besoins de l'atelier de tôlerie sont intégrés au sein du problème à l'aide de contraintes de ratio non prioritaires (voir plus loin).

Afin de lisser la charge de travail sur les différents postes en bord de chaîne de l'atelier de montage, on cherche à "espacer" les véhicules "difficiles", soit ceux dont les équipements ou options nécessitent des opérations lourdes. En d'autres termes, on cherche à limiter la densité des véhicules difficiles pour ne pas surcharger les postes de travail impliqués dans l'assemblage de ces véhicules.

Ce besoin d'espacement est formalisé par une contrainte de ratio N/P , où P est la longueur d'une sous-séquence et N est le nombre maximal de véhicules dans la sous-séquence auxquels doit s'appliquer la contrainte de ratio au sein de la sous-séquence. Par exemple, si $N/P = 3/5$ pour une contrainte de ratio donné, alors on ne doit pas trouver plus de 3 véhicules soumis à cette contrainte dans toute sous-séquence de 5 véhicules au sein de la séquence complète. Une contrainte de ratio porte sur une caractéristique technique qui nécessite des opérations particulières (par exemple, le toit ouvrant, la climatisation, etc.).

On définit deux catégories de contraintes de ratio, les prioritaires et les non prioritaires. Les contraintes de ratio prioritaires correspondent à des contraintes "lourdes" de l'atelier de montage, les non prioritaires étant associés à des opérations plus légères. On veut ainsi privilégier la satisfaction des contraintes de ratio prioritaires sur les contraintes de ratio non prioritaires. En d'autres termes, on veut éviter que le respect d'un ratio associé à une opération légère se fasse au détriment d'un ratio associé à une opération lourde. Les contraintes de ratio non-prioritaires permettent aussi de faire un arbitrage entre les contraintes de ratio prioritaires et les besoins de l'atelier de peinture. Le lissage de la consommation des composantes et les besoins de l'atelier de tôlerie sont également représentés par des contraintes de ratio non-prioritaires.

Les contraintes de ratio sont des contraintes molles : on ne peut pas savoir a priori si l'ensemble des contraintes de ratio peut être respecté lorsqu'on génère une journée de production. L'objectif consiste donc à minimiser le nombre de violations des contraintes

de ratio.

1.2.2 L'atelier de peinture

L'objectif de l'atelier peinture est de minimiser la consommation de solvant, lequel solvant est utilisé pour purger les pistolets de peinture à chaque changement de couleur dans la séquence de véhicules.

On cherche donc à grouper les véhicules de même couleur, ce qui revient à minimiser le nombre de purges, soit le nombre de changements de couleurs dans la séquence. On veut donc générer les rafales de peinture les plus longues possibles. Toutefois, chaque rafale ne doit pas dépasser une longueur maximale, car il faut périodiquement purger les pistolets de peinture, même s'il n'y a pas eu de changements de couleur. Cette dernière contrainte est une contrainte dure.

1.2.3 Rôle de la journée précédente

Lorsqu'on ordonne les véhicules de la journée J , on doit tenir compte des derniers véhicules de la journée $J - 1$. On considère que ces véhicules sont déjà ordonnés et que leurs positions sont figées pour l'ordonnancement de la journée J . Il faut donc tenir compte des derniers véhicules de la journée $J - 1$ pour compter les violations de contraintes de ratio sur la journée J . En revanche, on ne tiendra pas compte de la journée $J + 1$.

1.2.4 Objectifs

Il s'agit de créer un ordonnancement de véhicules satisfaisant au mieux les contraintes de l'atelier de peinture et ceux de l'atelier de montage. On peut adopter deux stratégies possibles selon les usines :

- “contraintes de ratio en priorité” : les contraintes de l'atelier de montage sont prioritaires par rapport à celles de l'atelier de peinture
- “rafales de peinture en priorité” : les contraintes de l'atelier de peinture sont prioritaires par rapport à celles de l'atelier de montage.

On cherche à optimiser trois objectifs, du plus prioritaire au moins prioritaire, afin de construire un ordonnancement pour les véhicules. On obtient ainsi différents types de problèmes dépendant de l'organisation hiérarchique des différents objectifs :

1. problèmes *EP_RAF* (écarts prioritaires, rafales de peinture)
 - (a) minimiser le nombre de violations des contraintes de ratio prioritaires
 - (b) minimiser le nombre de purges de peinture
 - (c) minimiser le nombre de violations des contraintes de ratio non prioritaires

ou

1. problèmes *EP_ENP* (écarts prioritaires, écarts non prioritaires)
 - (a) minimiser le nombre de violations des contraintes de ratio prioritaires
 - (b) minimiser le nombre de violations des contraintes de ratio non prioritaires
 - (c) minimiser le nombre de purges de peinture

ou

1. problèmes *RAF* (rafales de peinture, écarts prioritaires)
 - (a) minimiser le nombre de rafales de peinture
 - (b) minimiser le nombre de violations des contraintes de ratio prioritaires
 - (c) minimiser le nombre de violations des contraintes de ratio non prioritaires

Cette organisation hiérarchique est obtenue en pénalisant les différents objectifs dans une fonction objective agrégée :

$$\sum_{i=1,2,3} M_i * \text{objectif de } i^{\text{ème}} \text{ niveau}$$

où $M_1 = 10^6$, $M_2 = 10^3$ et $M_3 = 1$.

Il faut noter que certaines usines ne déclarent pas de contraintes de ratio non prioritaires. Dans tous les cas, la seule contrainte dure correspond au respect de la longueur maximale des rafales de peinture.

1.3 Revue de la littérature

Le problème d'ordonnement de véhicules est un problème NP-complet tel que démontré par I.P. Gent dans [7].

Plusieurs algorithmes exacts ont été développés, en utilisant soit une exploration de l'espace des solutions, [8], soit la programmation en nombres entiers [4, 11, 13]. Ces algorithmes sont toutefois limités quant à la taille des problèmes qui peuvent être abordés et les temps de calcul nécessaires. Avec l'augmentation de la puissance de calcul des ordinateurs et le développement de meilleures approches de résolution, on arrive maintenant à résoudre de manière exacte des problèmes comportant environ 300 véhicules [4, 11].

Plusieurs méthodes heuristiques ont été proposées pour résoudre ce problème [8]. Les approches gloutonnes commencent avec une solution vide et ajoutent à chaque itération le meilleur véhicule selon un certain critère de sélection. Une comparaison entre l'efficacité des différents critères de sélection est faite en [10]. Nous avons également développé plusieurs heuristiques de construction dans ce mémoire. Les critères de sélection et les améliorations que nous y avons apportées sont présentés au chapitre 3.

Les algorithmes de colonie de fourmis artificielles, qui ont également été appliqués à ce problème, reposent en grande partie sur les critères de sélection des approches gloutonnes. A chaque itération de l'algorithme, de nouvelles solutions sont construites par les fourmis avec une amélioration des performances d'une itération à l'autre obtenue par

un mécanisme de dépôt de phéromone. Cette dernière représente la mémoire du système et permet aux fourmis de se concentrer sur les régions de l'espace de recherche qui sont les plus intéressantes [10, 11].

La recherche locale est, à ce jour, l'approche la plus utilisée et la plus efficace pour résoudre le problème d'ordonnement de véhicules [3, 10]. Une solution de départ est souvent générée par des algorithmes gloutons. D'une itération à l'autre, la meilleure solution est choisie dans le voisinage de la solution courante. Généralement, l'exploration utilise des voisinages générés par des modifications mineures à la solution courante [10], mais on retrouve aussi des approches à large voisinage [3, 4].

D'autres méta-heuristiques, telles que la recherche locale itérée et la recherche tabou, reposent sur le concept de recherche locale, mais ajoutent des éléments de diversification et d'intensification [1, 14, 15]. Notre algorithme de recherche tabou fait partie de ces techniques et sera décrit aux chapitres 4 et 5.

Pour tester notre approche, nous allons nous intéresser au cas réel de Renault présenté dans le cadre de la compétition ROADEF'05 [5]. Les instances comportent en moyenne 1200 véhicules, environ 26 modèles de véhicules distincts, différents types de contraintes et un temps de calcul limité à 10 minutes. Ceci exige l'utilisation d'heuristiques. Un résumé des approches utilisées pendant le challenge et les résultats obtenus se trouvent en [16].

1.4 Contributions

1.4.1 Pendant le challenge ROADEF

La plus grande partie de ce projet a été développée pendant le challenge ROADEF.

Une première contribution originale se trouve au niveau des techniques de construction. Ces techniques reposent en particulier sur le nombre de violations introduites par une insertion ainsi que sur la notion de niveau de difficulté d'un véhicule. Cette dernière notion est nouvelle et s'appuie sur le nombre de contraintes actives qui s'appliquent à un véhicule. On introduit également dans ce mémoire des mesures de corrélation entre les différentes contraintes de ratio qui s'appliquent à un véhicule afin d'estimer plus précisément son niveau de difficulté. Pour les problèmes de type *RAF_EP*, on démontre également que la technique de construction proposée permet d'obtenir une solution optimale pour l'objectif de premier niveau.

En ce qui concerne la recherche tabou, la contribution originale se trouve au niveau des techniques de calcul qui permettent une évaluation très rapide des voisinages. Ainsi, grâce à des valeurs pré-calculées, l'échange de deux véhicules peut être évalué en un temps constant. Nous avons également développé une méthode originale pour sélectionner les véhicules candidats à un échange qui intègre des éléments de nature stochastique (voir chapitre 4). Cette approche permet une convergence rapide, une meilleure diversification et élimine tout risque de cyclage.

1.4.2 Après le challenge ROADEF

Après la fin du challenge ROADEF, nous avons amélioré encore davantage la structure générale et le paramétrage de notre algorithme. Entre autres, la gestion des trois objectifs a été modifiée. Dans la version utilisée pendant la compétition, les trois objectifs étaient activés de façon séquentielle du plus prioritaire au moins prioritaire. Le moment où on ajoutait les deuxième et troisième objectifs dépendaient de paramètres tenant compte de la difficulté des instances et du type de problème, ce qui était assez difficile à gérer. De plus, une fois qu'on laissait tomber un objectif pour s'intéresser au suivant selon l'ordre hiérarchique, on ne pouvait plus revenir en arrière. Pour contrer ces désavantages, nous proposons une nouvelle approche où les prochains objectifs considérés dépendent d'une loi de probabilité qui est appliquée à chaque fois que la recherche tabou est appelée (voir chapitre 2).

1.5 Plan du mémoire

Le chapitre 2 offre une vue d'ensemble de notre projet. On y présente le schéma général de l'algorithme utilisé.

Le chapitre 3 traite des techniques de construction. On définit premièrement le concept de niveau de difficulté d'un véhicule. On présente ensuite deux approches de construction, l'une dynamique et l'autre stochastique, pour les problèmes de type *EP_ENP* et *EP_RAF*. On décrit aussi l'approche de construction permettant d'obtenir une solution optimale pour l'objectif de premier niveau dans le cas des problèmes de type *RAF_EP*.

Les chapitres 4 et 5 traitent de la recherche tabou. Dans le chapitre 4, on présente plusieurs variantes de la recherche tabou, chacune ayant une efficacité plus grande selon son contexte d'utilisation et le type de problèmes abordé. On montre les avantages des éléments stochastiques introduits dans la recherche afin d'éviter les comportements cycliques et créer de la diversification. Enfin, on présente les paramètres de la recherche Tabou qui permettent de gérer l'amplitude des perturbations à la solution courante et la profondeur de la recherche. Le chapitre 5 décrit les voisinages utilisés au sein de la recherche tabou et souligne l'importance des pré-calculs qui contribuent à l'efficacité de la méthode.

Le chapitre 6 décrit les instances utilisées afin d'évaluer nos approches de résolution. On présente ensuite des résultats numériques regroupés selon le type de problème, soit *EP_ENP*, *EP_RAF* et *RAF_EP*.

Le chapitre 7 résume le tout et souligne à nouveau les concepts importants à la base de l'efficacité de notre approche.

CHAPITRE 2

ALGORITHME GÉNÉRAL

Nous proposons ici des approches heuristiques afin de résoudre le problème d'ordonnement de véhicules. Il s'agit, en l'occurrence, de méthodes de construction permettant de générer des solutions initiales, suivies d'une recherche tabou pour améliorer ces solutions.

2.1 Algorithme générique

Selon la hiérarchie des trois objectifs, on obtient des problèmes très différents. Ainsi, pour certaines instances de type *EP*, nous trouvons facilement une solution optimale pour le premier objectif et l'emphase doit donc porter sur l'optimisation de deuxième niveau. Pour d'autres instances de type *EP*, nous devons passer plus de temps au premier niveau afin de trouver la meilleure solution possible pour les contraintes de ratio prioritaires.

Par ailleurs, dans les instances de type *EP*, on tend à espacer les véhicules, alors que dans les instances de type *RAF*, on tend à les rapprocher. Il faut donc adapter les heuristiques utilisées et les paramètres au sein de notre algorithme en fonction de la hiérarchie des objectifs.

Nous pouvons quand même décrire une structure générique qui s'applique dans tous les cas. Soit t le temps courant, T le temps total disponible pour trouver une solution, ainsi que T_1 et T_2 , $T_1 < T_2 < T$, dont les valeurs dépendent de la difficulté de l'instance et de la hiérarchie des objectifs. Nous avons :

1. Initialiser les structures de données
2. Déterminer la difficulté de l'instance et ajuster les paramètres en conséquence.
3. Traiter, selon le cas, seulement le premier objectif ou bien les deux premiers objectifs.
4. Créer une solution S avec une heuristique de construction et poser $S^1 \leftarrow S$.
 - (a) Tant que $t < T_1$:
 - Appliquer la recherche tabou à partir de S pour un petit nombre d'itérations afin d'obtenir une solution améliorée S' .
 - Si S' est de meilleure qualité que S^1 alors $S^1 \leftarrow S'$.
 - Créer une nouvelle solution S avec une heuristique de construction.
 - (b) Appliquer la recherche tabou à la meilleure solution S^1 . Soit S^2 la solution courante à la fin de la recherche.
 - (c) Tant que $t < T_2$:

- Appliquer la recherche tabou à S^2 avec un nouvel ensemble de valeurs pour les paramètres. Soit S^3 la solution courante à la fin de la recherche.
- 5. Ajouter les objectifs mis de côté.
- 6. Appliquer des techniques d'amélioration de deuxième ou troisième niveau à S^3 , selon les objectifs ayant été mis de côté. Soit S^4 la solution obtenue.
- 7. Tant que $t < T$:
 - Appliquer la recherche tabou à S^4 avec un nouvel ensemble de valeurs pour les paramètres.
- 8. Retourner la meilleure solution rencontrée.

Nous allons maintenant présenter rapidement les principaux éléments de ce cadre de résolution. Les détails seront décrits dans les chapitres qui suivent.

2.2 Difficulté d'une l'instance

Afin d'améliorer la performance des heuristiques de construction, une analyse statistique est utilisée afin d'établir la difficulté de l'instance à résoudre, en particulier le niveau de difficulté associé à chaque véhicule. Pour les contraintes de ratio, des bornes inférieures sur la valeur de l'objectif nous permettent de déterminer la difficulté de l'instance. En ce qui a trait au niveau de difficulté associé à un véhicule, nous estimons d'abord la difficulté de chacune des contraintes de ratio qui s'appliquent au véhicule. Les corrélations observées entre les différentes contraintes de ratio sont ensuite exploitées afin d'améliorer les estimations initiales. Toutes ces estimations sont réajustées de façon dynamique lors de l'application des heuristiques de construction.

Dans le cas où l'objectif de premier niveau est la minimisation des purges de couleur, il est possible de calculer le nombre optimal de purges de peinture et d'atteindre ce nombre avec l'heuristique de construction seule. Les détails sont fournis au Chapitre 3.

2.3 Relaxation

Le fait de travailler avec des objectifs hiérarchiques à trois niveaux suggère, dans une première phase, d'ignorer les objectifs de deuxième et troisième niveau (ou seulement de troisième niveau), de même que la contrainte dure touchant à la longueur maximale pour une sous-séquence de véhicules de même couleur. Un gain important est alors obtenu au niveau des temps de calcul, puisque les pénalités pour les objectifs qui sont ignorés ne sont plus calculés. Dans le cadre de la recherche tabou, ceci permet aussi d'éviter des cas pathologiques où l'algorithme passe beaucoup de temps à améliorer la solution au troisième niveau seulement. L'introduction ultérieure des objectifs de niveau inférieur nous fait ensuite visiter d'autres régions de l'espace de recherche, ce qui peut être bénéfique, même pour les objectifs de niveau supérieur.

La relaxation proposée initialement s'appelle *Relaxation séquentielle* puisque la mise à l'écart des objectifs de niveau inférieur tient jusqu'au temps T_2 . A partir de ce moment, au moins un nouvel objectif de niveau inférieur s'ajoute et demeure actif jusqu'à la fin. L'évolution de l'algorithme est très sensible au choix de T_2 . Si on introduit les objectifs de niveau inférieur trop tôt, il y a le risque de se trouver assez loin d'une solution optimale pour les objectifs de niveau supérieur tout en investissant trop de temps de calcul avec les objectifs de niveau inférieur. Si on introduit les objectifs de niveau inférieur trop tard, il y a le risque d'allouer trop de temps de calcul aux objectifs de niveau supérieur, alors qu'on est déjà à l'optimum ou tout près de l'optimum pour ces derniers.

De façon générale, la valeur de T_2 est d'autant plus petite que la taille du problème est petite et que le nombre de contraintes de ratio non prioritaires est petit. Par ailleurs, si la valeur des objectifs de niveau supérieur ne peut plus être améliorée (parce qu'on a atteint une borne inférieure), on introduit aussitôt les objectifs de niveau inférieur et la valeur de T_2 est automatiquement déterminée.

L'évolution de l'algorithme s'est révélée très sensible aux valeurs des différents paramètres. Nous avons donc développé récemment une nouvelle approche, de nature à la fois dynamique et stochastique, appelée *Relaxation stochastique*. Ici, à chaque fois que la recherche tabou est appelée, les objectifs qui doivent être activés sont choisis de manière stochastique.

Les configurations possibles sont les suivantes :

- 1 : seul l'objectif de premier niveau est activé,
- 2 : seuls les objectifs de premier et deuxième niveau sont activés,
- 3 : tous les objectifs sont activés.

A chaque configuration $i = 1, 2, 3$, on associe une probabilité p_i . Quand on appelle la recherche tabou, une configuration est choisie en fonction de cette distribution de probabilités. Les valeurs des probabilités sont influencées par la taille du problème, le nombre de contraintes de ratio prioritaires et non-prioritaires et la valeur de la meilleure solution rencontrée.

Soit N le nombre de véhicules, N_R^{EP} le nombre de contraintes de ratio prioritaires, N_R^{ENP} le nombre de contraintes de ratio non-prioritaires, S^* la meilleure solution rencontrée et S^0 la solution de départ. Par ailleurs, dans ce qui suit, le nombre 1200 correspond à un estimé du nombre de véhicules dans une instance et le nombre 10 à un estimé du nombre de types de contraintes de ratio prioritaires et non prioritaires. En pratique, ces estimés peuvent facilement être ajustés selon les instances considérées.

Dans le cas des problèmes de type EP_{RAF} et EP_{ENP} , nous avons les probabilités suivantes :

- avec $p_1 = \min(N/1200, 0.90) * \min(N_R^{EP}/10, 0.90) * \max(0.3, Val(S^*)/Val(S^0))$, seul le premier objectif est activé. Le premier terme est d'autant plus grand que la taille de l'instance est grande. Par ailleurs, ce premier terme ne peut excéder 0.9 (afin d'éviter d'obtenir une valeur supérieure à 0.9 pour la probabilité p_1). Le second terme est d'autant plus grand que le nombre de contraintes de ratio priori-

taires est élevé. Enfin, le troisième terme est d'autant plus petit que l'amélioration apportée à la solution de départ S^0 est importante. Donc, on travaille davantage avec l'objectif de premier niveau lorsque l'instance est de grande taille et que le nombre de contraintes de ratio prioritaires est élevé, ce qui correspond à des instances difficiles à résoudre. Par ailleurs, lorsqu'on a déjà obtenu une amélioration importante de la solution de départ, on peut se permettre de s'attaquer aux autres objectifs, et la valeur de p_1 est alors plus petite.

- avec $p_2 = \min(1 - p_1, \min(N/1200, 0.90) * \min(N_R^{ENP}/10, 0.90))$, les deux premiers objectifs sont activés. Nous ne faisons pas intervenir ici l'amélioration apportée à la solution de départ. En effet, les améliorations au deuxième niveau sont beaucoup moins importantes que celles apportées au premier niveau.
- avec $p_3 = \max(0, 1 - p_1 - p_2)$, les trois objectifs sont activés.

Dans le cas des problèmes de type *RAF*, on obtient la solution optimale pour l'objectif de premier niveau avec la technique de construction proposée. Ainsi, on ne travaille jamais avec le premier objectif seulement et $p_1 = 0$.

Dans les cas des problèmes de type *RAFEP*, nous avons donc :

- avec $p_1 = 0$, le premier objectif est activé,
- avec $p_2 = \min(N/1200, 0.90) * \min(N_R^{ENP}/10, 0.90)$, les deux premiers objectifs sont activés,
- avec $p_3 = 1 - p_2$, les trois objectifs sont activés.

Ces formules permettent :

- de travailler davantage sur les objectifs de premier niveau si ceux-ci sont difficiles à optimiser ;
- d'exclure l'objectif de troisième niveau s'il est difficile à optimiser (sinon celui-ci risque d'entraîner beaucoup de calculs additionnels au détriment des niveaux supérieurs) ;
- de ne pas traiter exclusivement l'objectif de premier niveau, même si son niveau de difficulté est élevé.

Si jamais les objectifs de niveau supérieur ne peuvent plus être améliorés (atteinte d'une borne inférieure), la probabilité qui leur est associée devient automatiquement nulle. Dans tous les cas, ces règles ont été déterminées suite à des tests expérimentaux préliminaires sur les instances de la ROADEF. Cette approche augmente de beaucoup la robustesse de notre algorithme et améliore sa performance globale.

2.4 Heuristiques de construction

Partant avec la séquence de véhicules de la journée $J - 1$, les heuristiques de construction insèrent à chaque itération un nouveau véhicule dans la séquence courante parmi ceux de la journée J qui sont encore disponibles.

L'objectif est de construire une bonne solution de départ dans un temps assez court. De plus, la plupart des techniques de construction utilisées sont des heuristiques non

déterministes qui nous permettent d'obtenir différentes solutions de départ pour la recherche tabou. Ceci réduit donc les chances d'aboutir dans un mauvais minimum local.

2.5 Améliorations de deuxième ou troisième niveau

Ces techniques permettent d'améliorer rapidement la solution au deuxième (ou troisième niveau) tout en préservant ou en améliorant la valeur de la solution pour les niveaux supérieurs. Essentiellement, des échanges de véhicules sont réalisés pendant un certain nombre d'itérations dans une perspective d'optimisation locale. Une difficulté ici vient de ce que la structure de la solution aux niveaux supérieurs agit souvent comme une contrainte forte et limite les choix possibles au deuxième (ou troisième) niveau. Nous présentons les détails au chapitre 3.

2.6 Recherche tabou

Afin d'améliorer les solutions dans une perspective globale, nous faisons appel à une recherche tabou. À partir de la solution courante, l'algorithme considère un voisinage de cette solution en déplaçant des véhicules dans la séquence. On espère ainsi que la meilleure solution dans ce voisinage sera de meilleure qualité que la solution courante. Cette meilleure solution est alors choisie, même si elle est de moins bonne qualité que la solution courante. Ceci permet à la méthode de s'échapper des minima locaux. La procédure est ensuite répétée avec la nouvelle solution courante.

Nous présentons la recherche tabou au chapitre 4 et les voisinages utilisés au chapitre 5.

CHAPITRE 3

HEURISTIQUES DE CONSTRUCTION

Dans ce chapitre, nous décrivons les méthodes qui permettent l'évaluation de la difficulté de l'instance et, en particulier, la difficulté associée à chaque véhicule. Cette information est utilisée dans les heuristiques de construction. Deux heuristiques très différentes sont proposées selon que l'objectif de premier niveau est de type *EP* ou de type *RAF*.

Les techniques de construction commencent avec la séquence formée des véhicules de la journée $J - 1$ et ajoutent ensuite à chaque itération un véhicule parmi ceux de la journée J encore disponibles. Le but est ici de construire une bonne solution dans un temps de calcul assez court. Comme la plupart des techniques de construction proposées sont des heuristiques non déterministes, celles-ci nous permettent aussi d'obtenir des solutions de départ différentes pour la recherche tabou. Ceci réduit les chances de rester coincé dans un mauvais minimum local.

3.1 Traitements des contraintes de ratio

Puisqu'un grand nombre de véhicules partagent les mêmes caractéristiques au niveau des contraintes de ratio, on peut regrouper ces véhicules dans des classes d'équivalence, dites modèles de véhicules. Soit M le nombre total de modèles et N le nombre total de véhicules. On définit alors N_j^i , $j = 1, \dots, M$ et $i = 1, \dots, N$, le nombre de véhicules du modèle j disponibles à l'itération i de l'heuristique de construction. À chaque itération i de l'algorithme de construction, on choisit parmi les modèles pour lesquels $N_j^i > 0$ puisque l'insertion de deux véhicules de même modèle est équivalente. Les séquences finales seront donc constituées de modèles de véhicules (et où l'on retrouve autant de copies du modèle que de véhicules appartenant à ce modèle). Pour construire la vraie séquence de véhicules, il suffit alors de parcourir la séquence de modèles en remplaçant chaque modèle par un véhicule approprié. Avant de choisir le véhicule, on vérifie que sa couleur n'entraîne pas un dépassement de la séquence maximale de même couleur. Si un tel dépassement est inévitable, la recherche tabou pourra le corriger par la suite.

Notation :

r	contrainte de ratio
N	nombre de véhicules
M	nombre de modèles
N_R	nombre de contraintes de ratio
Num_r	numérateur d'une contrainte de ratio r
Den_r	dénominateur d'une contrainte de ratio r
$I_{\{C\}}$	fonction indicatrice qui a la valeur 1 si la condition C est satisfaite, 0 sinon
v_{ir}	$= I_{\{la\ contrainte\ } r\ s'applique\ au\ véhicule\ } v_i\}$
m_{jr}	$= I_{\{la\ contrainte\ } r\ s'applique\ au\ modèle\ } j\}$

3.1.1 Difficulté des contraintes de ratio

Soit R_r^i le nombre de véhicules pour lesquels la contrainte de ratio r est active à l'itération i , $R_r^i = \sum_{j=i}^N v_{jr} = \sum_{j=1}^M N_j^i * m_{jr}$. Chaque contrainte de ratio r est caractérisée par le rapport théorique $pTh_r = Num_r / Den_r$. À une itération i , le vrai ratio de la séquence est la somme de tous les véhicules restants à l'itération i pour lesquels la contrainte de ratio r est active, divisée par le nombre de véhicules restants, soit $pReel_r^i = R_r^i / (N - i + 1)$.

Si $pReel_r^i > pTh_r$, il y aura nécessairement des violations au niveau de la contrainte de ratio r , sinon il peut exister une solution sans violation de la contrainte. Un premier indicateur de la difficulté de r est donc :

$$DifRatio0_r^i = pReel_r^i / pTh_r \quad (3.1)$$

On peut améliorer l'estimateur de la difficulté d'une contrainte de ratio en tenant compte des interactions entre ces contraintes. On peut tester l'indépendance entre deux contraintes de ratio r_1 et r_2 en faisant un T-test sous l'hypothèse H_0 : r_1 et r_2 sont indépendantes. Pour calculer l'indépendance entre les deux contraintes de ratio, on s'intéresse au nombre de véhicules pour lesquels la paire (v_{r_1}, v_{r_2}) est dans $Z = \{z_1, z_2, z_3, z_4\} = \{(1, 1), (1, 0), (0, 1), (0, 0)\}$.

Sous H_0 , le nombre d'occurrences pour la paire :

- $(1, 1)$ est $ThO_{z_1}^i = R_{r_1}^i * R_{r_2}^i / N$
- $(1, 0)$ est $ThO_{z_2}^i = R_{r_1}^i * (N - R_{r_2}^i) / N$
- $(0, 1)$ est $ThO_{z_3}^i = (N - R_{r_1}^i) * R_{r_2}^i / N$
- $(0, 0)$ est $ThO_{z_4}^i = (N - R_{r_1}^i) * (N - R_{r_2}^i) / N$

Le vrai nombre d'occurrences observé pour la paire :

- (1, 1) est $O_{z_1}^i = \sum_{j=1}^M (N_j^i * m_{jr_1} * m_{jr_2}) / N$
- (1, 0) est $O_{z_2}^i = \sum_{j=1}^M (N_j^i * m_{jr_1} * (1 - m_{jr_2})) / N$
- (0, 1) est $O_{z_3}^i = \sum_{j=1}^M (N_j^i * (1 - m_{jr_1}) * m_{jr_2}) / N$
- (0, 0) est $O_{z_4}^i = \sum_{j=1}^M (N_j^i * (1 - m_{jr_1}) * (1 - m_{jr_2})) / N$

La valeur $Tobs_{r_1, r_2}^i$ est calculée comme la somme des écarts au carré entre les valeurs de la distribution sous H_0 et la distribution empirique, divisée par les valeurs sous H_0 , soit $Tobs_{r_1, r_2}^i = \sum_{j=1}^4 (O_{z_j}^i - ThO_{z_j}^i)^2 / ThO_{z_j}^i$. On peut alors choisir une valeur $T_{critique}$ et rejeter H_0 si $Tobs_{r_1, r_2}^i > T_{critique}$. Avec le seuil de confiance fixé à 1%, la valeur critique pour la loi Khi^2 avec un degré de liberté est $T_{critique}^{0.01} = 6.63$. Il est par contre plus intéressant d'utiliser les valeurs normalisées

$$Dep_{r_1, r_2}^i = \frac{Tobs_{r_1, r_2}^i}{\sum_{l=1}^{N_R} \sum_{k=1}^{N_R} Tobs_{l, k}^i}$$

comme estimateurs de la dépendance entre les contraintes r_1 et r_2 à l'itération i .

Lorsque les contraintes de ratio r_1 et r_2 sont indépendantes, l'optimisation d'une contrainte ne nuit pas à l'optimisation de l'autre contrainte. D'ailleurs, on observe que les instances avec des contraintes de ratio relativement indépendantes sont plus faciles à résoudre. Donc, de grandes valeurs pour Dep_{r_1, r_2}^i conduisent à une difficulté plus élevée pour les contraintes r_1 et r_2 .

On utilise la valeur $DifRatio_0^i$ dans (3.1) comme estimateur de départ de la difficulté d'une contrainte de ratio r . À la valeur $DifRatio_0^i$, on ajoute la difficulté des autres contraintes de ratio pondérée par les estimateurs de dépendance entre les contraintes. On obtient ainsi un meilleur estimateur de la difficulté des contraintes de ratio à une itération i . Plus précisément :

$$DifRatio_1^i = DifRatio_0^i + a * \frac{\sum_{r'=1, r' \neq r}^{N_R} (Dep_{r, r'}^i * DifRatio_0^{i, r'})}{(N_R - 1)} \quad (3.2)$$

où a est une constante, $0 < a \leq 1$. En pratique, une valeur de $a = 0.5$ permet, pour des contraintes de ratio r qui ont à peu près la même difficulté $DifRatio_0^i$, de bien discriminer entre ces contraintes.

Voici un exemple de dépendance entre certaines contraintes pour l'instance 023_EP_ENP_RAF_S23_J3 [5] :

Pour les contraintes r_1 et r_4 :

Distribution sous H_0 pour r_1 et r_4

$r_1 : r_4$	0	1	Σ
0	919.5	167.5	1087
1	19.5	3.5	23
Σ	939	171	1110

Distribution empirique pour r_1 et r_4

$r_1 : r_4$	0	1	Σ
0	916	171	1087
1	23	0	23
Σ	939	171	1110

La valeur de $Tobs_{1,4}^1$ observée est $4.21 < 6.63$ et l'hypothèse H_0 n'est pas rejetée avec un seuil de confiance de 1%.

Pour la même instance, pour les contraintes r_2 et r_6 :

Distribution sous H_0 pour r_2 et r_6				Distribution empirique pour r_2 et r_6			
$r_2 : r_6$	0	1	Σ	$r_2 : r_6$	0	1	Σ
0	255.4	552.6	805	0	348	457	805
1	95.6	209.4	305	1	0	305	305
Σ	348	762	1110	Σ	348	762	1110

La valeur de $Tobs_{2,6}^1$ est $189.36 > 6.63$. L'hypothèse d'indépendance H_0 est évidemment rejetée avec un seuil de confiance de 1

3.1.2 Heuristiques de construction pour les problèmes EP_ENP et EP_RAF

Dans la littérature, plusieurs heuristiques de construction ont été développées pour résoudre le problème d'espacement correspondant aux contraintes de ratio [8].

Gottlieb et al. [10] décrivent plusieurs heuristiques gloutonnes et montrent que les versions dynamiques sont plus performantes que les versions statiques. Dans les heuristiques dynamiques, les critères de décision s'adaptent en fonction des véhicules restants à l'itération courante. Avant de présenter notre approche, nous allons d'abord introduire les concepts de coût local d'une insertion, de coût futur d'une insertion et de difficulté associée à un véhicule. À partir de ces critères de base qui permettent de choisir le véhicule à insérer à une itération i , on présentera une version stochastique ainsi qu'une version dynamique de l'heuristique de construction.

3.1.2.1 Coût local d'une insertion

Le coût local d'insertion est le plus important critère de sélection pour la plupart des heuristiques de construction [10]. Ce coût, noté Cl_j^i , représente le nombre de violations de contraintes causées localement par l'insertion du modèle j à la fin de la séquence courante à l'itération i .

Afin de calculer Cl_j^i , on définit $A_k = \sum_{l=k}^{\max(k+Den_r-1,i)} m_{j_l r} - Num_r$, $A_k I_{\{A_k > 0\}}$ représente le nombre de violations pour une contrainte r dans la séquence de véhicules de longueur Den_r qui commence à la position k . Alors $Cl_j^i = \sum_{r=1}^{N_R} \sum_{k=i-Den_r+1}^i A_k I_{\{A_k > 0\}}$ où $j_i = j$ et $j_l, l = i - Den_r + 1, \dots, i - 1$ sont les modèles de véhicules déjà introduits dans la séquence aux positions $i - Den_r + 1, \dots, i - 1$.

Si on utilise comme seul critère de décision le coût local d'insertion on obtient des solutions qui peuvent contenir beaucoup de violations de contraintes de ratio vers la fin. Quand on minimise localement, le choix optimal correspond souvent aux véhicules avec un petit nombre d'options. Vers la fin, les véhicules restants sont des véhicules avec un grand nombre d'options, d'où le grand nombre de violations qui sont introduites. Pour

prévenir ce type de comportement, Gottlieb et al. [10] choisissent à chaque itération le véhicule qui minimise le nombre de violations de contraintes tout en requérant le nombre maximum d'options. Le poids de chaque option est pondérée par les taux d'utilisation de chaque contrainte. Nous utilisons cette approche dans le concept de *difficulté d'un véhicule* mais avec une définition plus élaborée pour la difficulté des contraintes.

3.1.2.2 Difficulté des modèles de véhicules

Un modèle de véhicule avec plus de contraintes de ratio est un modèle plus difficile. Une première définition de la difficulté d'un modèle est donc

$$DifM_j^i = \sum_{r=1}^{N_R} m_{jr} \quad (3.3)$$

Le désavantage de cette définition est de donner le même poids à toutes les contraintes de ratio, alors qu'en réalité, elles sont de difficultés différentes. Puisqu'on veut favoriser l'insertion des modèles difficiles en premier, on pondère la somme précédente par l'estimateur de difficulté des contraintes de ratio $DifRatio_r^i$. On définit $DifRatio_r^i$, soit selon (3.1) ou selon (3.2).

$$DifM_j^i = \sum_{r=1}^{N_R} DifRatio_r^i * m_{jr} \quad (3.4)$$

3.1.2.3 Coût futur d'une insertion

Le coût futur d'une insertion s'intéresse à l'impact d'une décision à l'itération courante i sur les itérations subséquentes. Par exemple, il peut arriver que l'insertion d'un modèle à l'itération i permette de réduire le nombre de violations qui seront introduites dans les itérations subséquentes. On ne peut pas connaître l'impact exact de l'insertion, puisque ceci reviendrait à résoudre le problème original. Il s'agit alors d'estimer les conséquences de la décision d'insérer un modèle de véhicule j à une itération i . Pour ce faire, on se concentre sur les sous-séquences que l'on peut former avec les modèles de véhicules restants à l'itération courante i .

Soit cf_r une borne inférieure sur le nombre de violations de la contrainte r qu'on peut introduire dans les sous-séquences formées des modèles de véhicules restants, si on insère un modèle qui n'est pas soumis à la contrainte r à l'itération i .

Le coût futur correspondant au modèle j sera

$$CFM_j = \sum_{r=1}^{N_R} (1 - m_{jr}) * cf_r \quad (3.5)$$

On peut obtenir facilement une borne cf_r à partir de $DifRatio0_r^i$ en (3.1). Dans ce qui suit, on assume l'indépendance de la contrainte de ratio r avec les autres contraintes.

C'est d'ailleurs une des raisons pour lesquelles le cf_r qu'on obtient est une borne inférieure. Supposons donc qu'on insère à l'itération i un modèle avec $m_{jr} = 0$. On distingue deux cas :

- si $DifRatio0_r^{i+1} < 1$, on peut trouver au cours des itérations subséquentes un ordonnancement des modèles de véhicules restants qui ne viole pas la contrainte r , et donc on a $cf_r = 0$.
- si $DifRatio0_r^{i+1} > 1$, tout ordonnancement des modèles de véhicules restants contiendra au moins une violation de la contrainte r . La décision d'insérer un modèle avec $m_{jr} = 0$ à l'itération courante i entraînera au moins une violation supplémentaire dans les itérations subséquentes, ce qui correspond à $cf_r = 1$ (en effet, en insérant un modèle avec $m_{jr} = 0$ plus tard, on peut réduire de 1 le nombre de violations).

En conclusion, si on insère un modèle avec $m_{jr} = 0$ à l'itération i et $DifRatio0_r^{i+1} > 1$, le coût futur $cf_r = 1$, sinon $cf_r = 0$:

$$cf_r = I_{\{\{DifRatio0_r^{i+1} | \{m_{jr}=0\}\} > 1\}}$$

Le calcul de $DifRatio0_r^{i+1}$ sachant que $m_{jr} = 0$ se fait facilement à partir de $DifRatio0_r^i$,

$$\begin{aligned} DifRatio0_r^{i+1} | \{m_{jr} = 0\} &= \frac{\sum_{j=i+1}^N v_{jr}}{(N-i) * pTh_r} \\ &= \frac{N-i+1}{N-i} * \frac{\sum_{j=i}^N v_{jr}}{(N-i+1) * pTh_r} \\ &= \frac{N-i+1}{N-i} * DifRatio0_r^i \end{aligned}$$

En remplaçant $DifRatio0_r^{i+1} | \{m_{jr} = 0\}$ dans la formule de cf_r , on obtient :

$$cf_r = I_{\{\frac{N-i+1}{N-i} * DifRatio0_r^i > 1\}}$$

Le coût futur d'une insertion pour un modèle de véhicule j avec cette définition de cf_r est donc :

$$CFM_j^i = \sum_{r=1}^{N_R} (1 - m_{jr}) * I_{\{\frac{N-i+1}{N-i} * DifRatio0_r^i > 1\}} \quad (3.6)$$

3.1.3 Coût global d'insertion

Soit C_j^i la fonction de coût global pour l'insertion d'un véhicule de modèle j à l'itération i . C_j^i est défini comme une combinaison linéaire du coût local d'une insertion, du coût futur d'une insertion et de la difficulté du modèle, soit

$$C_j^i = Cl_j^i + a_1 * CFM_j^i + a_2 * DifM_j^i \quad (3.7)$$

où a_1 et a_2 sont des constantes avec $1 \geq a_1 \gg a_2 > 0$.

Les critères principaux sont le coût local et le coût futur d'une insertion. La difficulté des modèles est un critère secondaire de décision. On construit ensuite une solution en choisissant à chaque itération le véhicule avec le C_j^i le plus bas et on l'insère à la fin de la séquence. Ceci correspond à une heuristique de nature statique. On peut toutefois obtenir de meilleures solutions avec des approches de nature dynamique ou stochastique. En particulier, l'approche stochastique permet d'obtenir une grande diversité de solutions, comme on le verra plus loin.

3.1.4 Heuristique de construction dynamique pour les problèmes EP_ENP et EP_RAF

Dans l'approche dynamique, l'idée de base est de considérer plusieurs choix à une itération i parmi les modèles avec les C_j^i les plus bas.

Évidemment, au fur et à mesure que i augmente, les choix de modèles deviennent de plus en plus limités. C'est pourquoi les premières décisions sont très importantes, et ce d'autant plus que selon le modèle choisi à l'itération i , on aura des valeurs très différentes pour C_j^{i+1} , $j = 1, \dots, M$ à l'itération suivante $i + 1$. On va donc considérer un grand nombre de branchements au début de l'heuristique, même si ces derniers se révèlent assez coûteux (puisque la longueur d'une branche est en $O(N)$).

On a aussi intérêt à considérer le plus de branchements possibles pour les derniers modèles de véhicules, étant donné que les choix pour les modèles restants sont assez limités et que ceux-ci correspondent à des véhicules difficiles. Heureusement, les branchements au niveau des derniers modèles sont peu coûteux et se réalisent en $O(1)$.

Pendant la recherche, on maintient une borne supérieure qui correspond à la meilleure solution connue. On arrête alors l'exploration d'une branche dès que la coût accumulé sur cette branche dépasse la borne.

Soit $M_1^i = \sum_{j=1}^M I_{N_j^i > 0}$ le nombre de modèles de véhicules disponibles à l'itération i , j^i le nombre de branchements à la position i et M_{max}^i le nombre maximum de branchements à la position i . L'heuristique de construction dynamique peut alors se résumer ainsi :

1. $i \leftarrow 1$
2. $j^1 \leftarrow 1$
3. Tant que $i > 1$
 - (a) Calculer les $C_{jk}^i, k \in \{1, \dots, M_1^i\}$ (3.7)
 - (b) Ordonner les modèles de véhicules j_k en ordre croissant des C_{jk}^i .
 - (c) Insérer le modèle de véhicule j^i à la position i dans la solution et calculer la valeur de la solution partielle $Val(S^i)$.
 - (d) Si $Val(S^i) > Val(S^*)$ où $j^i > \min(M_{max}^i, M_1^i)$
– $i \leftarrow i - 1$

- sinon
- $j^i \leftarrow j^i + 1$
 - $j^{i+1} \leftarrow 0$
 - $i \leftarrow i + 1$
- (e) Si $i = N + 1$
- Si $Val(S^N) < Val(S^*)$
 - sauvegarder S^N
 - $Val(S^*) \leftarrow Val(S^N)$
 - $i \leftarrow i - 1$

3.1.5 Heuristique de construction stochastique pour les problèmes EP_ENP et EP_RAF

Soit $\bar{C}_j^i = \max_k(C_k^i) - C_j^i$, les gains liés à une insertion et $M_1^i = \sum_{j=1}^M I_{N_j^i > 0}$ le nombre de modèles de véhicules disponibles à l'itération i . Posons $j_1, j_2, \dots, j_{M_1^i}$ les modèles de véhicules avec $m_j^i > 0$ et $\bar{C}_{j_1}^i \geq \bar{C}_{j_2}^i \geq \dots \geq \bar{C}_{j_{M_1^i}}^i$. À une itération i , on choisit de façon stochastique un modèle de véhicule tout en favorisant les modèles avec les gains $\bar{C}_{j_k}^i$ les plus élevés. Soit p_j^i , la probabilité de choisir le modèle de véhicule j à l'itération i . On définit les p_j^i directement à partir des $\bar{C}_{j_k}^i, k \in \{1, \dots, M_1^i\}$, soit :

$$p_{j_k}^i = \frac{\bar{C}_{j_k}^i}{\sum_{l=1}^{M_1^i} (\bar{C}_l^i)}$$

Cette approche génère de bonnes solutions, mais est assez limitée au niveau de la diversification, étant donné que l'on retrouve souvent des $p_{j_k}^i$ dominants. Pour obtenir une meilleure diversification, on génère donc les $p_{j_k}^i$ avec la méthode des rangs. Soit Ra_j^i le rang du modèle j après avoir ordonné les modèles de véhicules de manière décroissante selon les \bar{C}_j^i . Les probabilités obtenues sont alors :

$$p_{j_k}^i = \frac{(Ra_{j_k}^i)^b}{\sum_{l=1}^{M_1^i} (Ra_{j_l}^i)^b}$$

où b est une constante, $b \geq 1$, qui permet d'établir un équilibre entre la diversité des solutions et leur qualité moyenne. Une augmentation de b diminue la diversité des solutions mais augmente leur qualité moyenne.

En pratique, on limite le nombre de choix qu'on considère à chaque itération selon la formule $\min(M_1^i, 5)$. On sélectionne premièrement les $\min(M_1^i, 5)$ modèles de véhicules avec les plus grands $\bar{C}_{j_k}^i$. Par la suite on combine le critère des coûts avec le critère des rangs afin d'obtenir les p_j^i . Cette approche nous donne un bon équilibre entre la diversification et la qualité des solutions. Plus précisément :

– pour $k = 1, \dots, \min(5, M_1^i)$

$$p_{jk}^i = \frac{\bar{C}_{jk}^i + Ra_{jk}^i}{\sum_{l=1}^{\min(5, M_1^i)} (\bar{C}_{jl}^i + Ra_{jl}^i)}$$

– pour $k > 5$, $p_{jk}^i = 0$.

L'heuristique de construction stochastique peut se résumer ainsi :

1. Pour $i = 1$ à N

(a) Calculer les $C_{jk}^i, k \in \{1, \dots, M_1^i\}$ (3.7)

(b) $\bar{C}_j^i = \max_k(C_k^i) - C_j^i, \forall k \in \{1, \dots, M_1^i\}$

(c) Calculer les $p_{jk}^i, k \in \{1, \dots, \min(5, M_1^i)\}$ à partir des \bar{C}_{jk}^i

(d) Sélectionner un modèle de véhicule j^* en utilisant la distribution donnée par les $p_{jk}^i, k \in \{1, \dots, \min(5, M_1^i)\}$

(e) Insérer le modèle de véhicule j^* à la position i dans la solution.

Pour obtenir une bonne solution de départ, l'heuristique de construction stochastique est redémarrée plusieurs fois et, à chaque fois, la meilleure solution est sauvegardée. On observe en pratique que les approches dynamique et stochastique convergent vers des solutions ayant des valeurs comparables.

3.2 Heuristique de construction pour le problème *RAF_EP*

Notation

N_C^i	nombre de couleurs différentes à l'itération i
NC_k^i	nombre de véhicules de couleur k à l'itération i
c_i	couleur du véhicule i
L_{max}	longueur maximale pour une sous-séquence de véhicules consécutifs de la même couleur
$\%$	opérateur modulo

Ici, il est possible de construire une solution qui minimise le nombre de purges ou changements de couleurs. Donc, ce qui nous intéresse vraiment est de construire une bonne solution de deuxième niveau tout en nous assurant de préserver l'optimalité au premier niveau.

Dans ce qui suit, nous appelons modèles de véhicules les regroupements de véhicules qui partagent la même configuration au niveau des contraintes de ratio *et* qui ont la même couleur.

3.2.1 Préserver l'optimalité au niveau du nombre de changements de couleur

A chaque itération i de l'algorithme de construction, on crée une liste avec toutes les couleurs de véhicules qui peuvent être utilisées et qui préservent l'optimalité au premier

niveau. Pour avoir le plus petit nombre de changements de couleur, l'idée de base est de regrouper les véhicules de même couleur. On est toutefois limité par la longueur maximale d'une séquence de même couleur (ou bloc) L_{max} . Ainsi, une borne inférieure pour le nombre minimum de blocs de couleur k est $NB_k^i = \lceil NC_k^i / L_{max} \rceil$.

Ces blocs doivent être séparés par des blocs de couleur différente, sinon la taille combinée de deux blocs serait d'au moins :

- $2 * L_{max}$, si $NC_k^i \% L_{max} = 0$;
- $L_{max} + NC_k^i \% L_{max}$, sinon.

et dans les deux cas cette valeur est supérieure à L_{max} .

Une couleur kd avec le plus grand NB_k^i , c'est-à-dire $kd = \operatorname{argmax}_k \{NB_k^i\}$ est appelée couleur dominante. Soit c^* la couleur du dernier bloc de couleur et n^* la longueur de ce bloc. On va analyser les conditions nécessaires d'optimalité dans les deux scénarios suivants :

- Commencer un bloc d'une couleur différente ;
- Poursuivre la construction d'un bloc de même couleur.

3.2.1.1 Commencer un bloc d'une couleur différente

On peut déterminer la valeur de la solution optimale, les couleurs k qui sont optimales pour ce nouveau bloc, ainsi que des bornes inférieures et supérieures sur la taille du nouveau bloc, soit $MinB_k$ et $MaxB_k$. Pour trouver ces valeurs, on examine toutes les possibilités :

1. Si $NB_{kd}^i + I_{\{kd=c^*\}} < \sum_{k=1, k \neq kd}^{Nc} NB_k^i + 1$, il y a une solution optimale avec $\sum_{k=1}^{Nc} NB_k^i$ blocs et $\sum_{k=1}^{Nc} NB_k^i$ changements de couleur. La couleur du nouveau bloc peut être toute couleur $k \neq c^*$. Par ailleurs, la taille minimale d'un nouveau bloc de couleur k est $MinB_k = NC_k^i \% L_{max}$ tandis que la taille maximale est $MaxB_k = \min(NC_k^i, L_{max})$.
2. Si $NB_{kd}^i + I_{\{kd=c^*\}} = \sum_{k=1, k \neq kd}^{Nc} NB_k^i + 1$ avec $kd \neq c^*$ (i.e., $NB_{kd}^i = \sum_{k=1, k \neq kd}^{Nc} NB_k^i + 1$), il y a une solution optimale avec $\sum_{k=1}^{Nc} NB_k^i$ blocs et $\sum_{k=1}^{Nc} NB_k^i$ changements de couleur. Afin d'obtenir cette solution on doit commencer avec un bloc de couleur kd et finir avec un bloc de couleur kd . La taille minimale d'un nouveau bloc de couleur kd est $MinB_{kd} = NC_{kd}^i \% L_{max}$ tandis que la taille maximale est $MaxB_{kd} = \min(NC_{kd}^i, L_{max})$.
3. Si $NB_{kd}^i + I_{\{kd=c^*\}} = \sum_{k=1, k \neq kd}^{Nc} NB_k^i + 1$ avec $kd = c^*$ (i.e., $NB_{kd}^i = \sum_{k=1, k \neq kd}^{Nc} NB_k^i$), il y a une solution optimale avec $\sum_{k=1}^{Nc} NB_k^i$ blocs et $\sum_{k=1}^{Nc} NB_k^i$ changements de couleur. Afin d'obtenir cette solution on peut commencer avec n'importe quel bloc de couleur $k \neq kd$. La taille minimale d'un nouveau bloc de couleur k est $MinB_k = NC_k^i \% L_{max}$ tandis que la taille maximale est $MaxB_k = \min(NC_k^i, L_{max})$.
4. Si $NB_{kd}^i + I_{\{kd=c^*\}} > \sum_{k=1, k \neq kd}^{Nc} NB_k^i + 1$, on ne peut pas séparer tous les blocs de la couleur dominante kd avec des blocs de couleur $k \neq kd$. Il faut alors créer

$NB_{kd}^i + I_{\{kd=c^*\}} - 1$ blocs de couleur $k \neq kd$ afin de revenir au cas précédent. La solution optimale aura donc $2 * NB_{kd}^i + I_{\{kd=c^*\}} - 1$ changements de couleur. Pour arriver à cela, il faut toutefois avoir au moins $Nb_{kd}^i + I_{\{kd=c^*\}} - 1$ véhicules de couleur $k \neq kd$, c'est-à-dire

$$\sum_{k=1, k \neq kd}^{N_C} NC_k \geq NB_{kd}^i + I_{\{kd=c^*\}} - 1 \quad (3.8)$$

sinon il n'y a pas de solution réalisable.

- (a) Si $kd \neq c^*$, on doit commencer avec un bloc de couleur kd et finir avec un bloc de couleur kd pour obtenir une solution optimale. La taille minimale d'un nouveau bloc de couleur kd est $MinB_{kd} = NC_{kd}^i \% L_{max}$ tandis que la taille maximale est $MaxB_{kd} = \min(NC_{kd}^i, L_{max})$.
- (b) Si $kd = c^*$ on peut commencer avec n'importe quelle couleur $k \neq kd$ afin d'obtenir une solution optimale. Un seul représentant de cette couleur est suffisant pour former le nouveau bloc et donc $MinB_k = 1$. Pour la taille maximale il faut s'assurer qu'après avoir construit le bloc de couleur k , la contrainte (3.8) est respectée :

$$\sum_{l=1, l \neq kd}^{N_C} NC_l - MaxB_k \geq NB_{kd}^i - 1$$

Donc, la taille maximale est

$$MaxB_k = \min(NC_{kd}^i, L_{max}, \sum_{l=1, l \neq kd}^{N_C} NC_l - NB_{kd}^i + 1).$$

3.2.1.2 Poursuivre la construction d'un bloc de même couleur

Si la longueur courante du bloc n^* est plus petite que $MinB_c^*$ il faut ajouter un modèle de couleur c^* . Si la longueur courante du bloc $MinB_{c^*} \leq n^* \leq MaxB_{c^*}$

- On peut ajouter un modèle de couleur c^* ;
- On peut arrêter la construction du bloc courant et utiliser l'analyse de la section 3.2.1.1 afin d'identifier les choix de couleurs optimales.

Si la longueur courante du bloc $n^* = MaxB_{c^*}$, on doit arrêter la construction du bloc courant et utiliser l'analyse de la Section 3.2.1.1 afin d'identifier les choix de couleurs optimales.

Dans la suite, nous présentons l'heuristique de construction complète.

3.2.1.3 Pseudo code

1. $n^* = 0, c^* = \text{couleur de fin de la journée } J - 1$.
2. Utiliser l'analyse de la Section 3.2.1.1 afin de trouver les choix de couleurs optimales au départ.
entrée : $(n^*, c^*, \{NC_k\}_{k \in \{1, \dots, N_C\}}) \rightarrow$ sortie : $\{k, MinB_k, MaxB_k\}_{\{k \text{ couleur optimale}\}}$

3. Utiliser l'heuristique stochastique de la Section 3.1.5 afin d'insérer un modèle parmi les modèles de véhicules qui ont une couleur optimale.
entrée : $\{m_j | \text{couleur du modèle } m_j \text{ optimale}\} \rightarrow$ sortie : k^* la couleur du modèle choisi.
4. $c^* \leftarrow k^*$
 $n^* = 1$
 $MinB_{c^*} \leftarrow MinB_{k^*}$
 $MaxB_{c^*} \leftarrow MaxB_{k^*}$
 $NC_{k^*} = NC_{k^*} - 1$
5. pour $i = 2$ à N
 - (a) Utiliser l'analyse de la section 3.2.1.2 afin de trouver les choix de couleurs optimales.
entrée : $(n^*, c^*, \{NC_k\}_{\{1, \dots, N_C\}}) \rightarrow$ sortie : $\{k, MinB_k, MaxB_k\}_{\{k \text{ couleur optimale}\}}$
 - (b) Utiliser l'heuristique stochastique de la section 3.1.5 afin d'insérer un modèle parmi les modèles de véhicules qui ont une couleur optimale.
entrée : $\{m_j | \text{couleur de } m_j \text{ optimale}\} \rightarrow$ sortie : k^* la couleur du modèle choisi.
 - (c) si $c^* = k^*$ alors
– $n^* = n^* + 1$
sinon
– $n^* = 1$
– $c^* \leftarrow k^*$
– $MinB_{c^*} \leftarrow MinB_{k^*}$
– $MaxB_{c^*} \leftarrow MaxB_{k^*}$
 $NC_{k^*} = NC_{k^*} - 1$

3.3 Améliorations de 2^{ème} ou 3^{ème} niveau

Au début, un grand nombre de modifications sont faites sur la solution de départ pour optimiser l'objectif de premier niveau ou les objectifs de premier et deuxième niveaux. Au cours de ces modifications, les autres objectifs sont complètement négligés. En conséquence, la solution est plutôt mauvaise par rapport à ces derniers. Pour arriver rapidement à de bonnes solutions pour les objectifs de niveaux inférieurs, on utilise une heuristique d'amélioration par échange où les critères de sélection des véhicules impliqués dans un échange sont semblables à ceux utilisés dans les heuristiques de construction (c'est pourquoi cette section fait partie du chapitre sur les heuristiques de construction). Cependant, on s'oblige ici à préserver la valeur des objectifs aux niveaux supérieurs.

Le concept d'échanges de véhicules est présenté en détail dans les chapitres 4 et 5 sur la recherche tabou. Nous le résumons ici.

Soit la solution courante S de la forme $\{v_1, \dots, v_{i-1}, v_i, v_{i+1}, \dots, v_{j-1}, v_j, v_{j+1}, \dots, v_N\}$, alors un échange entre les véhicules v_i et v_j est défini comme :

$$S \oplus \text{échange}(i, j) = \{v_1, \dots, v_{i-1}, v_j, v_{i+1}, \dots, v_{j-1}, v_i, v_{j+1}, \dots, v_N\}.$$

L'heuristique d'amélioration parcourt la séquence de la première à la dernière position à la recherche d'échanges de véhicules profitables. Lorsque la fin de la séquence est atteinte, la procédure s'arrête.

Heuristique d'amélioration au 2^{ème} (3^{ème}) niveau

1. Pour $i = 1$ à N

- (a) Créer une liste $L^i = \{v_j | \text{Val}(S) = \text{Val}(S \oplus \text{échange}(i, j))\}$ selon les objectifs de 1^{er} ou (1^{er} et 2^{ème}) niveaux
- (b) Utiliser les mêmes critères de sélection que dans les heuristiques de construction mais appliquées au 2^{ème} (3^{ème}) niveau :
 - Dans le cas où le 2^{ème} (3^{ème}) niveau correspond à des contraintes de ratio, utiliser l'approche de la Section 3.1.5 afin d'obtenir une liste L_1^i avec les probabilités de sélection associées à chaque modèle de véhicule de 2^{ème} (3^{ème}) niveau.
 - Dans le cas où le 2^{ème} (3^{ème}) niveau correspond à minimiser le nombre de changements de couleurs, utiliser l'approche de la section 3.2 afin de générer une liste L_1^i avec des choix de couleurs optimales.
- (c) Sélectionner un véhicule candidat v_{j^*} parmi les véhicules de L^i selon les préférences qu'on retrouve dans la liste L_1^i :
 - Dans le cas où le 2^{ème} (3^{ème}) niveau correspond à des contraintes de ratio, choisir un véhicule v_{j^*} de façon aléatoire parmi les véhicules de L^i selon les probabilités qu'on retrouve dans la liste L_1^i .
 - Dans le cas où le 2^{ème} (3^{ème}) niveau correspond à minimiser le nombre de changements de couleur :
 - si $\{\text{couleur de } v_j | v_j \in L^i\} \cap L_1^i = \emptyset$, v_{j^*} sera automatiquement v_i
 - sinon, les véhicules de la liste $\{v_j | v_j \in L^i \text{ et couleur de } v_i \in L_1^i\}$ ont la même probabilité d'être choisis.
- (d) $S \leftarrow S \oplus \text{échange}(i, j^*)$

CHAPITRE 4

LA RECHERCHE TABOU

Dans ce chapitre, nous décrivons la recherche tabou que nous avons développée pour résoudre le problème d'ordonnancement de véhicules. Nous allons présenter plusieurs variantes de l'algorithme de recherche tabou, chacune ayant une efficacité plus grande selon son contexte d'utilisation. Nous allons montrer les avantages des éléments stochastiques pour la diversification et pour éviter des comportements cycliques. Enfin, nous allons présenter les paramètres de la recherche tabou qui permettent de gérer le niveau des perturbations et la profondeur de la recherche.

4.1 Description générale

La recherche tabou est une méta-heuristique introduite par Glover [9] en 1986. Il s'agit d'un algorithme intelligent de recherche locale auquel on peut ajouter une mémoire adaptative. Son plus grand atout est la capacité d'explorer efficacement l'espace des solutions tout en évitant le piège des minima locaux.

La recherche tabou est largement utilisée pour résoudre des problèmes fortement combinatoires et ayant un grand nombre de variables. La recherche tabou peut être utilisée conjointement avec d'autres techniques d'optimisation, telles que la programmation linéaire ou des heuristiques spécialisées. Ainsi, la technique peut être adaptée à une large gamme de problèmes et domaines (théorie des graphes, planification, ...).

La recherche tabou est une procédure itérative où, à chaque itération, le voisinage de la solution courante est examiné. La meilleure solution dans ce voisinage est alors choisie comme nouvelle solution courante, ce qui constitue une itération de la méthode. On espère, bien sûr, que la meilleure solution dans le voisinage de la solution courante sera de meilleure qualité que la solution courante, mais on choisit toujours cette meilleure solution, même si elle est de moins bonne qualité. En répétant cette procédure de manière itérative, on s'attend à obtenir une solution assez proche de la solution optimale. En effet, en permettant de détériorer la solution courante, la recherche tabou peut s'échapper des minima locaux.

La description générale de la recherche tabou qui est présentée dans ce qui suit repose sur les travaux de Michel Gendreau et Jean-Yves Potvin [6].

4.1.1 L'espace de recherche

Notation

- X l'espace des solutions admissibles
- M l'ensemble des modifications applicables à une solution
- F la fonction d'évaluation de la valeur d'une solution S

À chaque itération, on passe d'une certaine solution $S \in X$ à une solution $S' \in X$ en apportant une modification m à S (notation $S' = S \oplus m$). Soit $M_S \subseteq M$ l'ensemble des modifications possibles qu'on peut appliquer à S , c'est-à-dire $M_S = \{m \in M \mid S \oplus m \in X\}$.

À l'aide de M_S on peut définir le voisinage de S :

$$N_S = \{S' \mid m \in M_S \text{ et } S' = S \oplus m\}.$$

4.1.2 Listes tabou

Pour s'échapper des minima locaux, il faut pouvoir accepter des modifications m tels que $S' = S \oplus m$ et $F(S') > F(S)$, où S est la solution courante. Ceci entraîne souvent le danger de cycler si $S' \in N(S)$ et $S \in N(S')$.

Afin d'éviter de cycler au cours de la recherche, l'idée fondamentale qui donne son nom à cet algorithme est de maintenir une liste tabou $T = \{\text{sous-ensemble des solutions déjà visitées}\}$ où T a une cardinalité finie. Il est alors interdit de revenir à une solution qui se trouve dans la liste tabou. La nouvelle solution S' qui est choisie à l'itération courante prend toujours la place de la plus ancienne solution dans la liste T . Ainsi, puisque la taille de la liste tabou est finie, il y a toujours une possibilité de cycler si la longueur du cycle excède la taille de la liste.

4.1.3 Niveaux d'aspiration

Selon la façon d'implémenter la liste tabou, la recherche peut être plus ou moins permissive. Nous pouvons ainsi lever le statut tabou d'une certaine solution $S' = S \oplus m$ et de la modification m qui a permis d'atteindre cette solution si, par exemple, $F(S \oplus m) < F(S^*)$ ou $F(S \oplus m) = F(S^*)$ et $S \oplus m \neq S^*$, S^* étant la meilleure solution connue.

4.1.4 Critères d'arrêt

Les critères d'arrêt classiques de la recherche tabou sont :

- Si on ne peut améliorer la meilleure solution rencontrée S^* après $It_{max}^{S^*}$ itérations.
- Si un nombre absolu d'itérations It_{max} est réalisé.
- Si $F(S) = F^*$, où F^* est une borne inférieure pour F .
- Si le voisinage $N(S) - T = \emptyset$.

4.1.5 Intensification et diversification

Deux concepts vont revenir souvent dans la discussion sur la recherche tabou : l'intensification de la recherche et la diversification de la recherche.

L'intensification correspond à une exploration plus intense d'un espace des solutions. Si une région semble contenir de bonnes solutions, on veut alors intensifier la recherche dans cette région. L'intensification se fait généralement en augmentant la taille du voisinage exploré.

La diversification consiste à explorer de nouvelles régions dans l'espace des solutions. La diversification peut se faire soit par :

- la génération d'une nouvelle solution de départ ;
- une modification de la fonction objectif de façon à rendre les régions non explorées plus intéressantes ;
- une modification des paramètres de la recherche tabou, tels que la longueur de la liste tabou ou la façon de générer le voisinage.

L'algorithme de recherche tabou peut se résumer ainsi :

Notation.

S	solution courante
S^*	meilleure solution connue
It	nombre total d'itérations faites par l'algorithme
It_{max}	nombre maximal d'itérations alloué à l'algorithme
It^{S^*}	nombre d'itérations afin d'obtenir S^*
$It_{max}^{S^*}$	nombre d'itérations maximal pour l'amélioration de S^*
F	fonction d'évaluation d'une solution S

Algorithme classique :

1. Générer une solution initiale $S \in X$.
2. $S^* \leftarrow S$.
3. $It^{S^*} = 0$.
4. $It = 0$.
5. Tant que $It \leq It_{max}$ et $It - It^{S^*} \leq It_{max}^{S^*}$ et $N(S) - T \neq \emptyset$ faire
 - (a) $It = It + 1$.
 - (b) Générer $V \subseteq N(S) - T$
 - (c) Déterminer S' tel que $F(S') = \min_{\bar{S} \in V} F(\bar{S})$
 - (d) $S \leftarrow S'$
 - (e) Mettre à jour T
 - (f) Si $F(S') < F(S^*)$ alors
 - $S^* \leftarrow S$
 - $It^{S^*} = It$

4.2 Implémentation de la recherche tabou pour l'ordonnement de véhicules

Cette section décrit différentes adaptations de la recherche tabou que j'ai conçues dans le cadre du Challenge ROADEF'05.

4.2.1 Initialisation

Dans l'algorithme générique de la section 2.1, nous appelons à plusieurs reprises la recherche tabou. Nous avons d'abord intérêt à descendre rapidement après la boucle 3(a) où de courtes recherches tabou sont réalisées. La meilleure solution rencontrée est donc fournie comme solution de départ pour la recherche tabou de l'étape 3(b). Une fois que la meilleure solution rencontrée devient suffisamment bonne, le risque de rester coincé dans un bassin assez profond autour de cette solution est important. Pour toutes les étapes subséquentes, on entame donc la recherche tabou à partir de la solution courante obtenue à la fin de l'étape précédente.

4.2.2 L'espace de recherche

Souvent, l'espace de recherche X est étendu à plus que l'ensemble des solutions réalisables en relaxant certaines contraintes et en les pénalisant dans la fonction objectif. Dans notre cas, on a même relaxé la contrainte dure liée à la longueur maximale d'une séquence de véhicules de même couleur. Ceci permet un traitement plus facile de cette contrainte, une meilleure exploration de l'espace des solutions et la possibilité de commencer la recherche tabou avec des solutions non-réalisables. Soit :

$X =$ toutes les séquences de véhicules pour la journée J .

Nous résolvons une relaxation du problème initial proposé, en l'occurrence :

Minimiser $_{\{S \in X\}} F(S)$, où

$$\begin{aligned} F(S) = & M_0 * (\text{Nombre de dépassements de la contrainte dure}) + \\ & + M_1 * (\text{Nombre de violations EP}) + \\ & + M_2 * (\text{Nombre de violations ENP}) + \\ & + M_3 * (\text{Nombre de changements de couleur}) \end{aligned}$$

Ici, M_0 est une constante beaucoup plus grande que M_1 , M_2 et M_3 . Par ailleurs, ces derniers sont égaux à 10^6 , 10^3 ou 1, selon que l'objectif correspondant est au premier, deuxième ou troisième niveau, respectivement. Ces valeurs ont été proposées pendant le Challenge ROADEF'05 pour hiérarchiser les objectifs. Nous avons donc les problèmes EP_ENP avec $M_1 = 10^6$, $M_2 = 10^3$ et $M_3 = 1$; EP_RAF avec $M_1 = 10^6$, $M_2 = 1$ et $M_3 = 10^3$; RAF avec $M_1 = 10^3$, $M_2 = 1$ et $M_3 = 10^6$.

4.2.3 Voisinages

Plusieurs voisinages ont été proposés dans la littérature pour le problème d'ordonnement de véhicules [10], soit échanges, extractions\insertions, inversions des séquences, changements aléatoires. Pendant le challenge ROADEF'05, des échanges de blocs de même couleur ont aussi été utilisés [15]. Nous avons d'ailleurs nous-mêmes implémenté des échanges de blocs de même couleur, mais l'utilisation de cette approche est assez limitée et ne concerne que les instances de type *RAF* [2].

Puisque nous devons parcourir un espace des solutions fortement combinatoire, il faut une méthode d'exploration efficace. Nous avons donc choisi d'implémenter des voisinages par échange et par extraction\insertion. En exploitant des pré-calculs, ces deux voisinages permettent une exploration très rapide de l'espace de recherche. Nous présentons maintenant ces voisinages. Les pré-calculs afférents sont décrits au Chapitre 5.

4.2.4 Voisinage par échange

Le voisinage par échange consiste à obtenir de nouvelles solutions à partir de la solution courante en échangeant les positions de deux véhicules.

Soit la solution courante S de la forme $\{v_1, \dots, v_{i-1}, v_i, v_{i+1}, \dots, v_{j-1}, v_j, v_{j+1}, \dots, v_N\}$, alors un échange entre les véhicules v_i et v_j est défini comme étant :

$$S \oplus \text{échange}(i, j) = \{v_1, \dots, v_{i-1}, v_j, v_{i+1}, \dots, v_{j-1}, v_i, v_{j+1}, \dots, v_N\}.$$

Le voisinage de la solution S suite à un échange est :

$$N(S) = \{S' | S' = S \oplus \text{échange}(i, j), \forall (i, j) \text{ tel que } i, j \in [1, N], \\ \text{Tabou}_{v_i} = 0, \text{Tabou}_{v_j} = 0 \text{ et } i \neq j\}$$

En pratique, on retrouve souvent des véhicules qui appartiennent à la même classe d'équivalence, dans le sens qu'ils possèdent les mêmes contraintes actives, ce que nous notons par $v_i = v_j$. Dans ce cas, on dit que $\text{échange}(i, j)$ est un échange nul. Les échanges nuls ne nous intéressent pas puisqu'ils ne changent pas la valeur de la solution. Donc on peut restreindre la définition de $N(S)$ de la façon suivante :

$$N(S) = \{S' | S' = S \oplus \text{échange}(i, j), \forall (i, j) \text{ tel que } i, j \in [1, N], \\ \text{Tabou}_{v_i} = 0, \text{Tabou}_{v_j} = 0 \text{ et } v_i \neq v_j\}$$

En pratique, ceci réduit de façon considérable la cardinalité de $N(S)$, mais ne change pas la complexité.

4.2.5 Voisinage par extraction\insertion

Le voisinage par extraction\insertion consiste à obtenir de nouvelles solutions à partir de la solution courante en extrayant un véhicule et en le réinsérant entre deux autres véhicules dans la séquence.

Soit $S \oplus ExtIns(i, j)$ l'opérateur qui correspond à l'extraction du véhicule v_i et sa réinsertion entre les véhicules v_{j-1} et v_j .

$$S \oplus ExtIns(i, j) = \{v'_1, \dots, v'_N\}, \text{ où :}$$

- Si $i < j$, alors
 - $v'_l = v_l, \forall l, l < i$,
 - $v'_l = v_{l+1}, \forall l, i \leq l < j - 1$,
 - $v'_{j-1} = v_i$,
 - $v'_l = v_l, \forall l, l > j - 1$.
- Si $j < i$, alors
 - $v'_l = v_l, \forall l, l < j$,
 - $v'_j = v_i$,
 - $v'_l = v_{l-1}, \forall l, j < l \leq i$,
 - $v'_l = v_l, \forall l, l < i$.

$$N(S) = \{S' | S' = S \oplus ExtIns(i, j), \forall (i, j) \text{ tel que } i, j \in [1, N], Tabou_{v_i} = 0 \text{ et } i \neq j\}$$

4.2.6 Le choix des candidats en vue d'un mouvement

À une itération de la recherche tabou, le voisinage complet de la solution S est constitué du voisinage par échange et du voisinage par extraction\insertion. Ces voisinages sont dans l'ordre de $O(N^2)$, où N est le nombre de véhicules de la journée J . Pour la plupart des instances, N^2 est autour de 10^6 , ce qui est assez grand même si nous arrivons à faire l'évaluation de ces mouvements en $O(1)$. En plus, seulement un petit pourcentage de ces mouvements sont vraiment intéressants. Alors, pour ne pas évaluer autant d'échanges à chaque itération, une sélection en deux étapes est faite parmi les véhicules qui peuvent participer aux mouvements.

Une première sélection stochastique L_{rand} est faite parmi les véhicules admissibles (non tabou). Ceci réduit non seulement le voisinage, mais réduit aussi le risque de cycler. En effet, si une solution S est visitée plusieurs fois, il y a de fortes chances qu'on explore à chaque fois d'autres solutions à partir de S , grâce à cette sélection stochastique.

Une autre idée qui améliore l'efficacité est de sélectionner les véhicules qui sont responsables du plus grand nombre de violations de contraintes ou pénalités. Ce sont effectivement les véhicules qui causent des problèmes et on espère bien faire mieux en les déplaçant à d'autres positions. C'est pourquoi une deuxième sélection L_{Sel} est faite à partir de la liste L_{rand} , de façon à favoriser les véhicules qui entraînent le plus grand nombre de pénalités. Cette dernière liste L_{Sel} est implémentée comme un monceau.

Cette approche fonctionne bien, mais a le désavantage de sélectionner trop souvent les mêmes véhicules, qui sont souvent associés à des violations inévitables, même dans la solution optimale, et qui sont donc peu pertinents. En plus, des améliorations peuvent aussi être obtenues en déplaçant des véhicules qui ne causent pas de pénalités. Par exemple, si un véhicule n'a pas une contrainte de ratio r active et ne cause pas de pénalités, il pourrait quand même permettre d'espacer deux véhicules qui ont une contrainte de ratio r active.

On utilise donc plusieurs critères de sélection pour former la liste L_{Sel} . En l'occurrence, le critère de sélection basé sur l'ordre *décroissant* du nombre de pénalités est choisi avec une probabilité p_1 , le critère de sélection basé sur l'ordre *croissant* du nombre de pénalités est choisi avec une probabilité p_2 et, enfin, le critère de sélection *aléatoire* est choisi avec une probabilité p_3 . Pour une meilleure efficacité, p_1 a toujours une valeur plus grande que $p_2 + p_3$.

Le voisinage réduit de la solution S par rapport à L_{Sel} est :

- $N(S, L_{Sel}) = \{S' | S' = S \oplus \text{échange}(i, j), \forall (i, j) \text{ avec } v_i \in L_{Sel}, \text{Tabou}_{v_j} = 0 \text{ et } v_i \neq v_j\}$ dans le cas des échanges ;
- $N(S, L_{Sel}) = \{S' | S' = S \oplus \text{ExtIns}(i, j), \forall (i, j) \text{ avec } v_i \in L_{Sel} \text{ et } i \neq j\}$ dans le cas des extraction\insertions.

D'autres approches pour faire une telle réduction de $N(S)$ se retrouvent en [10], où la sélection est aléatoire ou selon des critères apparentés à ceux utilisés par les algorithmes gloutons. Pendant le challenge ROADEF, plusieurs équipes ont tenu compte du nombre de violations de contraintes afin de sélectionner les véhicules candidats pour un mouvement [3].

4.2.7 Listes tabou

En pratique, on ne garde pas des solutions complètes dans la liste tabou. Dans notre cas, les véhicules qui ont récemment participé à un mouvement sont déclarés tabou. Après un nombre d'itérations It_{Tabou} les véhicules tabou redeviennent disponibles pour un mouvement.

Une variable $Tabou_v$, associée à chaque véhicule v , gère le nombre d'itérations pendant lesquelles le véhicule v est tabou :

- si $Tabou_v > 0$ le véhicule v est tabou ;
- sinon le véhicule v est disponible pour un mouvement.

Les véhicules tabou sont gardés dans une liste tabou L_{Tabou} pour améliorer la gestion des variables $Tabou_v$.

Une fois qu'un véhicule v participe à un mouvement :

- $Tabou_v \leftarrow It_{Tabou}$;
- v est ajouté à L_{Tabou} .

A chaque itération, la valeur de $Tabou_v$ est décrémentée de 1 pour tous les véhicules dans la liste tabou L_{Tabou} . Quand la valeur de $Tabou_v$ tombe à 0, le véhicule v est éliminé de L_{Tabou} .

4.2.8 Niveau d'aspiration

Soit S la solution courante. Plusieurs techniques sont utilisées pour rendre la recherche plus agressive, comme les niveaux d'aspiration.

Le statut tabou d'un certain véhicule v est levé si, suite à une modification m qui implique v , on obtient :

- $F(S \oplus m) < F(S^*)$;
- $F(S \oplus m) = F(S^*)$ et $S \neq S^*$ et $Tabou_v < It_{Tabou}/a$, où a est une constante, $a > 1$.

Une deuxième approche consiste à réduire temporairement It_{Tabou} . Ceci accélère l'amélioration de la solution courante S .

4.2.9 Critère d'arrêt

L'efficacité de la recherche tabou est donnée par le nombre d'itérations réalisées depuis l'obtention de la meilleure solution It^{S^*} . Si ce nombre dépasse un seuil $It_{max}^{S^*}$ la recherche est arrêtée. Il faut noter que la valeur $It_{max}^{S^*}$ n'est pas fixée. Elle change de façon à ce que la recherche s'arrête plus rapidement si l'efficacité marginale est petite et se prolonge si l'efficacité marginale est grande.

Au départ, $It_{max}^{S^*}$ a une valeur assez grande. Ensuite, chaque fois qu'une meilleure solution S^* est trouvée, $It_{max}^{S^*}$ est multipliée par un coefficient prenant des valeurs entre 0.5 et 1.5 selon l'efficacité marginale de la recherche. Plus précisément :

$$It_{max}^{S^*} = (1 + \ln(1.6 - It^{S^*}/It_{max}^{S^*}))It_{max}^{S^*}$$

En posant $Ef_m = 1 - It^{S^*}/It_{max}^{S^*}$ comme estimateur de l'efficacité marginale, on obtient

$$It_{max}^{S^*} = (1 + \ln(0.6 + Ef_m))It_{max}^{S^*}$$

Puisque l'efficacité marginale Ef_m est une valeur comprise entre 0 et 1, le coefficient $(1 + \ln(1.6 - It^{S^*}/It_{max}^{S^*}))$ a effectivement une valeur comprise entre 0.5 et 1.5. Ainsi, lorsque Ef_m est près de 0, ce qui correspond à une efficacité marginale élevée, le coefficient a une valeur proche de 1.5 et on augmente le seuil $It_{max}^{S^*}$. À l'opposé, lorsque Ef_m est près de 1, ce qui correspond à une efficacité marginale faible, le coefficient a une valeur proche de 0.5 et on diminue le seuil $It_{max}^{S^*}$ car on ne veut pas faire trop d'itérations dans une région peu prometteuse.

Pour pouvoir respecter les contraintes sur le temps total d'optimisation, la recherche tabou est arrêtée si elle atteint le temps maximum qui lui est alloué. Le nombre maximum d'itérations It_{max} permet de mieux gérer la répartition du temps de calcul entre la recherche tabou courante, les recherches tabou à venir avec des paramètres différents ou d'autres méthodes d'optimisation.

4.2.10 Intensification et diversification

De façon générale, lorsqu'une meilleure solution S^* est trouvée (possiblement éloignée du voisinage de la meilleure solution précédente), il y a de fortes chances qu'on puisse encore l'améliorer grâce à de l'intensification. En effet, la solution courante n'est pas seulement améliorée mais aussi détériorée pendant la recherche. À long terme, plusieurs détériorations locales peuvent contribuer à obtenir une meilleure solution. D'autres détériorations peuvent, par contre, être inutiles à long terme. Une meilleure solution S^* peut donc contenir quelques détériorations inutiles. L'intensification de la recherche dans son voisinage peut permettre de remédier à ces détériorations.

Voici les techniques que nous utilisons pour intensifier la recherche.

- Lorsqu'une meilleure solution est trouvée, une descente locale est utilisée. Cette recherche auxiliaire évalue à chaque itération le voisinage complet de la solution courante et non un sous-ensemble du voisinage tel qu'on le fait avec la recherche tabou. Dans la descente locale, on n'accepte que les mouvements qui améliorent la solution et on s'arrête lorsqu'on ne peut plus améliorer la solution (ou après un nombre fixe d'itérations).
- Les paramètres de la recherche tabou sont modifiées. En particulier, la taille du voisinage est augmentée.

Maintenant, voici les techniques que nous utilisons pour diversifier la recherche :

- La recherche tabou est appliquée à partir de solutions de départ différentes.
- Les paramètres de la recherche tabou sont modifiées. En particulier, la taille du voisinage est réduite.

D'une façon générale, la recherche tabou que nous utilisons a la structure suivante :

Notations.

v_i	véhicule qui se trouve à la position i dans la séquence des véhicules de la journée courante
$Tabou_{v_i}$	nombre d'itérations pendant lesquelles le véhicule v_i est tabou
N	nombre de véhicules de la journée courante
S	solution courante, S a la forme $\{v_0, v_1, \dots, v_{N-1}\}$
S^*	meilleure solution connue
It	nombre total d'itérations faites par l'algorithme
It_{max}	nombre maximum d'itérations alloué à l'algorithme
It^{S^*}	nombre d'itérations depuis la dernière amélioration de S^*
T	temps utilisé par la recherche tabou
T_{max}	temps maximum alloué à l'algorithme
m	mouvement défini selon le voisinage
$S \oplus m$	solution S modifiée par le mouvement m
$argmin_x \{f(x)\}$	fonction qui retourne un x qui minimise $f(x)$
F	fonction d'évaluation d'un ordonnancement S
$\Delta(S', S)$	différence entre la valeur de la solution S' et celle de la solution S , $\Delta(S', S) = F(S') - F(S)$
L_{Rand}	liste de véhicules sélectionnés d'une façon aléatoire
L_{Sel}	liste de véhicules sélectionnés en vue d'un mouvement
$ L $	cardinalité d'un ensemble L
$N(S, L)$	voisinage de S en appliquant des mouvements utilisant au moins un véhicule de L

Début

1. Commencer avec une solution initiale S (S peut ne pas être la meilleure solution connue S^*). On reçoit comme paramètre la valeur de $\Delta(S, S^*)$, qui nous permet de ne jamais calculer la différence entre la valeur des deux solutions, mais seulement les différences dues aux mouvements, afin de détecter si on améliore S^* .
2. $It \leftarrow 0$
3. $It^{S^*} \leftarrow 0$
4. Tant que $It^{S^*} < It_{max}^*$, $It < It_{max}$ et $T < T_{max}$ faire :
 - (a) Pour tout $v_i \in L_{Tabou}$
 - $Tabou(v_i) = Tabou(v_i) - 1$
 - Si ($Tabou(v_i) = 0$) alors $L_{Tabou} = L_{Tabou} - v_i$
 - (b) Générer d'une façon aléatoire un critère de sélection choisi parmi *croissant*, *décroissant* ou *aléatoire*.

- (c) Générer d'une façon aléatoire $L_{Rand} \subseteq \{v_i | Tabou(v_i) = 0\}$
- (d) Créer $L_{Sel} \subset L_{Rand}$ en sélectionnant les véhicules de L_{Rand} selon le nombre total d'erreurs causées par les véhicules et le critère de sélection choisi en 4b.
- (e) Trouver $m^* = argmin_m \{\Delta(S, S \oplus m) | S \oplus m = S' \in N(S, L_{Sel})\}$
- (f) Le ou les véhicules qui participent au mouvement m^* deviennent tabou pour un nombre It_{Tabou} itérations et sont ajoutés à la liste L_{Tabou}
- (g) Si $\Delta(S', S) > 0$, alors :
 - $\Delta(S', S^*) \leftarrow \Delta(S', S) + \Delta(S, S^*)$
 - sinon
 - Si $\Delta(S', S) + \Delta(S, S^*) < 0$, alors :
 - $S^* \leftarrow S'$
 - $\Delta(S, S^*) \leftarrow 0$
 - $It_{max}^{S^*} = (1 + \ln(1.6 - It^{S^*} / It_{max}^{S^*})) It_{max}^{S^*}$
 - $It^{S^*} \leftarrow 0$
 - Utiliser une recherche locale pour un nombre fixe d'itérations afin d'intensifier la recherche dans le voisinage du nouveau S^*
 - sinon :
 - $\Delta(S', S^*) \leftarrow \Delta(S', S) + \Delta(S, S^*)$
- (h) $S \leftarrow S'$
- (i) $It \leftarrow It + 1$
- (j) $It^{S^*} \leftarrow It^{S^*} + 1$

Fin

4.3 La complexité de la recherche tabou

La complexité d'une itération de la recherche tabou correspond à parcourir une fois la boucle 4. Les étapes avec des complexités non constantes dans cette boucle sont :

- étape 4a :

$O(|L_{Tabou}|)$ ce qui est équivalent à $O(|It_{Tabou}|)$ puisqu'à chaque itération, le même nombre de véhicules est ajouté et enlevé à L_{Tabou} et un véhicule passe It_{Tabou} itérations dans la liste L_{Tabou} . $O(|It_{Tabou}|) < O(N)$.

- étape 4c :

$O(|L_{Rand}|) < O(N)$

– étape 4d :

$$O(|L_{Rand}| * \log(|L_{Sel}|)) < O(N * \log(N))$$

– étape 4e :

$$O(|N(S, L_{Sel})| * O(\Delta(S \oplus m, S))) = O(|L_{Sel}| * (N - |L_{Tabou}|)) * O(\Delta(S \oplus m, S)) < O(N^2) * O(\Delta(S \oplus m, S)), \text{ où } O(\Delta(S \oplus m, S)) \text{ dépend du voisinage.}$$

– étape 4g :

Si une meilleure solution est trouvée, elle est sauvegardée et une recherche locale commence, donc la complexité est $O(N) + O(\text{descente locale})$. Sinon, l'étape se fait en $O(1)$. Soit $P(\Delta(S', S^*) < 0)$ la probabilité de trouver une meilleure solution, la complexité de l'étape sera :

$$P(\Delta(S', S^*) < 0) * [O(N) + O(\text{descente locale})] + (1 - P(\text{amelioration}))O(1) \approx \approx P(\Delta(S', S^*) < 0) * [O(N) + O(\text{descente locale})]$$

– étape 4h :

$$O(S \leftarrow S \oplus m^*) \text{ dépend du voisinage.}$$

En sommant seulement sur les complexités qui sont dominantes, la complexité d'une itération est :

$$O(\text{iteration}) = O(|L_{Sel}| * (N - |L_{Tabou}|)) * O(\Delta(S \oplus m, S)) + O(S \leftarrow S \oplus m^*) + P(\Delta(S', S^*) < 0) * [O(N) + O(\text{Recherche locale})]$$

La recherche locale se fait pendant un nombre constant et réduit d'itérations, pendant lesquelles l'exploration se fait sur un voisinage plus grand $\approx O(N^2)$. Nous pouvons donc estimer sa complexité à $O(N^2) * O(\Delta(S \oplus m, S)) + O(S \leftarrow S \oplus m^*)$.

Donc la complexité d'une itération se réécrit comme :

$$O(\text{iteration}) = O(|L_{Sel}| * (N - |L_{Tabou}|) + P(\Delta(S \oplus m, S^*) < 0) * N^2) * O(\Delta(S \oplus m, S)) + O(S \leftarrow S \oplus m^*)$$

(4.1)

4.4 Techniques avancées

4.4.1 Bornes sur les valeurs des échanges

Une façon de mieux contrôler le niveau de perturbation de la solution courante S est d'imposer des bornes sur les coûts des mouvements. Ainsi, à l'étape 4e, on ne va accepter un mouvement m que si $\Delta(S \oplus m, S) \leq Borne_{inf}$ ou si $\Delta(S \oplus m, S) \geq Borne_{sup}$.

Lorsque la solution S ne peut pas être améliorée dans son voisinage, c'est-à-dire que $\Delta(S'', S) \geq 0 \forall S'' \in N(S, L_{Sel})$, le critère de sélection $m^* = \operatorname{argmin}_m \{\Delta(S \oplus m, S) | S \oplus m \in N(S, L_{Sel})\}$ nous guide vers des échanges de valeur $\Delta(S'', S) = 0$, s'il en existe. Si ces types d'échanges nous gardent longtemps dans des voisinages sans améliorations, nous donnons une valeur strictement négative à $Borne_{inf}$ et une valeur strictement positive à $Borne_{sup}$.

D'une façon plus générale, au début de la recherche nous essayons de faire de grandes améliorations tout en évitant d'explorer en profondeur un voisinage. Dans ce but, un haut seuil est imposé pour les détériorations ainsi que pour les améliorations à la solution courante. Donc, les bornes $Borne_{inf}$ et $Borne_{sup}$ reçoivent au début de grandes valeurs qui seront réduites avec le temps. Ceci accélère beaucoup la convergence.

Une autre utilité de ces bornes est d'imposer un niveau pour l'objectif. Si les bornes sont dans l'ordre de 10^6 , les améliorations ainsi que les détériorations vont impliquer l'objectif de premier niveau ; si l'ordre de grandeur est de 10^3 , les améliorations et détériorations vont impliquer principalement l'objectif de deuxième niveau, et possiblement de premier niveau, et ainsi de suite.

4.4.2 Niveaux

Dans la liste des véhicules sélectionnés L_{Sel} qui sont candidats pour un mouvement, le critère de base est le nombre de pénalités causées par un véhicule. Ceci n'est pas toujours pratique une fois qu'on a déjà trouvé une solution optimale pour l'objectif de premier niveau.

L'exemple le plus simple est le cas où le nombre de changements de couleur correspond à l'objectif de premier niveau. Dans ce cas, on peut trouver directement une solution optimale de premier niveau avec l'algorithme de construction. Comme la solution optimale contiendra toujours des changements de couleur, la valeur optimale du premier objectif ne sera jamais 0. De plus, les véhicules qui sont à l'extrémité d'une sous-séquence de même couleur auront une pénalité dans l'ordre de 10^6 , étant donné qu'ils sont responsables d'au moins un changement de couleur. Cette valeur n'est pas du tout pertinente pour l'objectif de deuxième niveau.

Donc, quand l'optimum est atteint au premier niveau, la recherche tabou doit être concentrée au deuxième niveau. Pour cela, lors de la création de la liste L_{Sel} , les véhicules sont choisis selon les pénalités associées aux deuxième et troisième niveaux seulement.

4.4.2.1 Diversification par des poids individuels pour les contraintes de ratio

Soit c_r le nombre de violations pour une contrainte de ratio r et $Poid_r$ le poids relatif de la contrainte r par rapport aux autres contraintes de ratio. Dans notre fonction objectif, chaque violation d'une contrainte de ratio prioritaire a le même poids par rapport aux autres $Poid_r = 1, r \in 1, \dots, N_R$. Afin de diversifier, nous perturbons notre objectif en affectant des poids individuels aux différentes contraintes de ratio. Il s'agit d'une diversification bien ciblée, car les poids des contraintes de ratio avec le plus grand nombre de violations sont augmentés :

$$Poid_r = 1 + a * \frac{c_r}{\sum_{k=1}^{N_R} c_k}, r \in 1, \dots, N_R,$$

où a est une constante dont la valeur est proche de 2.

Cette approche est intéressante seulement si un grand pourcentage de violations sont causées par un nombre réduit de contraintes de ratio. La recherche tabou utilise les nouveaux poids dans l'objectif pendant un nombre fixe d'itérations. Une fois ce nombre d'itérations accompli, la meilleure solution obtenue devient notre nouvelle solution de départ et on revient alors à l'objectif original. Cette approche entraîne une migration des violations des contraintes de ratio avec de grands c_r vers celles avec des c_r moins grands. Il faut noter que la valeur de l'objectif original peut être détériorée dans le processus.

La structure très différente de cette nouvelle solution de départ, ainsi que l'aspect stochastique de la recherche, nous assure d'éviter de cycler une fois que la recherche tabou revient à l'objectif original. Nous avons pu observer sur quelques instances que cette approche permet de passer de minima locaux très profonds jusqu'à la solution optimale.

CHAPITRE 5

VOISINAGES UTILISÉS DANS LA RECHERCHE TABOU

Dans ce chapitre, nous décrivons les voisinages utilisés dans le cadre de la recherche tabou. Il s'agit du voisinage par *échange* et du voisinage par *extraction\insertion*. Au niveau de l'implémentation, il faut souligner l'importance des pré-calculs qui permettent une évaluation très rapide de ces voisinages, tel qu'il sera expliqué dans la suite.

5.1 Introduction

D'une façon générale, à une itération de la recherche tabou, nous évaluons le voisinage de la solution actuelle, choisissons un mouvement intéressant parmi les mouvements possibles, effectuons le mouvement choisi afin de produire une nouvelle solution et sauvegardons cette solution s'il s'agit de la meilleure solution connue.

La complexité de la recherche tabou provient des opérations liées aux mouvements permettant de produire de nouvelles solutions dans le voisinage et à l'évaluation de ces nouvelles solutions.

Nous avons montré au chapitre précédent que la complexité d'une itération de la recherche tabou est au pire de :

$$O(\text{iteration}) = O(N^2) * O(\Delta(S \oplus m, S)) + O(S \leftarrow S \oplus m^*)$$

Donc, la complexité minimale qui peut être espérée à chaque itération est de $O(N^2)$. Cette complexité est obtenue en faisant l'évaluation d'un mouvement en $O(1)$ et le passage de S à $S \oplus m$ en moins de $O(N^2)$.

Afin d'avoir une complexité de $O(1)$ pour l'évaluation d'un mouvement, nous allons définir des variables pré-calculées dont les valeurs seront disponibles avant l'évaluation d'un voisinage. Ainsi, le passage de S à $S \oplus m$ consiste non-seulement à faire le mouvement m mais aussi à ajuster toutes les valeurs pré-calculées qui sont affectées par le mouvement m .

Notre approche dans la suite sera la suivante. Pour chacun des deux voisinages, nous commençons avec les contraintes de ratio. Pour ces dernières, nous présentons les choses dans l'ordre suivant :

- calculs permettant d'obtenir $\Delta(S \oplus m, S)$,
- définition des valeurs pré-calculées à partir de $\Delta(S \oplus m, S)$,
- ajustement des valeurs pré-calculées suite à un mouvement.

Cet ordre de présentation est ensuite repris pour les contraintes touchant aux purges de couleur et à la longueur maximale d'une séquence de véhicules de même couleur.

5.2 Voisinage par échange

Le voisinage par échange consiste à obtenir de nouvelles solutions à partir de la solution courante en échangeant les positions de deux véhicules qui n'appartiennent pas à la même classe d'équivalence. Si S est la solution courante de la forme

$$\{v_1, \dots, v_{i-1}, v_i, v_{i+1}, \dots, v_{j-1}, v_j, v_{j+1}, \dots, v_N\}$$

alors un échange entre les véhicules v_i et v_j est défini comme :

$$S \oplus \text{échange}(i, j) = \{v_1, \dots, v_{i-1}, v_j, v_{i+1}, \dots, v_{j-1}, v_i, v_{j+1}, \dots, v_N\}.$$

Le voisinage obtenu est donc :

$$N(S) = \{S' | S' = S \oplus \text{échange}(i, j), \forall (i, j) \text{ tel que } i, j \in [1, N] \text{ et } v_i \neq v_j\}.$$

5.2.1 Le coût d'un échange(i, j) pour les contraintes de ratio

Un coût est associé à chaque échange selon la position des véhicules et leurs voisins dans la séquence. Nous allons montrer que les N^2 échanges peuvent être calculés en $N^2 * O(\text{No. contraintes de ratio})$ opérations, en utilisant seulement $N * \text{No. contraintes de ratio}$ valeurs pré-calculées.

Notation

r	contrainte de ratio
Num_r	numérateur d'une contrainte de ratio r
Den_r	dénominateur d'une contrainte de ratio r
$I_{\{C\}}$	fonction indicatrice, qui a la valeur 1 si la condition C est satisfaite, 0 sinon
v_{ir}	$= I_{\{si \text{ la contrainte } r \text{ s'applique au véhicule } v_i\}}$

Nous allons considérer un échange entre les véhicules v_i et v_j avec $i < j$. Pour commencer, l'analyse est réduite à une seule contrainte de ratio r .

Soit N le nombre total de véhicules de la journée $J - 1$ et J . Par définition, les véhicules de la journée $J + 1$ sont inconnus et on peut prendre $v_{lr} = 0$ pour tout $l > N$. Cela permet de généraliser les formules du calcul des pénalités pour la fin de la journée J .

La valeur d'une solution S par rapport à la contrainte r est

$$Val_r(S) = \sum_{k=1}^N \left(\sum_{l=k}^{k+Den_r-1} v_{lr} - Num_r \right) I_{\{\sum_{l=k}^{k+Den_r-1} v_{lr} - Num_r > 0\}} \quad (5.1)$$

Soit $S' = S \oplus \text{échange}(i, j)$. Ce qui nous intéresse pour la recherche tabou est $\Delta_r(S', S) = Val_r(S') - Val_r(S)$.

5.2.1.1 L'évaluation de $\Delta_r(S', S)$ sous l'hypothèse $i \leq j - Den_r$

Notons :

$$A_k = \sum_{l=k}^{k+Den_r-1} v_{lr} - Num_r$$

$$A_{ik} = \sum_{l=k, l \neq i}^{k+Den_r-1} v_{lr} - Num_r.$$

Sous l'hypothèse $i \leq j - Den_r$, la somme correspondant à $Val_r(S)$ en (5.1), peut être décomposée ainsi :

$$\begin{aligned} Val_r(S) = & \sum_{k=1}^{i-Den_r} A_k I_{\{A_k > 0\}} + \sum_{k=i-Den_r+1}^i A_k I_{\{A_k > 0\}} + \sum_{k=i+1}^{j-Den_r} A_k I_{\{A_k > 0\}} + \\ & \sum_{k=j-Den_r+1}^j A_k I_{\{A_k > 0\}} + \sum_{k=j+1}^N A_k I_{\{A_k > 0\}} \end{aligned} \quad (5.2)$$

Notons que le troisième terme de la somme peut n'avoir aucun élément si $i = j - Den_r$. On va maintenant mettre en évidence v_{ir} (ou v_{jr}), en posant $A_k = A_{ik} + v_{ir}$:

$$\begin{aligned} Val_r(S) = & \sum_{k=1}^{i-Den_r} A_k I_{\{A_k > 0\}} + \sum_{k=i-Den_r+1}^i (v_{ir} + A_{ik}) I_{\{v_{ir} + A_{ik} > 0\}} + \\ & + \sum_{k=i+1}^{j-Den_r} A_k I_{\{A_k > 0\}} + \sum_{k=j-Den_r+1}^j (v_{jr} + A_{jk}) I_{\{v_{jr} + A_{jk} > 0\}} + \\ & + \sum_{k=j+1}^N A_k I_{\{A_k > 0\}} \end{aligned} \quad (5.3)$$

En S' , le véhicule v_i se trouve à la position j , v_j à la position i et tous les autres véhicules sont à la même position. Ainsi l'équation (5.3) s'écrit ainsi pour S' :

$$\begin{aligned} Val_r(S') = & \sum_{k=1}^{i-Den_r} A_k I_{\{A_k > 0\}} + \sum_{k=i-Den_r+1}^i (v_{jr} + A_{ik}) I_{\{v_{jr} + A_{ik} > 0\}} + \\ & + \sum_{k=i+1}^{j-Den_r} A_k I_{\{A_k > 0\}} + \sum_{k=j-Den_r+1}^j (v_{ir} + A_{jk}) I_{\{v_{ir} + A_{jk} > 0\}} + \\ & + \sum_{k=j+1}^N A_k I_{\{A_k > 0\}} \end{aligned}$$

(5.4)

Comme les premier, troisième et cinquième termes de droite dans les équations (5.3) et (5.4) sont identiques, ils disparaissent dans le calcul de $Val_r(S') - Val_r(S)$ et nous obtenons :

$$\begin{aligned} \Delta_r(S', S) = & Val_r(S') - Val_r(S) = \\ & \sum_{k=i-Den_r+1}^i (v_{ir} + A_{ik}) I_{\{v_{ir}+A_{ik}>0\}} + \sum_{k=j-Den_r+1}^j (v_{jr} + A_{jk}) I_{\{v_{jr}+A_{jk}>0\}} - \\ & - \sum_{k=i-Den_r+1}^i (v_{jr} + A_{ik}) I_{\{v_{jr}+A_{ik}>0\}} - \sum_{k=j-Den_r+1}^j (v_{ir} + A_{jk}) I_{\{v_{ir}+A_{jk}>0\}} \end{aligned} \quad (5.5)$$

Dans (5.5), les sommations sur les mêmes valeurs de l'indice k peuvent être regroupées et alors :

$$\begin{aligned} \Delta_r(S', S) = & Val_r(S') - Val_r(S) = \\ & \sum_{k=i-Den_r+1}^i [(v_{ir} + A_{ik}) I_{\{v_{ir}+A_{ik}>0\}} - (v_{jr} + A_{ik}) I_{\{v_{jr}+A_{ik}>0\}}] + \\ & + \sum_{k=j-Den_r+1}^j [(v_{jr} + A_{jk}) I_{\{v_{jr}+A_{jk}>0\}} - (v_{ir} + A_{jk}) I_{\{v_{ir}+A_{jk}>0\}}] \end{aligned} \quad (5.6)$$

5.2.1.2 Valeurs pré-calculées sous l'hypothèse $i \leq j - Den_r$

On peut utiliser comme valeurs pré-calculées :

$$U_{ir} = \sum_{k=i-Den_r+1}^i [(v_{ir} + A_{ik}) I_{\{v_{ir}+A_{ik}>0\}} - (v_{jr} + A_{ik}) I_{\{v_{jr}+A_{ik}>0\}}] \quad (5.7)$$

Le calcul de $\Delta_r(S, S')$ en (5.6) devient tout simplement $\Delta_r(S, S') = U_{ir} + U_{jr}$. On va maintenant montrer qu'on peut rendre la valeur de U_{ir} indépendante de l'indice j . Premièrement, on observe que U_{ir} ne dépend de l'indice j que par la valeur de v_{jr} . De plus, pour tout véhicule j , v_{jr} est une variable binaire puisqu'un véhicule v_j ne peut qu'être ou ne pas être concerné par la contrainte r . Alors $(v_{ir}, v_{jr}) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ et nous pouvons analyser la valeur de U_{ir} dans chacun des cas.

– Si $(v_{ir}, v_{jr}) \in \{(0, 0), (1, 1)\}$, alors $v_{ir} = v_{jr} = a$ et

$$U_{ir} = \sum_{k=i-Den_r+1}^i [(a + A_{ik})I_{\{a+A_{ik}>0\}} - (a + A_{ik})I_{\{a+A_{ik}>0\}}] = 0$$

– Si $(v_{ir}, v_{jr}) = (0, 1)$, alors

$$U_{ir} = \sum_{k=i-Den_r+1}^i [-I_{\{1+A_k>0\}}] \quad (5.8)$$

– Si $(v_{ir}, v_{jr}) = (1, 0)$, alors

$$U_{ir} = \sum_{k=i-Den_r+1}^i [I_{\{A_k>0\}}] \quad (5.9)$$

Preuve pour $(v_{ir}, v_{jr}) = (0, 1)$

$$U_{ir} = \sum_{k=i-Den_r+1}^i [(A_{ik})I_{\{A_{ik}>0\}} - (1 + A_{ik})I_{\{1+A_{ik}>0\}}] \text{ et}$$

– si $A_{ik} > 0, A_{ik} + 1 > 0$, alors $(A_{ik})I_{\{A_{ik}>0\}} - (1 + A_{ik})I_{\{1+A_{ik}>0\}} = -1$;

– si $A_{ik} = 0, A_{ik} + 1 > 0$, alors $(A_{ik})I_{\{A_{ik}>0\}} - (1 + A_{ik})I_{\{1+A_{ik}>0\}} = -1$;

– si $A_{ik} < 0, A_{ik} + 1 < 0$, alors $(A_{ik})I_{\{A_{ik}>0\}} - (1 + A_{ik})I_{\{1+A_{ik}>0\}} = 0$

donc

$$\begin{aligned} U_{ir} &= \sum_{k=i-Den_r+1}^i [-I_{\{1+A_{ik}>0\}}] = \\ &= \sum_{k=i-Den_r+1}^i [-I_{\{1+v_{ir}-v_{ir}+\sum_{l=k,l \neq i}^{k+Den_r-1} v_{lr}-Num_r>0\}}] = \\ &= \sum_{k=i-Den_r+1}^i [-I_{\{1+A_k>0\}}] \end{aligned}$$

Preuve pour $(v_{ir}, v_{jr}) = (1, 0)$

$$U_{ir} = \sum_{k=i-Den_r+1}^i [(1 + A_{ik})I_{\{1+A_{ik}>0\}} - (A_{ik})I_{\{A_{ik}>0\}}]$$

On utilise un résultat obtenu dans le cas $(0, 1)$, soit :

$$\sum_{k=i-Den_r+1}^i [(A_{ik})I_{\{A_{ik}>0\}} - (1+A_{ik})I_{\{1+A_{ik}>0\}}] = \sum_{k=i-Den_r+1}^i [-I_{\{1+A_k>0\}}].$$

Alors :

$$\begin{aligned} U_{ir} &= \sum_{k=i-Den_r+1}^i [(1+A_{ik})I_{\{1+A_{ik}>0\}} - (A_{ik})I_{\{A_{ik}>0\}}] = \\ &= - \sum_{k=i-Den_r+1}^i [(A_{ik})I_{\{A_{ik}>0\}} - (1+A_{ik})I_{\{1+A_{ik}>0\}}] = \\ &= - \sum_{k=i-Den_r+1}^i [-I_{\{1+A_k\}}] = \\ &= \sum_{k=i-Den_r+1}^i [I_{\{A_k>0\}}] \end{aligned}$$

q.e.d.

Les cas (0,0) et (1,1) ne nous intéressent pas puisque $\Delta_r(S', S) = 0$.

Si v_{ir} est 0, alors le seul cas qui nous intéresse est (0,1) où

$$U_{ir} = \sum_{k=i-Den_r+1}^i [-I_{\{1+A_k>0\}}].$$

Si v_{ir} est 1, alors le seul cas qui nous intéresse est (1,0) où

$$U_{ir} = \sum_{k=i-Den_r+1}^i [I_{\{A_k>0\}}].$$

Il y a autant de valeurs pré-calculées U_{ir} que de véhicules v_i et, avec ces valeurs, le calcul de $\Delta_r(S \oplus \text{échange}(i, j), S)$ se fait en $O(1)$.

5.2.1.3 La mise à jour des valeurs pré-calculées après un échange

Une fois l'échange fait, il faut mettre à jour les valeurs pré-calculées U_{ir} pour tout véhicule entre les positions $[i - Den_r + 1, i + Den_r - 1]$ et $[j - Den_r + 1, j + Den_r - 1]$. Les autres véhicules ne subissent aucun changement dans leur voisinage et ne sont pas concernés par l'échange du véhicule v_i avec le véhicule v_j .

La mise à jour d'une séquence de longueur L peut se faire en $O(L + Den_r)$. Alors, les calculs nécessaires pour la mise à jour des valeurs pré-calculées entre les positions $[i - Den_r + 1, i + Den_r - 1]$ et $[j - Den_r + 1, j + Den_r - 1]$ se font en

$$O([(i - Den_r + 1) - (i + Den_r - 1) + Den_r] + [(j - Den_r - 1) - (j - Den_r + 1) + Den_r]) = O(Den_r).$$

Preuve

Pour obtenir une complexité de $O(L + Den_r)$ pour la mise à jour d'une séquence de longueur L , des calculs sont réalisés lors du passage de la position i à la position $i + 1$. Soit :

$$U_{ir} = \sum_{k=i-Den_r+1}^i I_{\{1+A_k>0\}}.$$

Supposons qu'on connaît, une fois qu'on est à la position i :

$$- A_k = \sum_{l=k}^{k+Den_r-1} v_{lr} - Num_r \text{ pour tout } k \in [i - Den_r + 1, i]$$

$$- Tmp_i = \sum_{k=i-Den_r+1}^i I_{\{1+A_k>0\}}$$

Le passage à la position $i + 1$ se fait de la façon suivante :

$$- A_{i+1} = A_i + v_{(i+1)r} - v_{(i-Den_r+1)r}$$

$$- Tmp_{i+1} = Tmp_i - I_{\{1+A_{i-Den_r+1}>0\}} + I_{\{1+A_{i+1}>0\}}$$

Avec ces calculs on obtient $U_{(i+1)r}$ et les valeurs nécessaires pour calculer de la même manière $U_{(i+2)r}, \dots$. Le passage de i à $i + 1$ se fait en $O(1)$ ce qui nous assure $O(L)$ pour une sous-séquence de longueur L .

Pour le premier i traité, les calculs nécessaires pour avoir les A_k pour tout $k \in [i - Den_r + 1, i]$ se font en $O(Den_r)$. Un premier calcul de la somme A_{i-Den_r+1} est en $O(Den_r)$. Ensuite, pour tout $k \in [i - Den_r + 1, i - 1]$, on utilise la récurrence $A_{k+1} = A_k + v_{(k+1)r} - v_{(k-Den_r+1)r}$ ce qui nous donne A_k pour tout $k \in [i - Den_r + 2, i]$ en $O(Den_r)$.
q.e.d.

5.2.1.4 L'évaluation de $\Delta_r(S', S)$ sous l'hypothèse $i > j - Den_r$

Nous voulons calculer $\Delta_r(S', S)$ sous l'hypothèse $i > j - Den_r$. Sous cette hypothèse, la somme correspondant à $Val_r(S)$ en (5.1) peut être décomposée ainsi :

$$\begin{aligned} Val_r(S) &= \sum_{k=1}^{i-Den_r} A_k I_{\{A_k>0\}} + \sum_{k=i-Den_r+1}^{j-Den_r} A_k I_{\{A_k>0\}} + \\ &+ \sum_{k=j-Den_r+1}^i A_k I_{\{A_k>0\}} + \sum_{k=i+1}^j A_k I_{\{A_k>0\}} + \\ &+ \sum_{k=j+1}^N A_k I_{\{A_k>0\}} \end{aligned}$$

(5.10)

Nous pouvons mettre en évidence v_{ir} (ou v_{jr}) en posant $A_k = A_{ik} + v_{ir}$:

$$\begin{aligned}
Val_r(S) &= \sum_{k=1}^{i-Den_r} A_k I_{\{A_k\}} + \sum_{k=i-Den_r+1}^{j-Den_r} (v_{ir} + A_{ik}) I_{\{v_{ir}+A_{ik}>0\}} + \\
&+ \sum_{k=j-Den_r+1}^i (v_{ir} + v_{jr} + \sum_{l=k, l \neq i, j}^{k+Den_r-1} v_{lr} - Num_r) I_{\{v_{ir}+v_{jr}+\sum_{l=k, l \neq i, j}^{k+Den_r-1} v_{lr}-Num_r>0\}} + \\
&+ \sum_{k=i+1}^j (v_{jr} + A_{jk}) I_{\{v_{jr}+A_{jk}>0\}} + \sum_{k=j+1}^N A_k I_{\{A_k>0\}}
\end{aligned} \tag{5.11}$$

En S' , le véhicule v_i se trouve à la position j , v_j à la position i et tous les autres véhicules sont à la même position. Alors, l'équation (5.11) s'écrit ainsi pour S' :

$$\begin{aligned}
Val_r(S') &= \sum_{k=1}^{i-Den_r} A_k I_{\{A_k>0\}} + \sum_{k=i-Den_r+1}^{j-Den_r} (v_{jr} + A_{ik}) I_{\{v_{jr}+A_{ik}>0\}} + \\
&+ \sum_{k=j-Den_r+1}^i (v_{jr} + v_{ir} + \sum_{l=k, l \neq i, j}^{k+Den_r-1} v_{lr} - Num_r) I_{\{v_{jr}+v_{ir}+\sum_{l=k, l \neq i, j}^{k+Den_r-1} v_{lr}-Num_r>0\}} + \\
&+ \sum_{k=i+1}^j (v_{ir} + A_{jk}) I_{\{v_{ir}+A_{jk}>0\}} + \sum_{k=j+1}^N A_k I_{\{A_k>0\}}
\end{aligned} \tag{5.12}$$

Comme les premier, troisième et cinquième termes de droite dans les équations (5.11) et (5.12) sont identiques, ils disparaissent dans le calcul de $Val_r(S') - Val_r(S)$, et on obtient :

$$\begin{aligned}
\Delta_r(S', S) &= Val_r(S') - Val_r(S) = \\
&\sum_{k=i-Den_r+1}^{j-Den_r} (v_{ir} + A_{ik}) I_{\{v_{ir}+A_{ik}>0\}} + \sum_{k=i+1}^j (v_{jr} + A_{jk}) I_{\{v_{jr}+A_{jk}>0\}} - \\
&- \sum_{k=i-Den_r+1}^{j-Den_r} (v_{jr} + A_{ik}) I_{\{v_{jr}+A_{ik}>0\}} - \sum_{k=i+1}^j (v_{ir} + A_{jk}) I_{\{v_{ir}+A_{jk}>0\}}
\end{aligned} \tag{5.13}$$

Dans (5.13), les sommations sur les mêmes valeurs de l'indice k sont regroupées et :

$$\begin{aligned} \Delta_r(S', S) = & Val_r(S') - Val_r(S) = \\ & \sum_{k=i-Den_r+1}^{j-Den_r} [(v_{ir} + A_{ik})I_{\{v_{ir}+A_{ik}>0\}} - (v_{jr} + A_{ik})I_{\{v_{jr}+A_{ik}>0\}}] + \\ & + \sum_{k=i+1}^j [(v_{jr} + A_{jk})I_{\{v_{jr}+A_{jk}>0\}} - (v_{ir} + A_{jk})I_{\{v_{ir}+A_{jk}>0\}}] \end{aligned} \quad (5.14)$$

Comme les indices des sommes dépendent à la fois de i et j , il nous faut plusieurs valeurs pré-calculées. Puisque $i < j$ et $j < i + Den_r$ le nombre de valeurs pré-calculées par véhicule est de $Den_r - 1$. En pratique, le nombre d'évaluations où $i > j - Den_r$ est assez réduit. Ainsi, pour éviter d'avoir un trop grand nombre de valeurs pré-calculées à mettre à jour, nous procédons au calcul en (5.14) quand $i > j - Den_r$. Nous utilisons donc les U_{ir} comme valeurs pré-calculées seulement dans le cas $i \leq j - Den_r$.

5.2.2 Le coût d'un échange(i, j) pour le nombre de purges de peinture

Notation

N	nombre total de véhicules
N_0	début de la journée J
c_i	couleur du véhicule i
L_{max}	longueur maximale pour une sous-séquence de véhicules consécutifs de même couleur
no_i^g	nombre de véhicules consécutifs avec la même couleur situés à gauche de v_i
no_i^d	nombre de véhicules consécutifs avec la même couleur situés à droite de v_i

On s'intéresse ici à la fois au nombre de purges de peinture et à la contrainte forte de longueur maximale d'une séquence de même couleur. Soit $\Delta_{Depas}(S', S)$, la différence entre le nombre de dépassements la longueur maximale d'une séquence de même couleur en S' et S , et soit $\Delta_{Purges}(S', S)$ la différence entre le nombre de purges de peinture en S' et S . Nous avons :

$$\Delta_{Couleur}(S', S) = M * \Delta_{Depas}(S', S) + \Delta_{Purges}(S', S)$$

$S' = S \oplus \text{échange}(i, j)$ est $\{c'_i\}_{i=1, \dots, N}$, où :

- si $i < j - 1$
 - $c'_k = c_k \forall k \neq i, j$
 - $c'_i = c_j$
 - $c'_j = c_i$
- si $i = j - 1$
 - $c'_k = c_k \forall k \neq i, j$
 - $c'_i = c'_{j-1} = c_j$
 - $c'_j = c'_{i+1} = c_i$

L'objectif lié au nombre de purges est facile à évaluer. Il y a une purge de peinture chaque fois qu'on change de couleur ce qui correspond à $\sum_{k=N_0}^N I_{\{c_k \neq c_{k-1}\}}$. Lorsqu'on fait un échange, il suffit de vérifier une courte suite de scénarios pour obtenir Δ_{Purges} , où les seules informations nécessaires sont $c_{i-1}, c_i, c_{i+1}, c_{j-1}, c_j, c_{j+1}$.

5.2.2.1 L'évaluation de $\Delta_{Purges}(S', S)$, le nombre de purges causées par un échange

Si $c_i = c_j$, alors il n'y a aucun changement $\Delta_{Purges}(S', S) = 0$.

Si $c_i \neq c_j$ et $i < j - 1$, alors

$$\Delta_{Purges}(S', S) = \sum_{k=N_0}^N I_{c'_k \neq c'_{k-1}} - \sum_{k=N_0}^N I_{c_k \neq c_{k-1}} =$$

$$I_{c_{i-1} \neq c_j} + I_{c_j \neq c_{i+1}} + I_{c_{j-1} \neq c_i} + I_{c_i \neq c_{j+1}} - I_{c_{i-1} \neq c_i} - I_{c_i \neq c_{i+1}} - I_{c_{j-1} \neq c_j} - I_{c_j \neq c_{j+1}}$$

Si $c_i \neq c_j$ et $i = j - 1$, alors $c'_{i+1} = c_i$ et $c'_{j-1} = c_j$ et on obtient

$$\Delta_{Purges}(S', S) = \sum_{k=N_0}^N I_{c'_k \neq c'_{k-1}} - \sum_{k=N_0}^N I_{c_k \neq c_{k-1}} =$$

$$I_{c_{i-1} \neq c_j} + I_{c_i \neq c_{j+1}} - I_{c_{i-1} \neq c_i} - I_{c_j \neq c_{j+1}}$$

5.2.2.2 L'évaluation de $\Delta_{Depas}(S', S)$, les dépassements de la longueur maximale d'une séquence de véhicules de même couleur causés par un échange

Soit S_k une séquence maximale de véhicules de même couleur. La valeur de S au niveau des dépassements de longueur maximale est la somme :

$$\sum_{\forall S_k} I_{\{|S_k| > L_{max}\}}$$

Nous aimerions avoir l'évaluation d'un échange en $O(1)$. Pour ce faire, il nous faut des informations sur les voisinages immédiats $Voisinage_i$ et $Voisinage_j$ des véhicules i et j . Ces voisinages correspondent aux deux séquences de véhicules consécutifs de

même couleur qui se trouvent immédiatement à droite et à gauche des positions i et j , respectivement.

Dans ce but, nous introduisons les valeurs pré-calculées no_i^g , no_i^d , où no_i^g représente le nombre de véhicules consécutifs de même couleur situés à gauche de v_i et no_i^d le nombre de véhicules consécutifs de même couleur situés à droite de v_i . La couleur du groupe qui est à gauche d'une position i est donnée par c_{i-1} et à droite par c_{i+1} .

Pour calculer la contribution d'un véhicule i au nombre de dépassements de la longueur maximale, nous devons ensuite distinguer quatre cas.

- Si $c_i = c_{i-1}$ et $c_i = c_{i+1}$, alors le véhicule i est au milieu d'une sous-séquence S_{k_i} , l'objectif dans le voisinage de v_i est $I_{\{|S_{k_i}|>L_{max}\}} = I_{\{no_i^g+1+no_i^d>L_{max}\}}$.
- Si $c_i = c_{i-1}$ et $c_i \neq c_{i+1}$, alors le véhicule i est à l'extrémité droite d'une sous-séquence S_{k_i} et a une sous-séquence $S_{k_{i+1}}$ de couleur c_{i+1} à droite ; l'objectif dans le voisinage de v_i est

$$I_{\{|S_{k_i}|>L_{max}\}} + I_{\{|S_{k_{i+1}}|>L_{max}\}} = I_{\{no_i^g+1>L_{max}\}} + I_{\{no_i^d>L_{max}\}}$$

- Si $c_i \neq c_{i-1}$ et $c_i = c_{i+1}$ alors le véhicule i est à l'extrémité gauche d'une sous-séquence S_{k_i} et a une sous-séquence $S_{k_{i-1}}$ de couleur c_{i-1} à gauche ; l'objectif dans le voisinage de v_i est

$$I_{\{|S_{k_{i-1}}|>L_{max}\}} + I_{\{|S_{k_i}|>L_{max}\}} = I_{\{no_i^g>L_{max}\}} + I_{\{no_i^d+1>L_{max}\}}$$

- Si $c_i \neq c_{i-1}$ et $c_i \neq c_{i+1}$ alors le véhicule i est en soi une sous-séquence S_{k_i} , a une sous-séquence $S_{k_{i-1}}$ de couleur c_{i-1} à gauche et une sous-séquence $S_{k_{i+1}}$ de couleur c_{i+1} à droite ; l'objectif dans le voisinage de v_i est

$$\begin{aligned} I_{\{|S_{k_{i-1}}|>L_{max}\}} + I_{\{|S_{k_i}|>L_{max}\}} + I_{\{|S_{k_{i+1}}|>L_{max}\}} &= \\ I_{\{no_i^g>L_{max}\}} + I_{\{1>L_{max}\}} + I_{\{no_i^d>L_{max}\}} &= \\ I_{\{no_i^g>L_{max}\}} + I_{\{no_i^d>L_{max}\}} & \end{aligned}$$

Dans les quatre cas, v_i ne peut appartenir à une sous-séquence S_k en dehors de son voisinage immédiat (i.e., son influence est limitée à ce voisinage). Les valeurs pré-calculées no_i^g , no_i^d , no_j^g , no_j^d et les couleurs de v_{i-1} , v_i , v_{i+1} , v_{j-1} , v_j , v_{j+1} sont donc les informations nécessaires et suffisantes pour calculer $\Delta_{Depas}(S', S)$.

Il faut noter que $\Delta_{Depas}(S', S) = 0$ si $c_i = c_j$. Dans la suite, nous allons donc supposer $c_i \neq c_j$. Avec $c_i \neq c_j$ nous traitons deux cas distincts dans le calcul de $\Delta_{Depas}(S', S)$, soit :

- a) $i \notin Voisinage_j$ et $j \notin Voisinage_i$
- b) $i \in Voisinage_j$ ou $j \in Voisinage_i$

a) L'évaluation de $\Delta_{Depas}(S', S)$ si $i \notin Voisinage_j$ et $j \notin Voisinage_i$

A partir des quatre cas possibles pour une position k , on définit $Depas_k(S, c)$ le nombre de violations de la contrainte de longueur maximale dans le voisinage de la

position k si la couleur du véhicule v_k est c .

$$\begin{aligned} Depas_k(S, c) = & I_{\{c=c_{k-1} \wedge c=c_{k+1}\}} (I_{\{no_k^g+1+no_k^d > L_{max}\}}) \\ & I_{\{c=c_{k-1} \wedge c \neq c_{k+1}\}} (I_{\{no_k^g+1 > L_{max}\}} + I_{\{no_k^d > L_{max}\}}) \\ & I_{\{c \neq c_{k-1} \wedge c=c_{k+1}\}} (I_{\{no_k^g > L_{max}\}} + I_{\{no_k^d+1 > L_{max}\}}) \\ & I_{\{c \neq c_{k-1} \wedge c \neq c_{k+1}\}} (I_{\{no_k^g > L_{max}\}} + I_{\{no_k^d > L_{max}\}}) \end{aligned}$$

Le nombre de dépassements en S dans le voisinage de i et j est $Depas_i(S, c_i) + Depas_j(S, c_j)$ puisque les voisinages sont indépendants. On s'intéresse maintenant au nombre de dépassements en S' . Puisque $i \notin Voisinage_j$ et $j \notin Voisinage_i$ les voisinages de i et j ne changent pas en S' et on a $no_i^g = no_i^g$, $no_i^d = no_i^d$ et $no_j^g = no_j^g$, $no_j^d = no_j^d$. Les changements ne sont qu'au niveau des couleurs des positions i et j . On peut donc réutiliser la formule pour le calcul des dépassements en S . Ainsi, le nombre de dépassements en S' dans le voisinage des positions i et j est $Depas_i(S, c_j) + Depas_j(S, c_i)$.

Donc :

$$\Delta_{Depas}(S', S) = Depas_i(S, c_i) + Depas_j(S, c_j) - Depas_i(S, c_j) - Depas_j(S, c_i)$$

b) L'évaluation de $\Delta_{Depas}(S', S)$ si $i \in Voisinage_j$ ou $j \in Voisinage_i$

Si $i = j - 1$ alors $i \in Voisinage_j$ et $j \in Voisinage_i$, sinon soit $i \in Voisinage_j$ soit $j \in Voisinage_i$ puisque les véhicules correspondants sont de couleurs différentes. La valeur du voisinage conjoint de i et j peut donc être obtenue à partir de la somme des deux voisinages moins la partie commune.

$$\begin{aligned} Depas_{ij}(S) = & Depas_i(S, c_i) + Depas_j(S, c_j) - \\ & - I_{\{c_j=c_{j-1} \wedge c_j=c_{j+1}\}} (I_{\{no_j^g+1+no_j^d > L_{max}\}}) - I_{\{c_j=c_{j-1} \wedge c_j \neq c_{j+1}\}} I_{\{no_j^g+1 > L_{max}\}} - \\ & - I_{\{c_j \neq c_{j-1} \wedge c_j=c_{j+1}\}} I_{\{no_j^g > L_{max}\}} - I_{\{c_j \neq c_{j-1} \wedge c_j \neq c_{j+1}\}} I_{\{no_j^g > L_{max}\}} = \\ & = Depas_i(S, c_i) + I_{\{c_j=c_{j-1} \wedge c_j \neq c_{j+1}\}} I_{\{no_j^d > L_{max}\}} + \\ & + I_{\{c_j \neq c_{j-1} \wedge c_j=c_{j+1}\}} I_{\{no_j^d+1 > L_{max}\}} + I_{\{c_j \neq c_{j-1} \wedge c_j \neq c_{j+1}\}} I_{\{no_j^d > L_{max}\}} = \\ & = Depas_i(S, c_i) + I_{\{c_j \neq c_{j+1}\}} I_{\{no_j^d > L_{max}\}} + I_{\{c_j \neq c_{j-1} \wedge c_j=c_{j+1}\}} I_{\{no_j^d+1 > L_{max}\}} \end{aligned}$$

Pour S' , avec $i < j$, les valeurs no_i^g et no_j^d ne changent pas puisqu'il n'y a pas de véhicules enlevés ou ajoutés à ces sous-séquences, et donc, $no_i^g = no_i^g$ et $no_j^d = no_j^d$. Pour no_i^d , il y a trois cas selon que v_i a la même couleur que les voisins de j ou non.

$$no_i^d = I_{\{c_i \neq c_{j-1}\}} no_i^d + I_{\{c_i=c_{j-1} \wedge c_i \neq c_{j+1}\}} (no_i^d + 1) + I_{\{c_i=c_{j-1} \wedge c_i=c_{j+1}\}} (no_i^d + 1 + no_j^g)$$

Pour no_j^g , il y a aussi trois cas selon que v_j a la même couleur que les voisins de i ou non.

$$no_j^g = I_{\{c_j \neq c_{i+1}\}} no_j^g + I_{\{c_j = c_{i+1} \wedge c_j \neq c_{i-1}\}} (no_j^g + 1) + I_{\{c_j = c_{i+1} \wedge c_j = c_{i-1}\}} (no_j^g + 1 + no_j^d)$$

La valeur du voisinage conjoint de i et j en S' correspond donc à la formule pour S , mais en utilisant les nouvelles valeurs no_j^g , no_i^d , $c'_i = c_j$ et $c'_j = c_i$,

$$Depas_{ij}(S') = Depas_i(S', c_j) + I_{\{c_i \neq c_{j+1}\}} I_{\{no_j^d > L_{max}\}} + I_{\{c_i \neq c'_{j-1} \wedge c_i = c_{j+1}\}} I_{\{no_i^d + 1 > L_{max}\}}$$

où (sachant que no_i^g et c_{i-1} ne changent pas de S à S') :

$$Depas_i(S', c_j) = I_{\{c_j = c_{i-1} \wedge c_j = c'_{i+1}\}} (I_{\{no_i^g + 1 + no_i^d > L_{max}\}}) I_{\{c_j = c_{i-1} \wedge c_j \neq c'_{i+1}\}} (I_{\{no_i^g + 1 > L_{max}\}} + I_{\{no_i^d > L_{max}\}}) I_{\{c_j \neq c_{i-1} \wedge c_j = c'_{i+1}\}} (I_{\{no_i^g > L_{max}\}} + I_{\{no_i^d + 1 > L_{max}\}}) I_{\{c_j \neq c_{i-1} \wedge c_j \neq c'_{i+1}\}} (I_{\{no_i^g > L_{max}\}} + I_{\{no_i^d > L_{max}\}})$$

Il faut noter que le cas $i = j - 1$ est inclus dans la définition des c'_{i+1} et c'_{j-1} en S' . Donc, nous obtenons finalement :

$$\Delta_{Depas}(S', S) = Depas_{ij}(S') - Depas_{ij}(S)$$

Le calcul de $\Delta_{Depas}(S', S)$ et $\Delta_{Purges}(S', S)$ se fait en $O(1)$ en utilisant les valeurs pré-calculées et donc le calcul $\Delta(S', S)_{couleur} = M * \Delta_{Depas}(S', S) + \Delta_{Purges}(S', S)$ se fait en $O(1)$.

5.2.2.3 La mise à jour des valeurs pré-calculées no_i^d et no_i^g après un échange

actualiser_couleur(i_1, i_2)

1. si $i_1 = 1$, alors $no_i^g = 0$
2. si $c_i = c_{i-1}$, alors $i = i_1 - no_i^g$,
sinon $i = i_1$
3. si $c_i = c_{i-1}$, alors $i_2 = i_2 + no_i^d$
4. $j = i$
5. tant que $c_i = c_{i+1}$
 - 5.1 $longueur_s = longueur_s + 1$
 - 5.2 $i = i + 1$
 - 5.3 $no_i^g = i - j$

6. si $j - 1 > 0$, alors $no_{j-1}^d = i - j$
 7. pour k allant de j à $i - 2$ faire $no_k^d = i - k$
 8. si $i < i_2$, aller à 4.
- fin

Comme on le voit, la variable i est toujours incrémentée de 1 et prend des valeurs entre i_1 et i_2 . Les opérations liées à cette variable sont en $O(i_2 - i_1)$. On ne revient en arrière que pour mettre à jour les valeurs no_k^d à l'étape 7, une fois qu'on a identifié la sous-séquence. Puisque k prend nécessairement des valeurs entre i_1 et i_2 , les opérations liées à k sont donc en $O(i_2 - i_1)$. Ainsi, le pré-calcul se fait en $O(l)$, où l est la longueur de la sous-séquence pour laquelle on met à jour les valeurs no_i^d et no_i^g .

Lors d'un échange(i, j), on met à jour les valeurs pour les voisinages immédiats de i et j . Ces voisinages combinés ne peuvent avoir qu'au plus la longueur $4 * L_{max} + 2$ et donc la mise à jour pour un échange est dans $O(L_{max})$.

5.3 Voisinage par extraction\insertion

Le voisinage par extraction\insertion consiste à obtenir de nouvelles solutions à partir de la solution courante en extrayant un véhicule et en le réinsérant entre deux autres véhicules de la séquence. Soit $S \oplus ExtIns(i, j)$ l'opérateur qui correspond à l'extraction du véhicule v_i et sa réinsertion entre les véhicules v_{j-1} et v_j . On a alors :

$$S \oplus ExtIns(i, j) = \{v'_1, \dots, v'_N\}, \text{ où :}$$

- Si $i < j$, alors
 - $v'_l = v_l, \forall l, l < i$,
 - $v'_l = v_{l+1}, \forall l, i < l \leq j - 1$,
 - $v'_{j-1} = v_i$,
 - $v'_l = v_l, \forall l, l > j - 1$.
- Si $j < i$, alors
 - $v'_l = v_l, \forall l, l < j$,
 - $v'_j = v_i$,
 - $v'_l = v_{l-1}, \forall l, j < l \leq i$,
 - $v'_l = v_l, \forall l, l < i$.

$$N(S) = \{S' | S' = S \oplus ExtIns(i, j), \forall (i, j) \text{ tel que } i, j \in [1, N], Tabou_{v_i} = 0 \text{ et } i \neq j\}$$

Si on note S la solution courante, alors le voisinage par extraction\insertion est :

$$N(S) = \{S' | S' = S \oplus ExtIns(i, j), \forall (i, j) \text{ tel que } i, j \in [1, N] \text{ et } i \neq j\}$$

5.3.1 Le coût d'une $ExtIns(i, j)$ pour les contraintes de ratio

Notons :

$$\begin{aligned}
 S' &= S \oplus ExtIns(i, j) \\
 A_k &= \sum_{l=k}^{k+Den_r-1} v_{lr} - Num_r \\
 A_k^- &= \sum_{l=k}^{k+Den_r-2} v_{lr} - Num_r \\
 A_k^+ &= \sum_{l=k}^{k+Den_r} v_{lr} - Num_r \\
 A_{ik} &= \sum_{l=k, l \neq i}^{k+Den_r-1} v_{lr} - Num_r \\
 A'_k &= \sum_{l=k}^{k+Den_r-1} v'_{lr} - Num_r \\
 A'_{ik} &= \sum_{l=k, l \neq i}^{k+Den_r-1} v'_{lr} - Num_r
 \end{aligned}$$

Comme pour le voisinage par échanges, on s'intéresse au calcul de $\Delta_r(S', S) = Val_r(S') - Val_r(S)$ pour une contrainte de ratio r donnée.

5.3.1.1 L'évaluation de $\Delta_r(S', S)$ sous l'hypothèse $|i - j| \geq Den_r$

Soit $S' = S \oplus ExtIns(i, j)$. Nous calculons $\Delta_r(S', S)$ sous l'hypothèse $j - i \geq Den_r$. Sous cette hypothèse, nous pouvons décomposer la somme correspondant à $Val_r(S)$ ainsi :

$$\begin{aligned}
 Val_r(S) &= \sum_{k=1}^{i-Den_r} A_k I_{\{A_k > 0\}} + \sum_{k=i-Den_r+1}^i A_k I_{\{A_k > 0\}} + \sum_{k=i+1}^{j-Den_r} A_k I_{\{A_k > 0\}} + \\
 &+ \sum_{k=j-Den_r+1}^{j-1} A_k I_{\{A_k > 0\}} + \sum_{k=j}^N A_k I_{\{A_k > 0\}}
 \end{aligned} \tag{5.15}$$

Par ailleurs, nous décomposons la somme correspondant à $Val_r(S')$ ainsi :

$$\begin{aligned}
 Val_r(S') &= \sum_{k=1}^{i-Den_r} A'_k I_{\{A'_k > 0\}} + \sum_{k=i-Den_r+1}^{i-1} A'_k I_{\{A'_k > 0\}} + \sum_{k=i}^{j-1-Den_r} A'_k I_{\{A'_k > 0\}} + \\
 &+ \sum_{k=j-1-Den_r+1}^{j-1} A'_k I_{\{A'_k > 0\}} + \sum_{k=j}^N A'_k I_{\{A'_k > 0\}}
 \end{aligned} \tag{5.16}$$

En utilisant la définition de v'_{lr} , on obtient les relations suivantes pour A'_k et A_k :

- $A'_k = A_k, \forall k < i - Den_r$ puisque $v'_{lr} = v_{lr}$ pour $l < i$;

- $A'_k = A_{k+1} \forall k, i \leq k \leq j-1 - Den_r$ puisque $v'_l = v_{l+1}, \forall l, i \leq l < j-1$;
- $A'_k = A_k, \forall k \geq j$ puisque $v'_l = v_l, \forall l, l > j-1$.

Donc, $Val_r(S')$, Eq. (5.16) peut se réécrire comme suit :

$$\begin{aligned}
 Val_r(S') &= \sum_{k=1}^{i-Den_r} A_k I_{\{A_k > 0\}} + \sum_{k=i-Den_r+1}^{i-1} A'_k I_{\{A'_k > 0\}} + \sum_{k=i+1}^{j-Den_r} A_k I_{\{A_k > 0\}} + \\
 &+ \sum_{k=j-1-Den_r+1}^{j-1} A'_k I_{\{A'_k > 0\}} + \sum_{k=j}^N A_k I_{\{A_k > 0\}}
 \end{aligned} \tag{5.17}$$

Comme les premier, troisième et cinquième termes de droite dans les équations (5.17) et (5.15) sont identiques, ils disparaissent dans le calcul de $Val_r(S') - Val_r(S)$:

$$\begin{aligned}
 \Delta_r(S', S) &= Val_r(S') - Val_r(S) = \\
 &= \sum_{k=i-Den_r+1}^{i-1} A'_k I_{\{A'_k > 0\}} - \sum_{k=i-Den_r+1}^i A_k I_{\{A_k > 0\}} + \\
 &+ \sum_{k=j-1-Den_r+1}^{j-1} A'_k I_{\{A'_k > 0\}} - \sum_{k=j-Den_r+1}^{j-1} A_k I_{\{A_k > 0\}}
 \end{aligned} \tag{5.18}$$

Quand k prend des valeurs entre $i - Den_r + 1$ et $i - 1$, nous avons $A'_k = \sum_{l=k}^{i-1} v_{lr} + \sum_{l=i}^{k+Den_r-1} v_{(l+1)r}$. Pour rendre cette somme facile à calculer de façon itérative en passant de i à $i + 1$, on ajoute et on enlève v_{ir} dans A'_k . On obtient alors :

$$A'_k = \sum_{l=k}^{k+Den_r} v_{lr} - v_{ir}.$$

Ainsi, le premier terme de (5.18) peut se réécrire comme

$$\sum_{k=i-Den_r+1}^{i-1} (-v_{ir} + A_k^+) I_{\{-v_{ir} + A_k^+ > 0\}}.$$

Quand k prend des valeurs entre $j - 1 - Den_r + 1$ et $j - 1$, nous avons

$$A'_k = \sum_{l=k}^{j-2} v_{(l+1)r} + v_{ir} + \sum_{l=j}^{k+Den_r-1} v_{lr}.$$

En utilisant le fait que $\sum_{l=j}^{j-2} v_{(l+1)r} = \sum_{l=j+1}^{j-1} v_{lr}$, nous avons

$$A'_k = \sum_{l=k+1}^{k+Den_r-1} v_{lr} + v_{ir}$$

et donc le troisième terme de l'Eq. (5.18) peut se réécrire comme

$$\sum_{k=j-Den_r+1}^j (v_{ir} + A_k^-) I_{\{v_{ir} + A_k^- > 0\}}.$$

Nous obtenons enfin :

$$\begin{aligned} \Delta_r(S', S) = Val_r(S') - Val_r(S) = & \\ & \sum_{k=i-Den_r+1}^{i-1} (-v_{ir} + A_k^+) I_{\{-v_{ir} + A_k^+ > 0\}} - \sum_{k=i-Den_r+1}^i A_k I_{\{A_k > 0\}} + \\ & + \sum_{k=j-Den_r+1}^j (v_{ir} + A_k^-) I_{\{v_{ir} + A_k^- > 0\}} - \sum_{k=j-Den_r+1}^{j-1} A_k I_{\{A_k > 0\}} \end{aligned} \quad (5.19)$$

Si $i - j \geq Den_r$, on obtient le même résultat que dans le cas $j - i \geq Den_r$ puisque la structure de la séquence autour des positions i et j en S' est la même, plus précisément $v_{i-Den_r+1}, \dots, v_{i-1}, v_{i+1}, \dots, v_{i+Den_r-1}, v_{j-Den_r+1}, \dots, v_{j-1}, v_i, v_j, \dots, v_{i+Den_r-2}$.

5.3.1.2 Valeurs pré-calculées sous l'hypothèse $|i - j| \geq Den_r$

A partir de (5.19), on choisit comme valeurs pré-calculées

- $UExt_i^0 = \sum_{k=i-Den_r+1}^{i-1} A_k^+ I_{\{A_k^+ > 0\}} - \sum_{k=i-Den_r+1}^i A_k I_{\{A_k > 0\}}$, pour l'extraction des véhicules i avec $v_{ir} = 0$,
- $UExt_i^1 = \sum_{k=i-Den_r+1}^{i-1} (-1 + A_k^+) I_{\{-1 + A_k^+ > 0\}} - \sum_{k=i-Den_r+1}^i A_k I_{\{A_k > 0\}}$, pour l'extraction des véhicules i avec $v_{ir} = 1$,
- $UIns_j^0 = \sum_{k=j-Den_r+1}^j A_k^- I_{\{A_k^- > 0\}} - \sum_{k=j-Den_r+1}^{j-1} A_k I_{\{A_k > 0\}}$, pour l'insertion des véhicules v_i avec $v_{ir} = 0$,
- $UIns_j^1 = \sum_{k=j-Den_r+1}^j (1 + A_k^-) I_{\{1 + A_k^- > 0\}} - \sum_{k=j-Den_r+1}^{j-1} A_k I_{\{A_k > 0\}}$, pour l'insertion des véhicules v_i avec $v_{ir} = 1$.

Soit $UExt_i = UExt_i^{v_{ir}}$. En mémoire, on n'a besoin que de $UIns_i^0$, $UIns_i^1$ et $UExt_i$. Avec ces valeurs pré-calculées, on trouve facilement que $\Delta_r(S', S) = UExt_i + UIns_j^{v_{ir}}$.

5.3.1.3 La mise à jour des valeurs pré-calculées après une extraction\insertion

Après une extraction\insertion, il faut mettre à jour les valeurs pré-calculées U_{ir} pour tout véhicule entre les positions $[i - Den_r + 1, i + Den_r - 2]$ et $[j - Den_r + 1, j + Den_r]$.

Les autres véhicules ne subissent aucun changement dans leur voisinage et ne sont pas concernés par l'extraction ou réinsertion du véhicule v_i .

On peut faire la mise à jour d'une séquence de longueur L en $O(L + Den_r)$, de la même façon que pour les échanges à la Section 5.2.1.3. Ainsi, les calculs nécessaires pour la mise à jour des valeurs pré-calculées entre les positions $[i - Den_r + 1, i + Den_r - 2]$ et $[j - Den_r + 1, j + Den_r]$ se font en

$$O([(i - Den_r + 1) - (i + Den_r - 2) + Den_r] + [(j - Den_r - 1) - (j - Den_r) + Den_r]) = O(Den_r).$$

5.3.2 Le coût d'une $ExtIns(i, j)$ pour le nombre de purges de peinture

Notations

N	nombre total de véhicules
N_0	début de la journée J
c_i	la couleur du véhicule i
L_{max}	la longueur maximale pour une sous-séquence de véhicules consécutifs de la même couleur
no_i^g	le nombre de véhicules consécutifs avec la même couleur situés à gauche de v_i
no_i^d	le nombre de véhicules consécutifs avec la même couleur situés à droite de v_i

On s'intéresse en même temps au nombre de purges de peinture et à la contrainte forte de ne pas dépasser la longueur maximale d'une séquence.

$$\Delta_{Couleur}(S', S) = M * \Delta_{Depas}(S', S) + \Delta_{Purges}(S', S)$$

$$S' = S \oplus ExtIns(i, j) \text{ avec } \{c'_i\}_{i=1, \dots, N} \text{ ou}$$

- si $i < j$,
 - $c'_k = c_k \forall k, k < i$
 - $c'_k = c_{k+1} \forall i, k, \leq k < j - 1$
 - $c'_{j-1} = c_i$
 - $c'_k = c_k \forall k, k \geq j$
- si $j > i$,
 - $c'_k = c_k \forall k < j$
 - $c'_j = c_i$
 - $c'_k = c_{k-1} \forall k, j < k \leq i$
 - $c'_k = c_k \forall k, k > j$

- si $i = j - 1$, l'insertion ne change pas la séquence ; après $ExtIns(i, j)$, le véhicule v_i est toujours à la position i .
- si $i = j + 1$, le mouvement $ExtIns(i, j)$ est équivalent à $échange(i, j)$.
- si $i > j$, le mouvement $ExtIns(i, j)$ est équivalent à $ExtIns(i', j')$ avec $i' < j'$; il suffit d'inverser l'ordre de la séquence avec $i' = N - i < j' = N - j$ et on traite $ExtIns(N - i, N - j - 1)$. Pour la suite de l'analyse, nous assumons donc $i < j$.

5.3.2.1 L'évaluation du $\Delta_{Purges}(S', S)$, le nombre de purges causées par une extraction\insertion

L'objectif lié au nombre de purges est facile à évaluer. Il y a une purge de peinture chaque fois qu'on change de couleur, ce qui correspond à

$$\sum_{k=N_0}^N I_{\{c_k \neq c_{k-1}\}}.$$

Tout comme pour les échanges, lorsqu'on fait une extraction\insertion, il suffit de vérifier une courte suite de scénarios pour obtenir Δ_{Purges} , où les seules informations nécessaires sont $c_{i-1}, c_i, c_{i+1}, c_{j-1}, c_j, c_{j+1}$.

Si $i < j - 1$, alors

$$\begin{aligned} \Delta_{Purges}(S', S) &= \sum_{k=N_0}^N I_{c'_k \neq c'_{k-1}} - \sum_{k=N_0}^N I_{c_k \neq c_{k-1}} = \\ &I_{c_{i-1} \neq c_{i+1}} + I_{c_{j-1} \neq c_j} + I_{c_i \neq c_{j+1}} - I_{c_{i-1} \neq c_i} - I_{c_i \neq c_{i+1}} - I_{c_{j-1} \neq c_j} \end{aligned}$$

La formule est la même si $j < i - 1$ puisque les voisins immédiats de v_i et v_j sont les mêmes.

5.3.2.2 L'évaluation de $\Delta_{Dépas}(S', S)$, les dépassements de la longueur maximale d'une séquence de véhicules de même couleur causés par une extraction\insertion

Soit S_k une séquence maximale de véhicules de même couleur. La valeur de S au niveau des dépassements de longueur maximale est la somme :

$$\sum_{\forall S_k} I_{\{|S_k| > L_{max}\}}$$

On aimerait avoir l'évaluation d'une extraction \ insertion en $O(1)$. Pour faire ceci, nous utilisons le même voisinage immédiat des véhicules i et j que celui pour les échanges.

Ce voisinage est formé des deux séquences de véhicules de même couleur situés immédiatement à droite et à gauche des positions i et j , respectivement.

Soit no_i^g le nombre de véhicules consécutifs avec la même couleur situés à gauche de v_i et no_i^d le nombre de véhicules consécutifs avec la même couleur situés à droite de v_i . La couleur du groupe qui est à la gauche d'un véhicule i est donnée par c_{i-1} et à la droite par c_{i+1} . Nous allons traiter deux cas distincts dans le calcul de $\Delta_{Depas}(S', S)$, soit :

- a) $i \notin Voisinage_j$ et $i \notin Voisinage_j$
- b) $i \in Voisinage_j$ ou $i \in Voisinage_j$

a) L'évaluation de $\Delta_{Depas}(S', S)$ si $i \notin Voisinage_j$ et $j \notin Voisinage_i$

Puisque $i \notin Voisinage_j$ et $j \notin Voisinage_i$, les voisinages à gauche et à droite de v_i ne changent pas suite à la réinsertion de v_i en S' . Le seul changement de S à S' dans le voisinage de i est l'extraction de v_i . Soit $\Delta_{Depas}(S', S)_i$ la différence de dépassements entre S' et S dans le voisinage de i .

Nous allons procéder à une analyse de cas sur la valeur de $\Delta_{Depas}(S', S)_i$ pour l'extraction. Par la suite, ce résultat sera utilisé pour déduire le Δ correspondant à la réinsertion de v_i à la position j . En effet, $\Delta_{Depas}(S', S)_j = -\Delta_{Depas}(S, S')_j$, l'extraction étant l'opération inverse de l'insertion.

Dans le voisinage de i , il y a quatre cas possibles à traiter afin d'obtenir $\Delta_{Depas}(S', S)_i$:

1. Si $c_i = c_{i-1} \wedge c_i = c_{i+1}$, le véhicule v_i est au milieu d'une sous-séquence. Lors de son extraction la longueur totale de la séquence diminue de 1.

$$\Delta_{Depas}(S', S)_i^{c_i=c_{i-1} \wedge c_i=c_{i+1}} = I_{\{no_i^g+no_i^d>L_{max}\}} - I_{\{no_i^g+1+no_i^d>L_{max}\}}$$

2. Si $c_i = c_{i-1} \wedge c_i \neq c_{i+1}$, le véhicule v_i est l'extrémité droite d'une sous-séquence. Lors de son extraction la longueur totale de la sous-séquence gauche diminue de 1, la sous-séquence droite demeure inchangée.

$$\Delta_{Depas}(S', S)_i^{c_i=c_{i-1} \wedge c_i \neq c_{i+1}} = I_{\{no_i^g>L_{max}\}} - I_{\{no_i^g+1>L_{max}\}}$$

3. Si $c_i \neq c_{i-1} \wedge c_i = c_{i+1}$, le véhicule v_i est l'extrémité gauche d'une sous-séquence. Lors de son extraction, la longueur totale de la sous-séquence droite diminue de 1, la sous-séquence gauche demeure inchangée.

$$\Delta_{Depas}(S', S)_i^{c_i \neq c_{i-1} \wedge c_i=c_{i+1}} = I_{\{no_i^d>L_{max}\}} - I_{\{1+no_i^d>L_{max}\}}$$

4. Si $c_i \neq c_{i-1} \wedge c_i \neq c_{i+1}$, le véhicule v_i est une sous-séquence en soi. Le véhicule ne peut pas causer un dépassement puisque $L_{max} > 1$. Les sous-séquences à gauche et à droite peuvent toutefois contenir des dépassements. Lors de l'extraction, si

$c_{i-1} \neq c_{i+1}$, rien ne change et $\Delta_{Depas}(S', S)$ est 0, sinon les deux sous-séquences sont fusionnées.

$$\Delta_{Depas}(S', S)_i^{c_i \neq c_{i-1} \wedge c_i \neq c_{i+1}} = I_{\{c_{i-1}=c_{i+1}\}} (I_{\{no_i^g + no_i^d > L_{max}\}} - I_{\{no_i^g > L_{max}\}} - I_{\{no_i^d > L_{max}\}})$$

En mettant tous les cas ensemble, on obtient :

$$\begin{aligned} \Delta_{Depas}(S', S)_i = & I_{\{c_i=c_{i-1} \wedge c_i=c_{i+1}\}} (I_{\{no_i^g + no_i^d > L_{max}\}} - I_{\{no_i^g + 1 + no_i^d > L_{max}\}}) \\ & I_{\{c_i=c_{i-1} \wedge c_i \neq c_{i+1}\}} (I_{\{no_i^g > L_{max}\}} - I_{\{no_i^g + 1 > L_{max}\}}) \\ & I_{\{c_i \neq c_{i-1} \wedge c_i=c_{i+1}\}} (I_{\{no_i^d > L_{max}\}} - I_{\{1 + no_i^d > L_{max}\}}) \\ & I_{\{c_i \neq c_{i-1} \wedge c_i \neq c_{i+1} \wedge c_{i-1}=c_{i+1}\}} (I_{\{no_i^g + no_i^d > L_{max}\}} - I_{\{no_i^g > L_{max}\}} - I_{\{no_i^d > L_{max}\}}) \end{aligned} \quad (5.20)$$

En utilisant le fait que l'extraction est l'opération inverse de l'insertion, on peut ré-écrire la formule précédente pour S' avec la nouvelle position de v_i . Puisque $i \notin \text{Voisinage}_j$ et $j \notin \text{Voisinage}_i$, la seule modification dans le voisinage à droite de $j-1$ et à gauche de j correspond à l'insertion de v_i entre les deux positions. Ainsi $no_i^g = no_j^g$ et $no_i^d = no_{j-1}^d$.

$$\begin{aligned} \Delta_{Depas}(S', S)_j = & -\Delta_{Depas}(S, S')_j = \\ = & I_{\{c_i=c_{j-1} \wedge c_i=c_j\}} (-I_{\{no_j^g + no_{j-1}^d > L_{max}\}} + I_{\{no_j^g + 1 + no_{j-1}^d > L_{max}\}}) + \\ & + I_{\{c_i=c_{j-1} \wedge c_i \neq c_j\}} (-I_{\{no_j^g > L_{max}\}} + I_{\{no_j^g + 1 > L_{max}\}}) + \\ & + I_{\{c_i \neq c_{j-1} \wedge c_i=c_{j+1}\}} (-I_{\{no_{j-1}^d > L_{max}\}} + I_{\{1 + no_{j-1}^d > L_{max}\}}) + \\ & + I_{\{c_i \neq c_{j-1} \wedge c_i \neq c_j \wedge c_{j-1}=c_j\}} (-I_{\{no_j^g + no_{j-1}^d > L_{max}\}} + I_{\{no_j^g > L_{max}\}} + I_{\{no_{j-1}^d > L_{max}\}}) \end{aligned} \quad (5.21)$$

En conclusion, la valeur de $\Delta_{Depas}(S', S)$ est

$$\Delta_{Depas}(S', S) = \Delta_{Depas}(S', S)_i + \Delta_{Depas}(S', S)_j \quad (5.22)$$

b) L'évaluation de $\Delta_{Depas}(S', S)$ si $i \in \text{Voisinage}_j$ ou $j \in \text{Voisinage}_i$

Soit $i < j$, on peut résumer la discussion à deux grands cas :

- $i < j$ et $i \in \text{Voisinage}_j$
- $i < j$, $i \notin \text{Voisinage}_j$ et donc $j \in \text{Voisinage}_i$

a. $i < j$ et $i \in \text{Voisinage}_j$

Ceci correspond à $i > j - no_j^g$, ce qui est équivalent à dire que $c_i = c_{i+1} = \dots = c_{j-1}$. Le véhicule v_i fait partie d'une sous-séquence de même couleur située à la gauche du véhicule v_j . Après l'extraction et l'insertion, le véhicule v_i appartiendra à la même sous-séquence. Donc, au niveau des dépassements, il n'y a aucun changement de S à S' et $\Delta_{Depas}(S', S) = 0$, peu importe les autres conditions.

b. $i < j$, $i \notin \text{Voisinage}_j$ et $j \in \text{Voisinage}_i$

Puisque $j \in \text{Voisinage}_i$, $i + no_i^d > j$, ce qui est équivalent à $c_i \neq c_{i+1} = \dots = c_{j-1} = c_j$ ($c_i \neq c_{i+1}$, sinon on se trouve de nouveau dans le cas $i \in \text{Voisinage}_j$). Les modifications suite à l'extraction sont :

i. Si $c_{i-1} = c_i$, le véhicule v_i est à l'extrémité droite d'une sous-séquence. Dans ce cas, la longueur de la sous-séquence diminue de 1 suite à l'extraction. Pour $\Delta_{Depas}(S', S)$, la contribution sera

$$I_{\{c_{i-1}=c_i\}}(I_{\{no_i^g > L_{max}\}} - I_{\{no_i^g + 1 > L_{max}\}})$$

ii. Si $c_{i-1} \neq c_i$, le véhicule v_i est une sous-séquence en soi, cette sous-séquence ne peut pas causer de dépassements.

Les modifications suite à l'insertion sont :

Puisque $c_i \neq c_{j-1}$ et $c_j = c_{j-1}$, l'insertion du véhicule v_i entre v_{j-1} et v_j brise la sous-séquence dont v_{j-1} et v_j faisaient partie.

i. Si $c_{i-1} \neq c_{i+1}$, la nouvelle sous-séquence contenant v_{j-1} a $j - 1 - i$ véhicules, et celle contenant v_j a no_{j-1}^d véhicules. Pour $\Delta_{Depas}(S', S)$, la contribution sera

$$I_{\{c_{i-1} \neq c_{i+1}\}}(I_{\{j-1-i > L_{max}\}} + I_{\{no_{j-1}^d > L_{max}\}} - I_{\{no_j^g + no_{j-1}^d > L_{max}\}})$$

ii. Si $c_{i-1} = c_{i+1}$, la sous-séquence à gauche de i et celle à droite de i deviennent une seule sous-séquence suite à l'extraction. Alors, dans S' , il y aura à la gauche de v_i $no_i^g + j - 1 - i$ véhicules et à la droite no_{j-1}^d véhicules. Pour $\Delta_{Depas}(S', S)$, la contribution sera

$$I_{\{c_{i-1}=c_{i+1}\}}(I_{\{no_i^g + (j-1-i) > L_{max}\}} - I_{\{no_i^g > L_{max}\}} - I_{\{no_i^d > L_{max}\}})$$

En sommant sur tous les cas possibles lors de l'extraction et de l'insertion, on obtient finalement :

$$\begin{aligned} \Delta_{Depas}(S', S) = & I_{\{c_{i-1}=c_i\}}(I_{\{no_i^g > L_{max}\}} - I_{\{no_i^g + 1 > L_{max}\}}) + \\ & + I_{\{c_{i-1} \neq c_{i+1}\}}(I_{\{j-1-i > L_{max}\}} + I_{\{no_{j-1}^d > L_{max}\}} - I_{\{no_j^g + no_{j-1}^d > L_{max}\}}) + \\ & + I_{\{c_{i-1}=c_{i+1}\}}(I_{\{no_i^g + (j-1-i) > L_{max}\}} - I_{\{no_i^g > L_{max}\}} - I_{\{no_i^d > L_{max}\}}) \end{aligned}$$

CHAPITRE 6

RÉSULTATS

Dans ce chapitre, nous présentons d'abord une description générale des instances, suivie des résultats et d'une analyse de l'évolution de la qualité des solutions en fonction des temps de calcul. Nous présentons aussi des résultats touchant à la difficulté des instances et à la complexité en temps de calcul de l'algorithme. Pour ce qui est des résultats, les meilleures solutions connues pour les instances considérées sont comparées avec celles des deux heuristiques décrites dans le chapitre 2, la première utilisant la *Relaxation séquentielle* et la deuxième utilisant la *Relaxation stochastique*. Toutes les valeurs ont été obtenues dans les conditions du Challenge ROADEF [5], soit 600 secondes de temps de calcul sur un PC Pentium4/1.6 Ghz/win2000/1 Go RAM.

Trois ensembles d'instances tests ont été proposés pour la compétition ROADEF. Deux étaient fournis aux participants pendant la compétition, soit les ensembles *A* et *B*. Un troisième, l'ensemble *X*, n'a été révélé que pour la phase finale de la compétition. Les ensembles *A*, *B* et *X* contiennent 16, 45 et 19 instances, respectivement, pour un total de 80 instances.

Globalement, nous obtenons la meilleure solution connue pour l'objectif de premier niveau pour 77 instances et la meilleure solution connue pour les trois niveaux d'objectifs pour 10 instances. De plus, en donnant le même poids à toutes les instances et en sommant sur les valeurs brutes des solutions, nous nous classons au troisième rang parmi les 18 équipes participantes.

La nature du problème ainsi que les approches utilisées diffèrent selon la hiérarchie des objectifs. Nous allons donc regrouper les résultats selon les trois grands types de problèmes induits par cette hiérarchie des objectifs, soit :

- *EP_ENP*, contraintes de ratio prioritaires, contraintes de ratio non prioritaires, purges de peinture ;
- *EP_RAF*, contraintes de ratio prioritaires, purges de peinture, contraintes de ratio non prioritaires ;
- *RAF_EP*, rafales de peinture, contraintes de ratio prioritaires, contraintes de ratio non prioritaires.

Comme on peut le voir dans le Tableau 6.1, les instances de type *EP_ENP*, *EP_RAF* et *RAF_EP* ne sont pas uniformément distribuées entre les ensembles *A*, *B* et *X*. Ainsi, on a 22, 36 et 22 instances de type *EP_ENP*, *EP_RAF* et *RAF_EP*, respectivement. Dans les tableaux à venir, où les résultats sont divisés par type de problèmes, nous ajoutons donc pour chaque instance sa provenance (*A*, *B* ou *X*) à la fin de l'identificateur.

Type de problème	Set A	Set B	Set X	Σ
EP_ENP	4	15	3	22
EP_RAF	8	15	13	36
RAF_EP	4	15	3	22
Σ	16	45	19	80

Tableau 6.1 – Provenance des instances

Notation

M	nombre total de modèles de véhicules
M_{EP}	nombre de modèles avec contraintes de ratio prioritaires
N	nombre total de véhicules
N_C	nombre de couleurs différentes
\overline{Den}	moyenne des dénominateurs des contraintes de ratio
Dif^{max}	maximum des difficultés des contraintes de ratio (voir equation 3.1)

6.1 Instances EP_ENP

Dans les 22 instances EP_ENP, le nombre de modèles M varie entre 11 et 340 pour une moyenne de 157. En considérant seulement le nombre de modèles associés à des contraintes de ratio prioritaires, M_{EP} , nous retrouvons entre 2 et 26 modèles pour une moyenne de 11. Si le nombre de modèles est assez peu élevé, il y a par contre entre 65 et 1315 véhicules par instance, pour une moyenne de 718 véhicules.

Le nombre de contraintes de ratio prioritaires varie entre 1 et 11 et est en moyenne de 5. Par ailleurs, le nombre de contraintes de ratio non-prioritaires varie entre 1 et 19 et est en moyenne de 9. Le dénominateur moyen est de 7,2 et 9,2, pour les contraintes de ratio prioritaires et non-prioritaires, respectivement. La difficulté maximale peut aller jusqu'à 2, 17 et 4, 17, pour les contraintes de ratio prioritaires et non-prioritaires, respectivement. Enfin, le nombre de couleurs varie entre 4 et 25 pour une moyenne de 14.

Il faut noter que le nombre de contraintes de ratio prioritaires est deux fois moins élevé en moyenne que le nombre de contraintes de ratio non-prioritaires. Cela veut dire qu'en relaxant les contraintes non-prioritaires, le temps de calcul peut être réduit de 66% environ. En pratique, le temps où la relaxation est appliquée dépend du nombre de contraintes non-prioritaires, de leur dénominateur moyen, ainsi que de la valeur courante de l'objectif de premier niveau.

La complexité des heuristiques de construction de premier niveau est $O(N * M_{EP} * N_{EP})$. La construction est donc favorisée par un petit nombre de modèles associé à des contraintes de ratio prioritaires ainsi que par un petit nombre de contraintes de ratio prioritaires. Cela permet une utilisation efficace de l'heuristique de construction dynamique ou de l'utilisation répétitive de l'heuristique de construction stochastique.

La complexité liée à une itération tabou est $O(N_R * N^2)$, où N_R est le nombre total

de contraintes de ratio, soit $N_R^{EP} + N_R^{ENP}$. Cette complexité est due à l'évaluation du voisinage. C'est donc une quantité qu'on peut réduire en limitant la taille du voisinage exploré.

Dans le tableau 6.3, nous présentons les résultats obtenus pour les instances *EP_ENP*. La première colonne contient l'identificateur de l'instance. La deuxième colonne contient les solutions de départ fournies par Renault. Nos solutions se trouvent dans les colonnes *Algorithme original* (relaxation séquentielle) et Algorithme avec relaxation stochastique. Dans chacun des cas, les valeurs numériques correspondent à l'écart avec la meilleure solution rapportée. Par exemple, un écart de 0,0522 signifie que la solution est à 5,22% de la meilleure solution.

Pour les 22 instances *EP_ENP*, nous avons obtenu la meilleure solution de premier niveau pour 21 instances, la meilleure solution de premier et deuxième niveau pour 12 instances, et la meilleure solution pour les trois niveaux d'objectifs pour 3 instances. On observe un écart moyen de 9,6% avec la meilleure solution. Par contre pour 14 instances sur 22 on obtient un écart inférieur à 1%. Il y a entre autres un petit nombre d'instances où les résultats sont mauvais et qui dégradent beaucoup la moyenne.

<i>Instances EP_ENP</i>	<i>M</i>	<i>M_{EP}</i>	<i>N</i>	<i>N_C</i>	<i>EP</i>			<i>ENP</i>		
					#	<i>Den</i>	<i>Di^m</i>	#	<i>Den</i>	<i>Di^m</i>
024_38_3_EP_ENP_RAF_A	238	8	1260	14	5	6.4	0.96	8	6.8	0.95
024_38_5_EP_ENP_RAF_A	215	8	1315	14	5	6.4	0.96	8	6.8	0.96
025_38_1_EP_ENP_RAF_A	288	11	1004	25	4	4.5	0.75	18	22.3	1.12
048_39_1_EP_ENP_RAF_A	230	17	600	13	5	4.4	0.99	12	8.1	0.97
022_EP_ENP_RAF_B	80	4	526	14	2	4.5	0.86	7	6.4	1.00
023_EP_ENP_RAF_B	147	19	1110	14	9	5.2	1.05	8	2.5	0.54
024_V2_EP_ENP_RAF_B	262	11	1270	14	6	12.5	1.36	7	29.6	1.26
025_EP_ENP_RAF_B	254	11	1161	20	4	4.5	0.81	12	14.1	4.17
028_ch1_EP_ENP_RAF_B	144	17	365	21	9	8.7	1.14	11	7.9	0.95
028_ch2_EP_ENP_RAF_B	12	2	65	4	1	6.0	0.74	8	6.5	2.00
029_EP_ENP_RAF_B	56	6	730	13	4	5.8	1.03	3	16.0	1.09
035_ch1_EP_ENP_RAF_B	25	3	128	8	2	5.5	1.33	2	2.5	1.34
035_ch2_EP_ENP_RAF_B	25	7	269	8	3	6.7	2.17	2	13.5	2.39
039_ch1_EP_ENP_RAF_B	340	2	1231	16	2	11.5	0.58	9	12.2	0.85
039_ch3_EP_ENP_RAF_B	120	3	1000	18	2	4.0	0.66	9	46.1	0.88
048_ch1_EP_ENP_RAF_B	254	16	591	15	6	3.0	0.90	19	56.9	0.70
048_ch2_EP_ENP_RAF_B	132	24	546	13	7	5.6	0.90	16	13.3	0.59
064_ch1_EP_ENP_RAF_B	193	21	825	15	11	9.6	0.89	3	10.3	0.97
064_ch2_EP_ENP_RAF_B	40	6	412	11	4	12.3	0.90	1	21.0	1.17
025_EP_ENP_RAF_X	236	26	996	20	6	4.3	0.94	14	16.6	1.20
028_CH1_EP_ENP_RAF_X	151	20	325	16	9	11.1	1.13	17	25.1	1.97
028_CH2_EP_ENP_RAF_X	11	2	65	5	1	6.0	0.86	5	5.8	0.90

Tableau 6.2 – Description des instances *EP_ENP*

<i>Instances EP_ENP</i>	Solution de départ	Algorithme original	Algorithme avec relaxation stochastique
024_38_3_EP_ENP_RAF_A	20.5191	0.0070	0.0058
024_38_5_EP_ENP_RAF_A	26.3079	0.0067	0.0056
025_38_1_EP_ENP_RAF_A	20.8112	0.8926	0.8027
048_39_1_EP_ENP_RAF_A	686.8711	0.2126	0.2457
022_EP_ENP_RAF_B	5.1856	0.0055	0.0119
023_EP_ENP_RAF_B	5.8750	0.0000	0.0000
024_V2_EP_ENP_RAF_B	1.1426	0.0272	0.0290
025_EP_ENP_RAF_B	1.0440	0.0041	0.0059
028_ch1_EP_ENP_RAF_B	1.2424	0.0001	0.0005
028_ch2_EP_ENP_RAF_B	1.0001	0.0000	0.0000
029_EP_ENP_RAF_B	1.6999	0.0000	0.0000
035_ch1_EP_ENP_RAF_B	1.0001	0.0000	0.0000
035_ch2_EP_ENP_RAF_B	1.0078	0.0000	0.0000
039_ch1_EP_ENP_RAF_B	2.0022	0.0033	0.0076
039_ch3_EP_ENP_RAF_B	52.0000	0.3401	0.2792
048_ch1_EP_ENP_RAF_B	5041.4400	0.2150	0.1750
048_ch2_EP_ENP_RAF_B	3.6664	0.0000	0.0000
064_ch1_EP_ENP_RAF_B	204.6648	0.3187	0.2637
064_ch2_EP_ENP_RAF_B	1.0147	0.0001	0.0001
025_EP_ENP_RAF_X	237.8402	0.0882	0.0450
028_CH1_EP_ENP_RAF_X	1.3075	0.1118	0.0006
028_CH2_EP_ENP_RAF_X	1000002.6667	0.0000	0.0000

Tableau 6.3 – Résultats pour les instances *EP_ENP*

6.2 Instances *EP_RAF*

Pour les 36 instances *EP_RAF*, le nombre de modèles M varie entre 12 et 340 pour une moyenne de 142. En considérant seulement le nombre de modèles associés à des contraintes de ratio prioritaire, M_{EP} , le nombre varie entre 1 et 30 pour une moyenne de 10. Si le nombre de modèles est assez peu élevé, il y a par contre entre 65 et 1319 véhicules pour une moyenne de 735 véhicules par instance.

Le nombre de contraintes de ratio prioritaires varie entre 1 et 11 et est en moyenne de 5. Par ailleurs, le nombre de contraintes de ratio non-prioritaires varie entre 0 et 19 et est en moyenne de 7. Le dénominateur moyen est de 9, 2 et 20, 5, pour les contraintes de ratio prioritaires et non-prioritaires, respectivement. La difficulté maximale peut aller jusqu'à 2, 17 et 4, 17, pour les contraintes de ratio prioritaires et non-prioritaires, respectivement. Enfin, le nombre de couleurs varie entre 4 et 25 pour une moyenne de 14.

Dans le tableau 6.5, nous présentons les résultats obtenus sur les instances *EP_RAF*. Le format de ce tableau est identique à celui du tableau 6.3. Pour les 36 instances *EP_RAF*, nous obtenons la meilleure solution de premier niveau pour 34 instances, la meilleure solution de premier et deuxième niveau pour 5 instances, et la meilleure solution en pour les trois niveaux d'objectifs pour 3 instances. On observe un écart moyen de 7,3% avec la meilleure solution. Il y a 13 instances où l'écart est inférieur à 1% et 7 instances additionnelles où l'écart est dans l'ordre de 2 – 3%.

<i>Instances EP_RAF</i>	<i>M</i>	<i>M_{EP}</i>	<i>N</i>	<i>N_C</i>	<i>EP</i>			<i>ENP</i>		
					<i>#</i>	<i>Den</i>	<i>Di^m</i>	<i>#</i>	<i>Den</i>	<i>Di^m</i>
022_3_4_EP_RAF_ENP_A	59	5	485	13	3	4.3	0.85	6	7.2	0.33
024_38_3_EP_RAF_ENP_A	238	8	1260	14	5	6.4	0.96	8	6.8	0.95
024_38_5_EP_RAF_ENP_A	215	8	1315	14	5	6.4	0.96	8	6.8	0.96
025_38_1_EP_RAF_ENP_A	288	11	1004	25	4	4.5	0.75	18	22.3	1.12
039_38_4_EP_RAF_ch1_A	114	20	954	15	5	16.0	1.03	0	0.0	0.00
048_39_1_EP_RAF_ENP_A	230	17	600	13	5	4.4	0.99	12	8.1	0.97
064_38_2_EP_RAF_ENP_ch1_A	134	15	875	15	7	10.0	0.71	2	4.5	1.83
064_38_2_EP_RAF_ENP_ch2_A	38	6	335	13	4	36.5	0.96	2	11.5	0.87
022_EP_RAF_ENP_B	80	4	526	14	2	4.5	0.86	7	6.4	1.00
023_EP_RAF_ENP_B	147	19	1110	14	9	5.2	1.05	8	2.5	0.54
024_V2_EP_RAF_ENP_B	262	11	1270	14	6	12.5	1.36	7	29.6	1.26
025_EP_RAF_ENP_B	254	11	1161	20	4	4.5	0.81	12	14.1	4.17
028_ch1_EP_RAF_ENP_B	144	17	365	21	9	8.7	1.14	11	7.9	0.95
028_ch2_EP_RAF_ENP_B	12	2	65	4	1	6.0	0.74	8	6.5	2.00
029_EP_RAF_ENP_B	56	6	730	13	4	5.8	1.03	3	16.0	1.09
035_ch1_EP_RAF_ENP_B	25	3	128	8	2	5.5	1.33	2	2.5	1.34
035_ch2_EP_RAF_ENP_B	25	7	269	8	3	6.7	2.17	2	13.5	2.39
039_ch1_EP_RAF_ENP_B	340	2	1231	16	2	11.5	0.58	9	12.2	0.85
039_ch3_EP_RAF_ENP_B	120	3	1000	18	2	4.0	0.66	9	46.1	0.88
048_ch1_EP_RAF_ENP_B	254	16	591	15	6	3.0	0.90	19	56.9	0.70
048_ch2_EP_RAF_ENP_B	132	24	546	13	7	5.6	0.90	16	13.3	0.59
064_ch1_EP_RAF_ENP_B	193	21	825	15	11	9.6	0.89	3	10.3	0.97
064_ch2_EP_RAF_ENP_B	40	6	412	11	4	12.3	0.90	1	21.0	1.17
023_EP_RAF_ENP_X	203	17	1260	14	5	9.8	0.87	7	5.0	0.81
024_EP_RAF_ENP_X	276	15	1319	16	7	5.6	0.99	11	8.9	0.66
029_EP_RAF_ENP_X	74	6	780	15	4	5.3	0.69	3	16.0	0.98
034_VP_EP_RAF_ENP_X	65	4	921	11	3	12.0	0.94	5	18.2	1.32
034_VU_EP_RAF_ENP_X	25	6	231	8	6	9.0	1.10	2	9.0	1.13
039_CH1_EP_RAF_ENP_X	329	1	1247	16	1	20.0	0.00	11	38.5	0.84
039_CH3_EP_RAF_ENP_X	157	3	1037	18	2	4.0	0.76	10	53.4	0.88
048_CH1_EP_RAF_ENP_X	210	16	519	14	6	3.0	0.93	16	31.4	3.46
048_CH2_EP_RAF_ENP_X	142	30	459	14	8	5.5	1.07	12	16.0	0.64
064_CH1_EP_RAF_ENP_X	157	19	875	14	9	19.0	1.07	2	7.0	0.86
064_CH2_EP_RAF_ENP_X	43	10	273	13	6	23.2	0.69	0	0.0	0.00
655_CH1_EP_RAF_ENP_X	20	2	264	10	4	5.3	0.09	1	5.0	0.11
655_CH2_EP_RAF_ENP_X	19	5	219	8	4	8.5	1.69	0	0.0	0.00

Tableau 6.4 – Description des instances *EP_RAF*

<i>Instances EP_RAF</i>	Solution de départ	Algorithme original	Algorithme avec relaxation stochastique
022_3_4_EP_RAF_ENP_A	66.7721	0.2580	0.1613
024_38_3_EP_RAF_ENP_A	17.2657	0.0144	0.0137
024_38_5_EP_RAF_ENP_A	23.0061	0.0117	0.0126
025_38_1_EP_RAF_ENP_A	1.2637	0.0698	0.0448
039_38_4_EP_RAF_ch1_A	9.9192	0.0771	0.0779
048_39_1_EP_RAF_ENP_A	201.4917	0.0863	0.0808
064_38_2_EP_RAF_ENP_ch1_A	19.4733	0.1686	0.2041
064_38_2_EP_RAF_ENP_ch2_A	823.6484	0.0002	0.0588
022_EP_RAF_ENP_B	2.9324	0.6803	0.4181
023_EP_RAF_ENP_B	5.6374	0.0008	0.0010
024_V2_EP_RAF_ENP_B	1.1555	0.0318	0.0271
025_EP_RAF_ENP_B	13.1288	0.2935	0.2335
028_ch1_EP_RAF_ENP_B	1.1484	0.0002	0.0001
028_ch2_EP_RAF_ENP_B	1.9828	0.0000	0.0000
029_EP_RAF_ENP_B	1.9112	0.0002	0.0001
035_ch1_EP_RAF_ENP_B	1.0450	0.0000	0.0000
035_ch2_EP_RAF_ENP_B	1.0312	0.0000	0.0000
039_ch1_EP_RAF_ENP_B	1.3866	0.1228	0.0534
039_ch3_EP_RAF_ENP_B	1.1276	0.0382	0.0163
048_ch1_EP_RAF_ENP_B	7.6438	0.0631	0.0440
048_ch2_EP_RAF_ENP_B	4.5675	0.0068	0.0065
064_ch1_EP_RAF_ENP_B	32.1017	0.2544	0.2077
064_ch2_EP_RAF_ENP_B	1.0773	0.0077	0.0308
023_EP_RAF_ENP_X	32.3647	0.0551	0.0604
024_EP_RAF_ENP_X	194.1361	0.2167	0.2196
029_EP_RAF_ENP_X	1.5685	0.1793	0.0888
034_VP_EP_RAF_ENP_X	376.5996	0.1025	0.0313
034_VU_EP_RAF_ENP_X	8.2975	0.0005	0.0002
039_CH1_EP_RAF_ENP_X	1.2785	0.0015	0.0021
039_CH3_EP_RAF_ENP_X	5.4203	0.0263	0.0221
048_CH1_EP_RAF_ENP_X	36.5636	0.0208	0.0308
048_CH2_EP_RAF_ENP_X	2.0308	0.0004	0.0004
064_CH1_EP_RAF_ENP_X	1.2133	0.0006	0.0008
064_CH2_EP_RAF_ENP_X	1.3784	0.1081	0.0811
655_CH1_EP_RAF_ENP_X	2.2333	0.0000	0.0000
655_CH2_EP_RAF_ENP_X	1.0198	0.0000	0.0000

Tableau 6.5 – Résultats pour les instances *EP_RAF*

6.3 Instances *RAF_EP*

Pour les 22 instances *RAF_EP* le nombre de modèles M varie entre 12 et 340 pour une moyenne de 116 par instance. En considérant seulement le nombre de modèles associés à des contraintes de ratio prioritaires, M_{EP} , le nombre varie entre 2 et 24 pour une moyenne de 10. Si le nombre de modèles est assez peu élevé, il y a par contre entre 65 et 1270 véhicules par instance, pour une moyenne de 639.

Le nombre de contraintes de ratio prioritaires varie entre 1 et 11 et est en moyenne de 4. Par ailleurs, le nombre de contraintes de ratio non-prioritaires varie entre 0 et 57 et est en moyenne de 13. Le dénominateur moyen est de 9,0 et 19,6, pour les contraintes de ratio prioritaires et non-prioritaires, respectivement. La difficulté maximale peut aller jusqu'à 2,17 et 4,17, pour les contraintes de ratio prioritaires et non-prioritaires, respectivement. Enfin, le nombre de couleurs varie entre 4 et 21 pour une moyenne de 13.

Dans le tableau 6.7, nous présentons les résultats obtenus sur les instances *RAF_EP*. Le format du tableau 6.7 est identique à celui du tableau 6.3. Pour les 22 instances *RAF_EP*, nous obtenons la solution optimale de premier niveau pour toutes les instances, la meilleure solution de premier et deuxième niveau pour 10 instances et la meilleure solution pour les trois niveaux d'objectifs pour 4 instances. On observe un écart moyen de seulement 0,026% avec la meilleure solution.

<i>Instances RAF_EP</i>	<i>M</i>	<i>M_{EP}</i>	<i>N</i>	<i>N_C</i>	<i>EP</i>			<i>ENP</i>		
Instance					#	<i>Den</i>	<i>Di^m</i>	#	<i>Den</i>	<i>Di^m</i>
022_3_4_RAF_EP_ENP_A	59	5	485	13	3	4.3	0.85	6	7.2	0.33
039_38_4_RAF_EP_ch1_A	114	20	954	15	5	16.0	1.03	0	0.0	0.00
064_38_2_RAF_EP_ENP_ch1_A	134	15	875	15	7	10.0	0.71	2	4.5	1.83
064_38_2_RAF_EP_ENP_ch2_A	38	6	335	13	4	36.5	0.96	2	11.5	0.87
022_RAF_EP_ENP_B	80	4	526	14	2	4.5	0.86	7	6.4	1.00
023_RAF_EP_ENP_B	147	19	1110	14	9	5.2	1.05	8	2.5	0.54
024_V2_RAF_EP_ENP_B	262	11	1270	14	6	12.5	1.36	7	29.6	1.26
025_RAF_EP_ENP_B	254	11	1161	20	4	4.5	0.81	12	14.1	4.17
028_ch1_RAF_EP_ENP_B	144	17	365	21	9	8.7	1.14	11	7.9	0.95
028_ch2_RAF_EP_ENP_B	12	2	65	4	1	6.0	0.74	8	6.5	2.00
029_RAF_EP_ENP_B	56	6	730	13	4	5.8	1.03	3	16.0	1.09
035_ch1_RAF_EP_ENP_B	25	3	128	8	2	5.5	1.33	2	2.5	1.34
035_ch2_RAF_EP_ENP_B	25	7	269	8	3	6.7	2.17	2	13.5	2.39
039_ch1_RAF_EP_ENP_B	340	2	1231	16	2	11.5	0.58	9	12.2	0.85
039_ch3_RAF_EP_ENP_B	120	3	1000	18	2	4.0	0.66	9	46.1	0.88
048_ch1_RAF_EP_ENP_B	254	16	591	15	6	3.0	0.90	19	56.9	0.70
048_ch2_RAF_EP_ENP_B	132	24	546	13	7	5.6	0.90	16	13.3	0.59
064_ch1_RAF_EP_ENP_B	193	21	825	15	11	9.6	0.89	3	10.3	0.97
064_ch2_RAF_EP_ENP_B	40	6	412	11	4	12.3	0.90	1	21.0	1.17
022_RAF_EP_ENP_X	85	6	704	14	3	4.0	0.63	9	5.9	0.68
035_CH1_RAF_EP_X	12	2	90	7	1	2.0	1.12	0	0.0	0.00
035_CH2_RAF_EP_X	20	3	376	8	2	2.0	0.94	0	0.0	0.00

Tableau 6.6 – Description des instances *RAF_EP*

<i>Instances RAF_EP</i>	Solution de départ	Algorithme original	Algorithme avec relaxation stochastique
022_3_4_RAF_EP_ENP_A	1.0008	0.0001	0.0002
039_38_4_RAF_EP_ch1_A	1.0181	0.0014	0.0015
064_38_2_RAF_EP_ENP_ch1_A	1.0164	0.0002	0.0002
064_38_2_RAF_EP_ENP_ch2_A	1.1034	0.0000	0.0000
022_RAF_EP_ENP_B	1.0001	0.0000	0.0000
023_RAF_EP_ENP_B	1.0208	0.0005	0.0006
024_V2_RAF_EP_ENP_B	1.0103	0.0016	0.0014
025_RAF_EP_ENP_B	1.0082	0.0002	0.0001
028_ch1_RAF_EP_ENP_B	1.0264	0.0000	0.0000
028_ch2_RAF_EP_ENP_B	2.0000	0.0000	0.0000
029_RAF_EP_ENP_B	1.0202	0.0003	0.0004
035_ch1_RAF_EP_ENP_B	1.0010	0.0000	0.0002
035_ch2_RAF_EP_ENP_B	1.0004	0.0001	0.0001
039_ch1_RAF_EP_ENP_B	1.2725	0.0000	0.0000
039_ch3_RAF_EP_ENP_B	1.0169	0.0000	0.0000
048_ch1_RAF_EP_ENP_B	1.0160	0.0001	0.0001
048_ch2_RAF_EP_ENP_B	1.0178	0.0001	0.0001
064_ch1_RAF_EP_ENP_B	1.0168	0.0003	0.0002
064_ch2_RAF_EP_ENP_B	1.0322	0.0000	0.0000
022_RAF_EP_ENP_X	1.0833	0.0000	0.0000
035_CH1_RAF_EP_X	1.1996	0.0000	0.0002
035_CH2_RAF_EP_X	1.1651	0.0000	0.0000

Tableau 6.7 – Résultats pour les instances *RAF_EP*

6.4 Étude de sensibilité des paramètres

En étudiant l'écart par rapport à la meilleure solution connue, on peut observer que les paramètres les plus importants sont la taille des instances et le nombre de modèles de véhicules associé à des contraintes de ratio prioritaires, M_{EP} . Par ailleurs, une analyse selon le nombre de rafales de peinture n'est pas pertinente puisque ce nombre ne varie presque pas d'une instance à l'autre.

Dans les tableaux 6.8, 6.9 et 6.10, nous présentons les résultats obtenus selon la taille des instances. La première colonne présente les différentes catégories pour la taille, soit 0 – 500 véhicules, 500 – 1000 véhicules et 1000 – 1500 véhicules. Dans les quatre colonnes suivantes, on retrouve le pourcentage des instances où l'écart par rapport à la meilleure solution connue se situe sous un certain seuil, soit 5%, 1%, 0.1%, 0.01% pour les problèmes EP_{ENP} et EP_{RAF} et 0.05%, 0.01%, 0.001%, 0.0001%, pour les problèmes de type RAF_{EP} . La dernière colonne contient le nombre d'instances dans chaque catégorie.

Lorsque la taille des instances augmente, on observe que l'écart avec la meilleure solution connue tend également à augmenter. Il y a toutefois une anomalie dans les tableaux 6.8, 6.9, puisque les résultats sont meilleurs pour la catégorie 1000 – 1500 véhicules que pour la catégorie 500 – 1000. En fait, ceci s'explique par le fait que les instances avec 500 à 1000 véhicules ont en moyenne un nombre de modèles de véhicules avec contraintes de ratio prioritaires plus élevé que les instances avec 1000 à 1500 véhicules (voir plus bas).

Nombre de véhicules	Écart de la meilleure solution				Nombre d'instances
	<5%	<1%	<0.1%	<0.01%	
0-500	100%	100%	100%	57%	7
500-1000	57%	43%	29%	29%	7
500-1500	88%	63%	13%	13%	8

Tableau 6.8 – Analyse des résultats selon la taille des instances EP_{ENP}

Nombre de véhicules	Écart de la meilleure solution				Nombre d'instances
	<5%	<1%	<0.1%	<0.01%	
0-500	83%	67%	67%	42%	12
500-1000	50%	25%	17%	0%	12
500-1500	75%	17%	0%	0%	12

Tableau 6.9 – Analyse des résultats selon la taille des instances EP_{RAF}

Nombre de véhicules	Écart de la meilleure solution				Nombre d'instances
	<0.05%	<0.01%	<0.001%	<0.0001%	
0-500	100%	89%	67%	67%	9
500-1000	88%	38%	25%	25%	8
500-1500	80%	40%	20%	0%	5

Tableau 6.10 – Analyse des résultats selon la taille pour les instances *RAF_EP*

Dans les tableaux 6.11, 6.12 et 6.13, nous présentons les résultats obtenus en fonction du nombre de modèles de véhicules avec contraintes de ratio prioritaires M_{EP} . Les tableaux ont la même structure que les tableaux 6.8, 6.9 et 6.10, sauf pour la première colonne qui contient maintenant les différents catégories d'instances en fonction de M_{EP} . Nous avons choisi les trois catégories de façon à obtenir une répartition uniforme des instances.

Lorsque M_{EP} augmente, l'écart avec la meilleure solution connue tend à augmenter aussi. Comme M_{EP} est un facteur qui semble avoir un impact particulièrement important, son intégration au sein de notre estimateur de difficulté d'une instance pourrait mener à une amélioration de nos résultats.

Nombre de modèles M_{EP}	Écart de la meilleure solution				Nombre d'instances
	<5%	<1%	<0.1%	<0.01%	
2-6	88%	75%	63%	50%	8
7-16	71%	57%	14%	14%	7
17-24	71%	57%	43%	29%	7

Tableau 6.11 – Analyse des résultats pour les instances *EP_ENP* selon le nombre de modèles M_{EP}

Nombre de modèles M_{EP}	Écart de la meilleure solution				Nombre d'instances
	<5%	<1%	<0.1%	<0.01%	
1-5	73%	73%	45%	36%	11
6-14	67%	25%	25%	8%	12
15-30	54%	38%	31%	0%	13

Tableau 6.12 – Analyse des résultats pour les instances *EP_RAF* selon le nombre de modèles M_{EP}

Nombre de modèles M_{EP}	Écart de la meilleure solution				Nombre d'instances
	<0.05%	<0.01%	<0.001%	<0.0001%	
2-5	100%	63%	63%	38%	8
6-11	86%	43%	29%	29%	7
11-24	71%	14%	14%	14%	7

Tableau 6.13 – Analyse des résultats pour les instances RAF_{EP} selon le nombre de modèles M_{EP}

6.5 Évolution moyenne de la solution dans le temps

Les graphiques suivants montrent l'évolution de la qualité des solutions avec le temps de calcul. Cette évolution correspond à une moyenne sur toutes les instances d'une classe de problèmes, soit EP_{ENP} , EP_{RAF} et RAF_{EP} .

6.5.1 Graphiques sur l'évolution de la meilleure solution

La valeur qu'on observe dans les Figures 6.1, 6.3 et 6.5 est :

$$\frac{\text{Valeur de la meilleure solution rencontrée au temps } t - \text{Valeur de la meilleure solution}}{\text{Valeur de la solution de départ} - \text{Valeur de la meilleure solution}}$$

Nous présentons l'évolution de l'algorithme original ainsi que celle de l'algorithme avec relaxation stochastique. Cette évolution est assez semblable, entre autres parce qu'au début la phase de construction est la même pour les deux algorithmes. Il faut d'ailleurs remarquer qu'une partie très importante de l'optimisation se fait en une dizaine de secondes, grâce aux heuristiques de construction et aux premières recherches tabou.

À long terme, l'algorithme avec relaxation stochastique produit les meilleures solutions. Il optimise de façon plus constante à tous les niveaux, ce qui le rend plus stable. En plus, il semble s'adapter mieux aux différentes caractéristiques d'une instance (nombre de véhicules, nombre de contraintes de ratio et valeur courante de l'objectif).

A la droite des graphiques, le petit carré noir correspond à la valeur moyenne des solutions de départ, le petit carré rouge à la valeur moyenne des solutions obtenues par l'équipe qui s'est classée au dernier rang et le petit losange noir à la valeur moyenne des solutions obtenues par la meilleure équipe.

6.5.2 Graphiques sur l'évolution de la solution courante

La valeur qu'on observe dans les Figures 6.2, 6.4 et 6.6 est :

$$\frac{\text{Valeur de la solution courante au temps } t - \text{Valeur de la meilleure solution}}{\text{Valeur de la solution de départ} - \text{Valeur de la meilleure solution}}$$

Les graphiques représentent le niveau des modifications à la valeur de l'objectif pendant l'évolution de l'algorithme. Les pics élevés correspondent généralement à une période de diversification, tandis que les vallées correspondent à une période d'intensification de la recherche tabou.

La solution courante varie beaucoup, parfois sur des intervalles de temps très courts, ce qui nuit à la clarté des graphiques. En conséquence, la valeur moyenne (moyenne mobile) des solutions visitées a été ajoutée sur les graphiques.

Figure 6.1 - Évolution moyenne de la meilleure solution rencontrée sur les instances EP_ENP

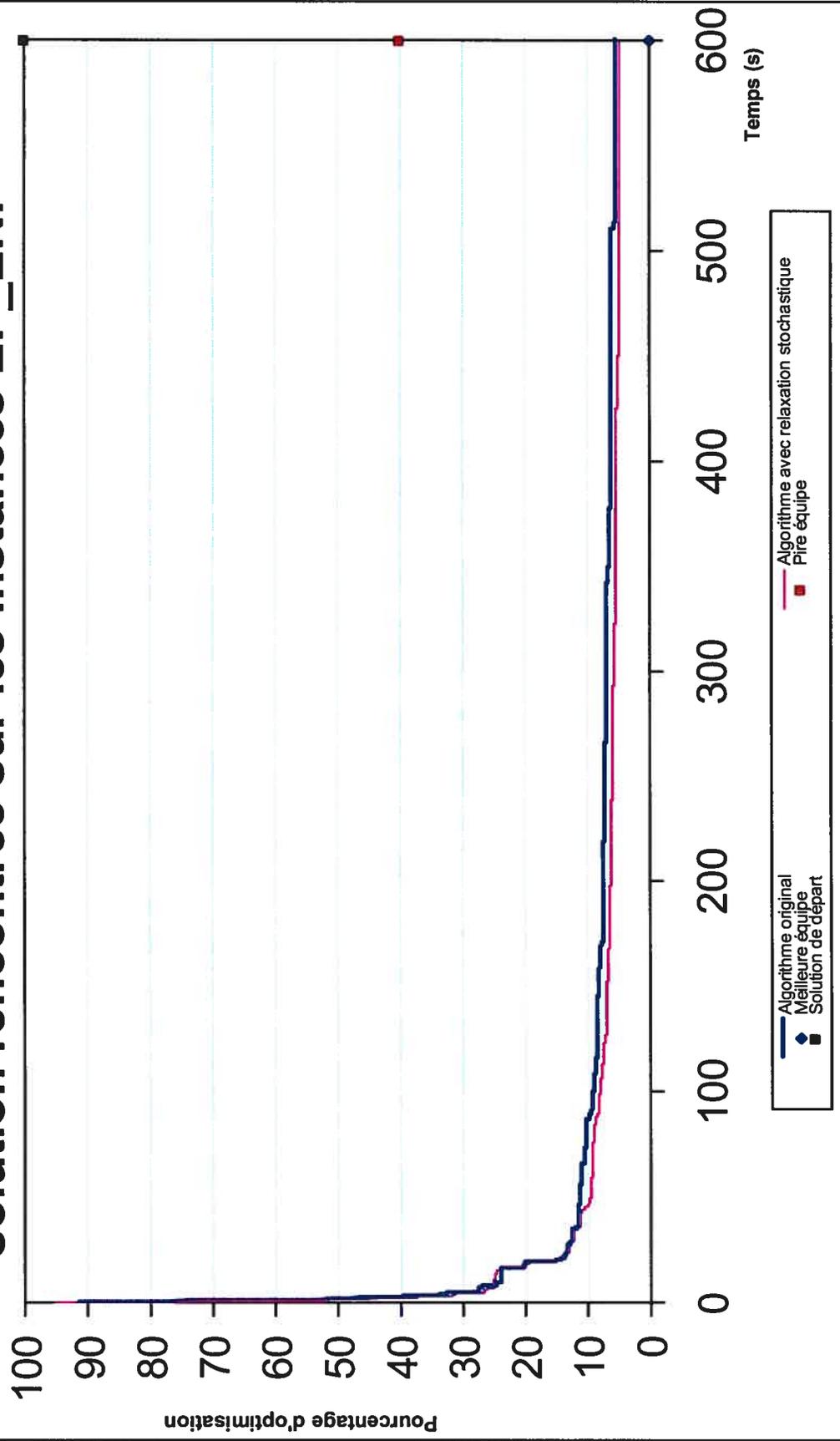


Figure 6.2 - Évolution moyenne de la solution courante sur les instances EP_ENP

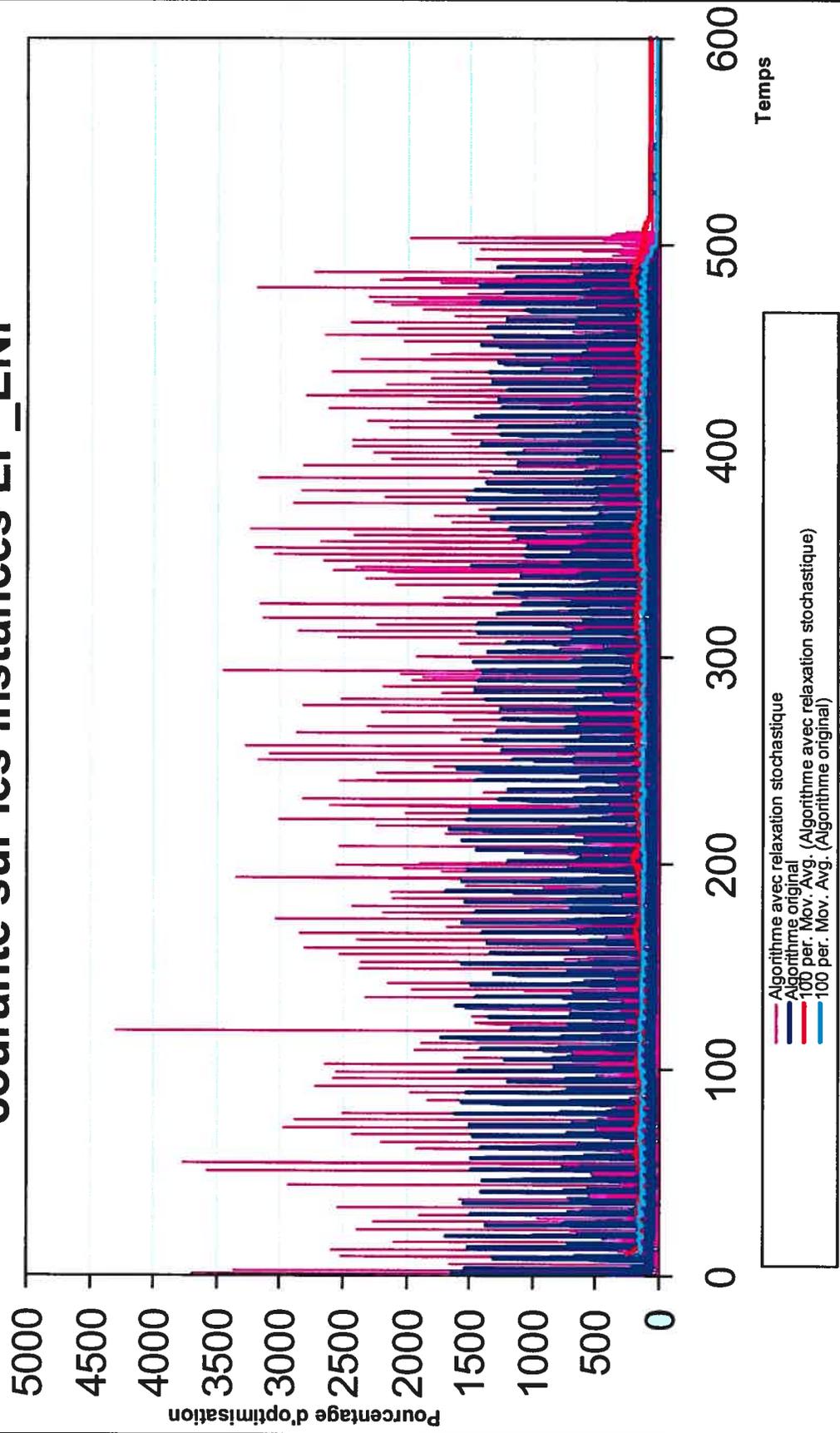


Figure 6.3 - Évolution moyenne de la meilleure solution rencontrée sur les instances EP_RAF

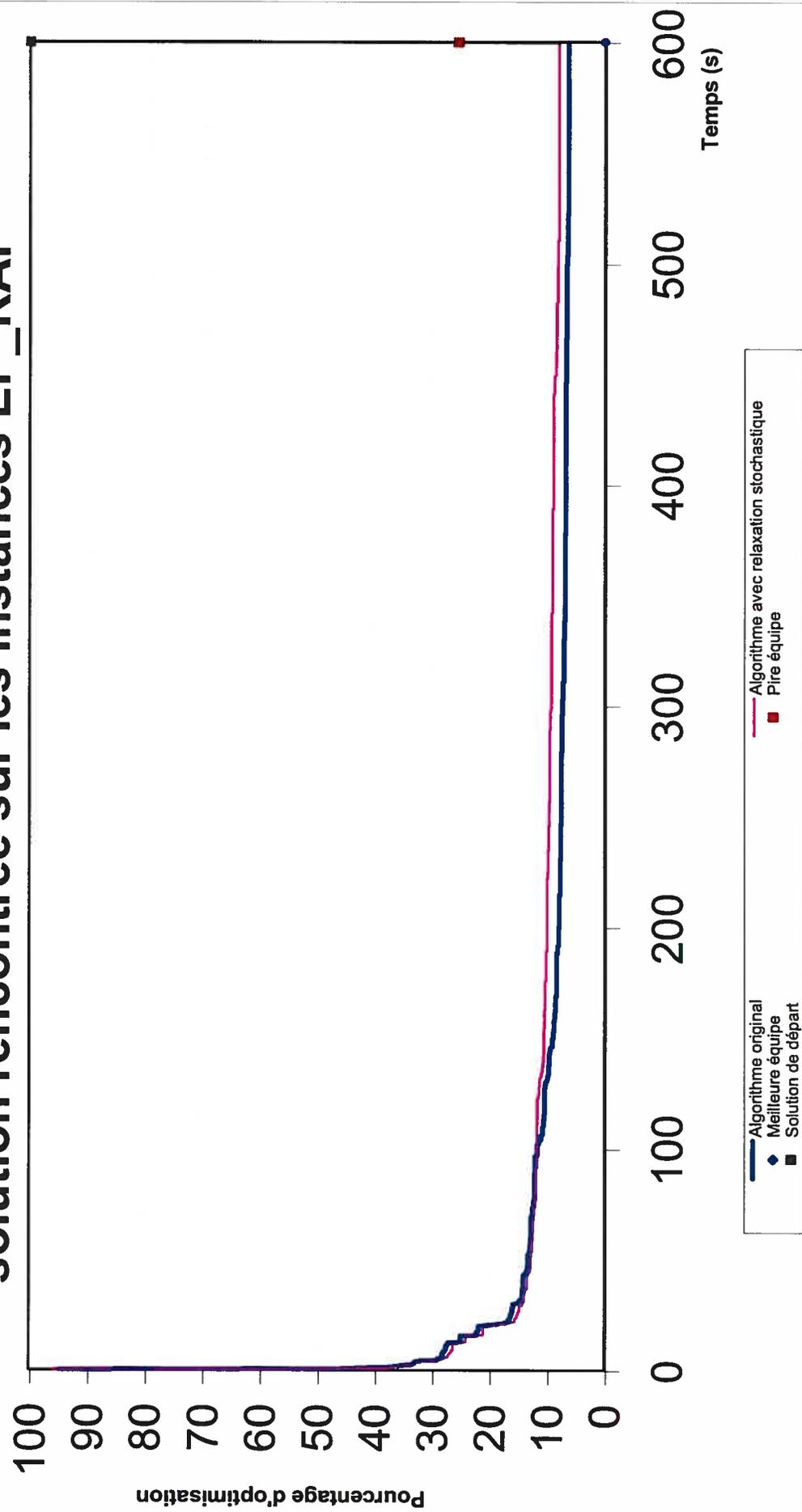


Figure 6.4 - Évolution moyenne de la solution courante sur les instances EP_RAF

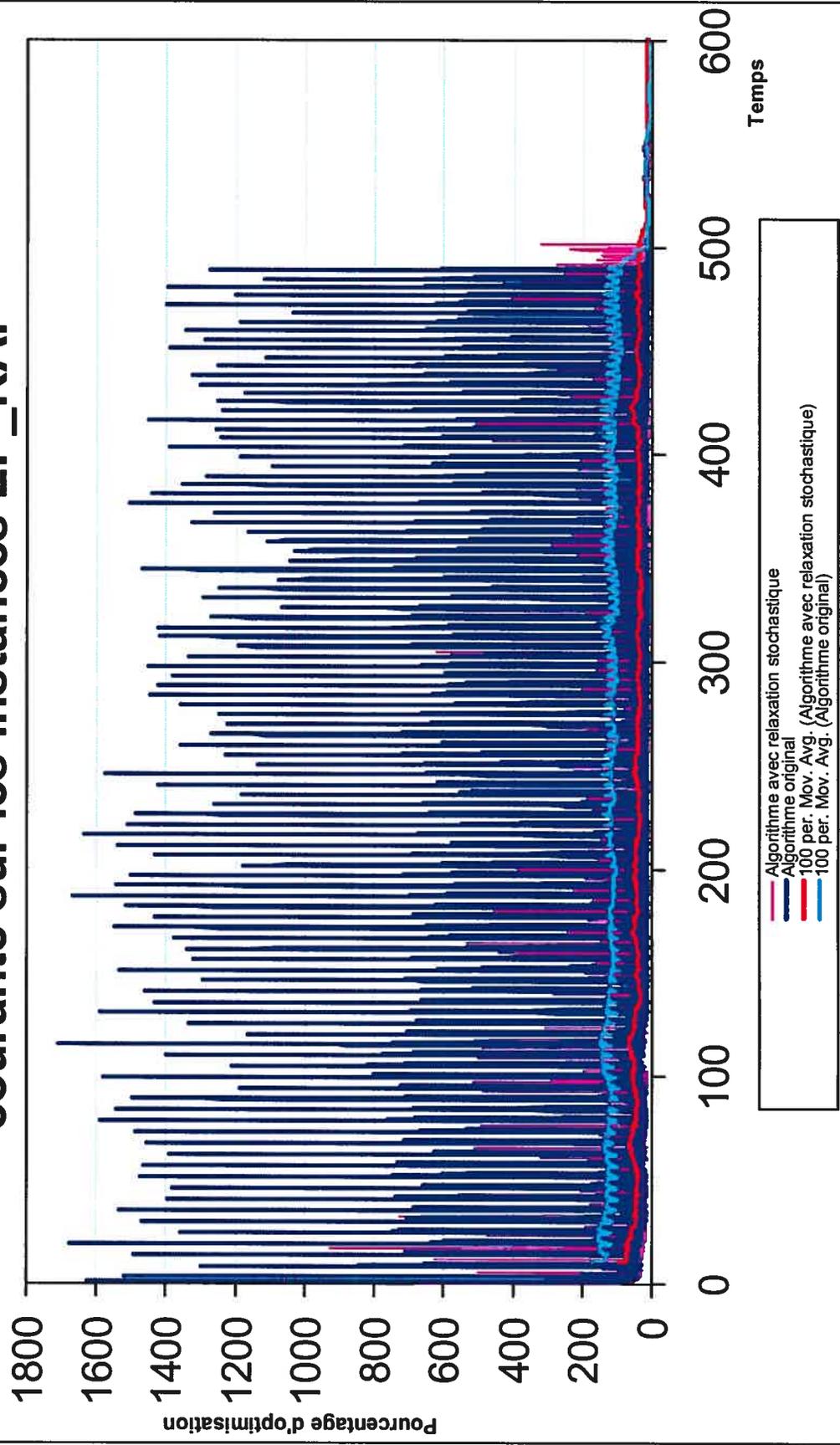


Figure 6.5 - Évolution moyenne de la meilleure solution rencontrée sur les instances RAF_EP

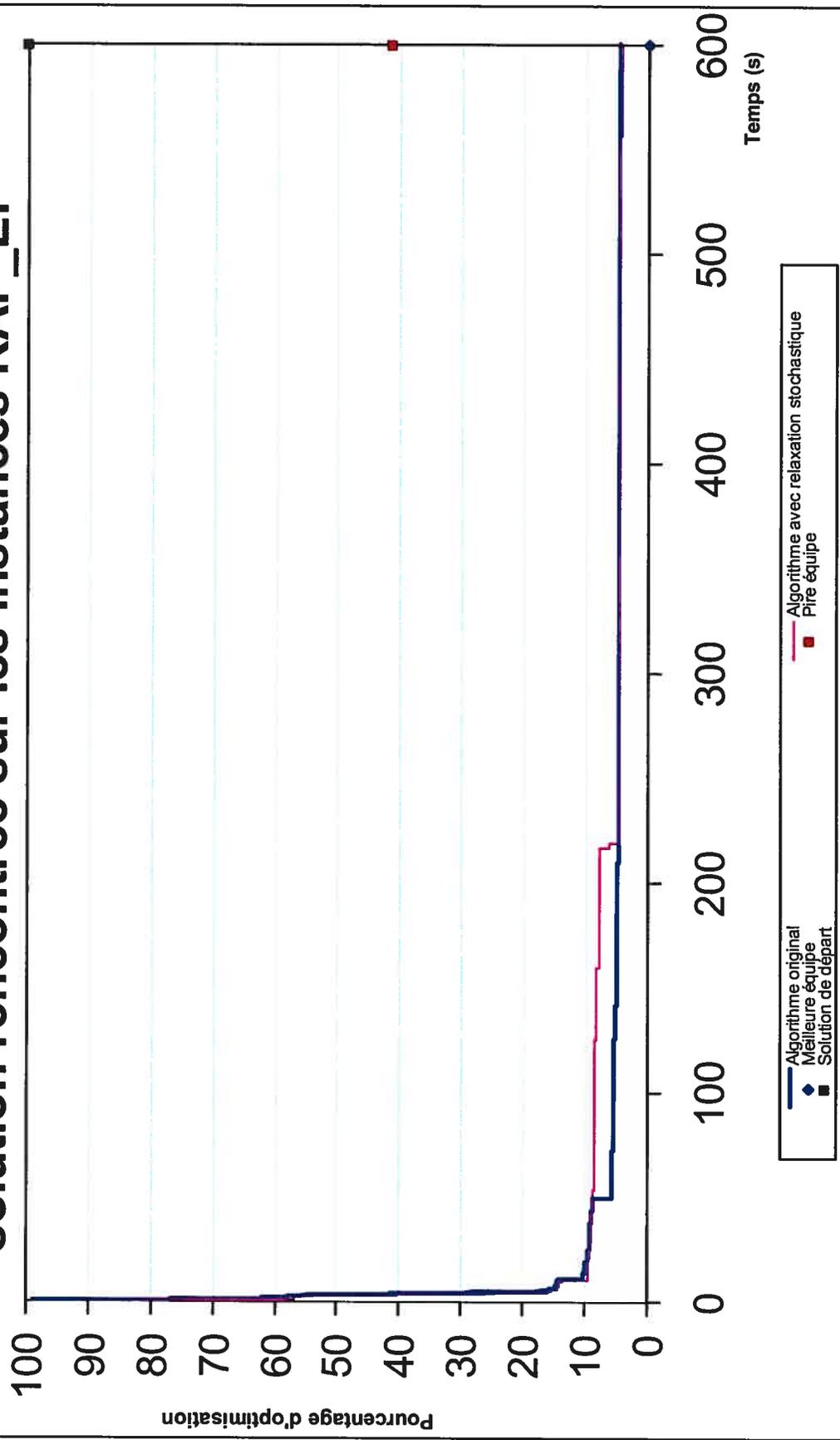
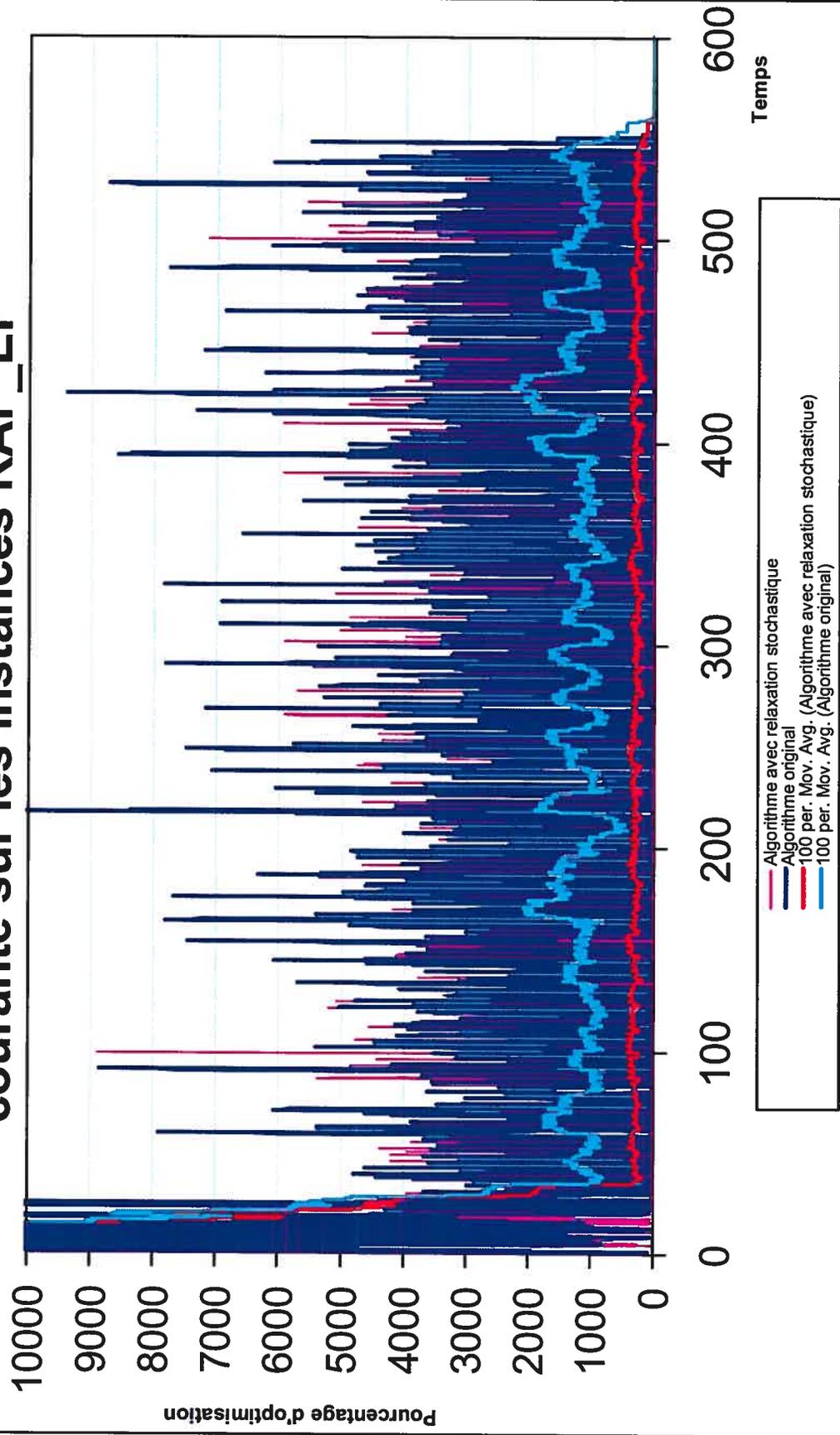


Figure 6.6 - Évolution moyenne de la solution courante sur les instances RAF_EP



CHAPITRE 7

CONCLUSIONS

Dans ce mémoire, nous avons présenté une approche de résolution basée sur la recherche tabou pour résoudre le problème d'ordonnement de véhicules sur une ligne de production. Ce travail a été réalisé en partie dans le cadre du Challenge ROADEF 2005, alors que certains raffinements et améliorations ont été apportés par la suite. L'efficacité de notre algorithme repose sur quelques concepts importants.

Les heuristiques de construction permettent de produire de bonnes solutions de départ dans des temps très courts, ce qui facilite beaucoup le travail de la recherche tabou. Ces heuristiques tiennent compte, en particulier, du coût futur d'une insertion ainsi que des mesures de corrélation entre les contraintes de ratio, ce qui les rend très performantes et leur permet de s'approcher ou même de trouver l'optimum pour l'objectif de premier niveau. En fait, dans le cas où l'objectif de premier niveau correspond aux rafales de peinture, l'heuristique de construction développée permet toujours de produire une solution optimale de premier niveau.

La recherche tabou est appliquée avec différentes valeurs pour les paramètres en fonction des caractéristiques de l'instance (taille, nombre de contraintes de ratio, objectifs, valeur de la solution courante, valeur de la meilleure solution connue, ...). La modification des valeurs des paramètres permet également d'intensifier ou de diversifier la recherche.

Une sélection soignée des véhicules qui sont impliqués dans un mouvement permet d'obtenir une convergence rapide. Cette sélection est de nature stochastique mais est biaisée par le nombre de violations de contraintes associées à un véhicule.

L'utilisation de valeurs pré-calculées permet une évaluation très rapide du voisinage. La complexité d'une itération de la recherche tabou est ainsi réduite d'un ordre de grandeur pour les instances de la ROADEF.

La relaxation des objectifs est particulièrement cruciale pour une bonne performance. Pendant le challenge, nous avons utilisé la relaxation séquentielle. L'approche est plus efficace que la recherche tabou sans relaxation, mais elle s'est par contre montrée un peu trop sensible aux valeurs de paramètres. Nous avons donc développé une approche stochastique plus robuste. Dans cette approche, un objectif peut être actif à un moment et relaxé plus tard. Chaque fois que la recherche tabou est redémarrée, les objectifs actifs sont choisis de façon stochastique.

Pendant le challenge ROADEF'05, notre voisinage principal était le voisinage par échange. Nous avons ajouté le voisinage par extraction\insertion par la suite. De façon générale, l'ajout d'un nouveau voisinage améliore la robustesse et la performance de l'algorithme, mais le gain marginal lié à l'implémentation d'un nouveau voisinage devient de moins en moins grand. Nous avons observé que l'utilisation individuelle de chaque voisinage, où l'on choisit soit le voisinage par échange soit celui par extraction\insertion,

est moins efficace que l'utilisation conjointe des deux voisinages.

BIBLIOGRAPHIE

- [1] Cordeau J.-F., Laporte G. et Pasin F. "Iterated Tabu Search for the Car Sequencing Problem". *Technical Report, CRT-2005-37, soumis à EJOR*.
- [2] Craciunas D., Gendreau M., Potvin J.-Y. "Approches heuristiques pour le séquençement de véhicules". CONFERENCE ROADEF'05, Tours, France, 2005.
- [3] Estellon B., Gardi F., Nouioua K. "Real-life Car Sequencing : Very Large Neighborhood Search vs Very Fast Local Search" . *soumis à EJOR, 2005*.
- [4] Estellon B., Gardi F., Nouioua K. "Large Neighborhood Improvements for Solving Car Sequencing Problems". *RAIRO 40*, pages 355–379, 2006.
- [5] French OR Society ROADEF. "Challenge ROADEF", 2005. www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2005/.
- [6] Gendreau M. et Potvin J.-Y. "*Search Methodologies : Introductory Tutorials in Optimization and Decision Support Techniques, Ch. 6 : Recherche Tabou*". Springer , 2005.
- [7] Gent I. P. "Two results on Car-sequencing Problems". *Report APES-02-1998, Univ. of St. Andrews*, 1998.
- [8] Giard V. et Jeunet J. "Modélisation du problème général d'ordonnancement de véhicules sur une ligne de production et d'assemblage". *Annales du Lamsade 2*, pages 235–256, 2004.
- [9] Glover F. "Future Paths for Integer Programming and Lines to Artificial Intelligence". *Computers and Operations Research 13*, pages 533–549, 1986.
- [10] Gottlieb J., Puchta M. et Solnon C. "A Study of Greedy, Local search, and Ant Colony Optimization Approaches for Car Sequencing Problems". *EvoWorkshops 2003, Lecture Notes in Computer Science 2611*, 2003.
- [11] Gravel M., Gagné C. et Price W. "Review and Comparison of Three Method for the Solution of Car Sequencing Problem". *Journal of Operational Research Society 56*, pages 1287–1295, 2005.
- [12] Nguyen A. et Cung V. "Le problème du Car Sequencing RENAULT et le Challenge ROADEF'2005". *Premières Journées Francophones de Programmation par Contraintes* , pages 3–10, 2005.
- [13] Prandtstetter M. "Exact and Heuristic Methods for Solving the Car Sequencing Problem". *Master's thesis, Vienna University of Technology, Institute of Computer Graphics and Algorithms*, 2005.

- [14] Ribeiro C., Aloise D., Noronha T., Rocha C., Urrutia S. “A Hybrid Heuristic for a Real-Life Car Sequencing Problem with Painting and Assembly Line Constraints”. *Les Cahiers du GERAD 2006-02*, 2006.
- [15] Risler M., Chiarandini M., Paquete L., Schiavinotto T. et Stutzle T. “An algorithm for the car sequencing problem of the ROADEF 2005 Challenge”. *Technical Report AIDA-04-06, Fachgebiet Intellektik, Fachbereich Informatik, Technische Universität Darmstadt*, 2005.
- [16] Solnon C., Cung V., Nguyen A. et Artigues C. “The Car Sequencing Problem : Overview of State-of-the-art methods and industrial case-study of the ROADEF’2005 challenge problem”. *soumis à EJOR*, 2005.