

**Titre:** Ordonnancement et routage sans conflit de véhicules autoguidés  
Title: dans un atelier de fabrication flexible : approche heuristique

**Auteur:** Amira Dems  
Author:

**Date:** 2007

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Dems, A. (2007). Ordonnancement et routage sans conflit de véhicules  
Citation: autoguidés dans un atelier de fabrication flexible : approche heuristique [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie.  
<https://publications.polymtl.ca/8044/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/8044/>  
PolyPublie URL:

**Directeurs de recherche:** Louis-Martin Rousseau, & Jean-Yves Potvin  
Advisors:

**Programme:** Non spécifié  
Program:

UNIVERSITÉ DE MONTRÉAL

ORDONNANCEMENT ET ROUTAGE SANS CONFLIT DE VÉHICULES  
AUTOGUIDÉS DANS UN ATELIER DE FABRICATION FLEXIBLE :  
APPROCHE HEURISTIQUE

AMIRA DEMS

DÉPARTEMENT DE MATHÉMATIQUES ET DE GÉNIE INDUSTRIEL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(MATHÉMATIQUES APPLIQUÉES)

AOÛT 2007



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 978-0-494-35677-7*  
*Our file* *Notre référence*  
*ISBN: 978-0-494-35677-7*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

ORDONNANCEMENT ET ROUTAGE SANS CONFLIT DE VÉHICULES  
AUTOGUIDÉS DANS UN ATELIER DE FABRICATION FLEXIBLE :  
APPROCHE HEURISTIQUE

présenté par : DEMS Amira

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de :

M. DESAULNIERS Guy, Ph.D., président

M. ROUSSEAU Louis-Martin, Ph.D., membre et directeur de recherche

M. POTVIN Jean-Yves, Ph.D., membre et codirecteur de recherche

Mme. RIOPEL Diane, Ing., Docteure, membre

*À mes chers parents, mon frère Hichem, mon mari Mehdi, ma nièce Hawa, ma famille et tous mes proches et ami(e)s. Ceci est le fruit d'un long parcours allant de la mer méditerranéenne et traversant l'Atlantique pour ouvrir d'autres horizons que j'espère assez vastes pour contenir mes rêves !*

## REMERCIEMENTS

Avant de passer aux choses très techniques, j'aimerais témoigner ma gratitude envers les personnes qui ont contribué de près ou de loin à l'accomplissement de ce travail.

Je commence bien sûr par monsieur Louis-Martin Rousseau. C'est à travers lui que j'ai commencé à découvrir le monde de la recherche durant mon projet de fin d'études, une expérience qui s'est étendue agréablement jusqu'à ma maîtrise. Je le remercie pour tout ce qu'il a fait pour moi.

Mes pensées vont aussi à monsieur Jean-Yves Potvin qui m'a guidé tout au long de ce travail avec gentillesse et efficacité.

Je souhaite remercier également toute l'équipe du Centre de recherche sur les transports et plus précisément Serge Bisailon qui m'a souvent soutenu dans les tâches de programmation les plus difficiles.

Je remercie vivement Francine Benoît pour son aide quant à l'utilisation de LaTeX. J'exprime ma gratitude à tous mes amis Karima, Khadija, Mounira, Nabila, Mohamed et Nizar qui étaient toujours à mes côtés.

J'aimerais aussi adresser mes remerciements à Marielos et Marie-Carline pour leur gentillesse et leur empressement à m'aider, ainsi que toutes les personnes travaillant au département de Mathématiques et de Génie Industriel de l'École Polytechnique de Montréal, plus précisément Ayoub Insa Correa qui m'a soutenu avec ses conseils judicieux et m'a procuré les données avec lesquelles il a travaillé durant son Ph.D. Enfin, je voudrais terminer en remerciant les membres de mon jury.

## RÉSUMÉ

Le design et le contrôle de véhicules autoguidés prend de plus en plus d'importance avec la multiplication de leurs usages dans différents systèmes industriels, comme les mines souterraines et les ateliers de production flexibles. Dans ce projet de maîtrise, nous nous intéressons à deux problématiques très importantes touchant à l'utilisation des véhicules autoguidés dans un atelier de production flexible, soit l'affectation et l'ordonnancement de leurs tâches, ainsi que l'identification d'un routage pour les véhicules sans aucun conflit.

Nous disposons d'un ensemble d'activités de collectes et de livraisons devant être exécutées par des véhicules à des stations de travail précises. Le temps de début de chacune des deux activités est disponible. Notre objectif est de minimiser les retards de livraison sous différentes contraintes existantes entre les tâches.

Nous nous proposons d'aborder ces deux problématiques au sein d'une même approche de résolution afin d'obtenir une solution aussi proche que possible de la solution optimale, et ce, dans des temps de calcul raisonnables.

L'approche de résolution proposée décompose d'abord le problème en deux sous-problèmes, soit l'affectation des tâches aux véhicules et le routage de ces derniers. L'affectation se fait en insérant successivement chacune des tâches selon leur temps de début au plus tôt, tout en respectant les contraintes de précédence qui peuvent exister entre elles.

Au cours du processus, le cheminement exact des véhicules dans le réseau (ou routage) est considéré. En effet, l'algorithme développé produit le cheminement complet de chaque véhicule dans le réseau. Ceci s'oppose aux approches incrémentales où chaque nouveau segment de route est déterminé à mesure que le véhicule progresse.

De plus, chaque segment peut être utilisé par différents véhicules, à la condition que cette utilisation survienne dans des intervalles de temps différents. Les deux sous-problèmes d'affectation et de routage sont liés l'un à l'autre, de telle sorte que l'affectation des tâches aux véhicules peut être modifiée lorsque des conflits sont identifiés lors de la génération du cheminement exact des véhicules. Différents types de conflits sont traités et des stratégies spécifiques de résolution sont appliquées pour chacun des types.

Nous avons testé notre heuristique sur 15 problèmes tests. Ces problèmes sont distincts par la densité spatiale des lieux d'exécutions des tâches dans l'atelier de production (problèmes compacts, dispersés et mixtes). Ces problèmes contiennent de 7 à 12 tâches et de 1 à 9 relations d'antériorité. Chaque problème est testé pour un nombre de véhicule allant de deux à six.

Nous avons obtenu des solutions pour 71% des problèmes en des temps de calcul inférieurs à 3 secondes. Les problèmes à plus de trois véhicules sont résolus à l'optimalité. Mais, des retards de production considérables sont obtenus pour certains problèmes à deux véhicules, comparés aux retards obtenus par Correa et al. (2005).

## ABSTRACT

The control of automated guided vehicle systems (AGVs) become more and more complex because of their increasing use in different contexts. The problem considered here belongs to the operational level of the decision-making process, namely the assignment and ordering of tasks for AGVs and their conflict-free routing.

Our problem can be formulated as follows. Given a number of AGVs, their starting positions and a set of transportation tasks with a pick-up and a delivery point, find an assignment and an ordering of those requests for each vehicle, as well as a conflict-free routing in the network that minimizes the sum of lateness at delivery points, subject to various operational constraints. Lateness at a delivery point occurs when the actual delivery time exceeds the latest delivery time and is defined as the difference between the two values.

We have developed a construction heuristic that solves both the assignment and ordering of tasks and the identification of conflict-free routes.

First, an assignment is performed that satisfies all precedence constraints among tasks using an insertion algorithm.

Afterward, a heuristic is applied to detect and resolve any violation of operational constraints.

Finally, a route is produced, where conflicts are detected and solved. Five different types of conflicts are considered, each with its own resolution strategy.

The instances we used are built from Correa et al. (2005). We used 15 different request sets differing by the spatial density of the request in the manufacturing system (compact, spreadout and mixed). The number of transportation task in those 15 sets varies from 7 to 13. We solved each of them with 2, 3, 4, 5 and 6 AGVs. For each of the 15 sets, the number of precedence relationship between tasks varies from 1 to 9.

We report that 71% of instances are solved in less than 3 seconds. The problems with more than 3 AGVs are all solved to optimality. We encountered some problems with the instances with 2 AGVs, in fact the lateness at delivery point are important comparing to the results of Correa et al. (2005).

## TABLE DES MATIÈRES

DÉDICACE . . . . .	iv
REMERCIEMENTS . . . . .	v
RÉSUMÉ . . . . .	vi
ABSTRACT . . . . .	viii
TABLE DES MATIÈRES . . . . .	x
LISTE DES TABLEAUX . . . . .	xii
LISTE DES FIGURES . . . . .	xiii
INTRODUCTION . . . . .	1
CHAPITRE 1 REVUE DE LITTÉRATURE . . . . .	3
1.1 Niveau de décision stratégique . . . . .	3
1.2 Niveau de décision tactique . . . . .	4
1.3 Niveau de décision opérationnel . . . . .	6
CHAPITRE 2 DÉFINITION DU PROBLÈME . . . . .	12
2.1 Description de l'atelier . . . . .	12
2.2 Données du problème . . . . .	12
2.3 Contraintes . . . . .	14
2.4 Fonctionnement du système . . . . .	16
CHAPITRE 3 APPROCHE DE RÉOLUTION . . . . .	18
3.1 Affectation des tâches aux véhicules . . . . .	19
3.2 Vérification des contraintes opérationnelles . . . . .	22
3.3 Vérification et résolution des conflits de routage . . . . .	24

3.3.1	Modélisation des conflits . . . . .	25
3.3.2	Résolution des conflits . . . . .	29
3.4	Stratégie de résolution des conflits résiduels . . . . .	37
CHAPITRE 4	APPLICATIONS ET RÉSULTATS . . . . .	43
CONCLUSION	. . . . .	50
RÉFÉRENCES	. . . . .	52

**LISTE DES TABLEAUX**

Tableau 4.1	Description des problèmes . . . . .	44
Tableau 4.2	Description des problèmes . . . . .	45
Tableau 4.3	Problèmes à deux véhicules . . . . .	46
Tableau 4.4	Problèmes à trois véhicules . . . . .	47
Tableau 4.5	Problèmes à quatre véhicules . . . . .	48
Tableau 4.6	Problèmes à cinq véhicules . . . . .	49
Tableau 4.7	Problèmes à six véhicules . . . . .	49

## LISTE DES FIGURES

Figure 2.1	L'atelier de production flexible . . . . .	13
Figure 3.1	L'algorithme complet . . . . .	20
Figure 3.2	Différents états possibles d'un véhicule . . . . .	26
Figure 3.3	Conflit mN1 . . . . .	27
Figure 3.4	Conflit mN2 . . . . .	27
Figure 3.5	Conflit mN3 . . . . .	28
Figure 3.6	Conflit MN1 . . . . .	29
Figure 3.7	Conflit MS2 . . . . .	29
Figure 3.8	Stratégie de résolution des conflits majeurs . . . . .	31
Figure 3.9	Stratégie de résolution des conflits MN1 . . . . .	33
Figure 3.10	Stratégie de résolution des conflits MN1(2) . . . . .	34
Figure 3.11	Stratégie de résolution des conflits mN1 . . . . .	36
Figure 3.12	Stratégie de résolution des conflits mN2 . . . . .	37
Figure 3.13	Stratégie de résolution des conflits mN3 . . . . .	38
Figure 3.14	Conflits résiduels . . . . .	40

## INTRODUCTION

Ce mémoire de maîtrise s'inscrit dans le domaine du contrôle des véhicules auto-guidés. Il s'agit d'un domaine où l'on observe un recours de plus en plus fréquent à des équipements de manutention automatisés dans plusieurs domaines industriels comme dans les mines, les ports et les ateliers de production flexibles.

Dans notre travail, un ensemble de tâches doit être effectué sur un horizon de production donné. Chaque tâche est composée de deux activités, soit une collecte et une livraison. Essentiellement, une tâche pour un véhicule consiste à transporter une palette de produits entre deux stations de travail dans l'atelier tout en respectant certaines contraintes opérationnelles.

L'objectif est de desservir toutes les tâches tout en minimisant la somme des retards sur les livraisons.

La difficulté du problème vient des éléments suivants :

- la dépendance importante entre l'affectation des tâches à un véhicule donné et son routage effectif au sein de l'atelier, qui peut être source de conflits (voir point suivant);
- deux véhicules ne peuvent partager un même noeud ou un même segment à un même instant. Des mesures de recours sont toutefois possibles afin de régler de tels conflits comme modifier le chemin emprunté par un véhicule entre deux stations, retarder le départ d'un véhicule ou déplacer temporairement un véhicule afin de laisser passer un autre véhicule.
- dans une séquence de production, certaines tâches doivent absolument être exécutées avant d'autres tâches, d'où la présence de contraintes d'antériorité.
- quand un produit est livré à une station, ce produit doit être traité. Cette

station demeure donc inaccessible jusqu'à ce que le produit en question ait été traité et recueilli.

- la taille de la flotte de véhicules est un facteur important à considérer. En effet, l'augmentation du nombre de véhicules entraîne une augmentation des coûts (coûts relatifs à l'achat et à l'entretien des véhicules, coûts relatifs aux retards de production dû à la congestion sur le réseau) et leur diminution engendre une augmentation sur les retards de production.

Dans ce projet de maîtrise, nous nous proposons de développer une heuristique constructive permettant de résoudre le problème intégré d'ordonnancement et de routage sans conflit de véhicules autoguidés. Ce choix est basé sur les faits que ce problème n'a pas été résolu par une méthode heuristique (les heuristiques existantes ont porté séparément sur le routage ou sur l'ordonnancement), et que les méthodes exactes qui ont combiné les deux aspects sont très peu flexibles et très coûteuses en temps de calcul.

Nous avons testé notre méthode sur 15 problèmes différents par la densité spatiale des tâches dans l'atelier de fabrication, pour un nombre de véhicules allant de 2 à 6, car le contexte étudié ne justifie pas l'utilisation d'un plus grand nombre. Toutefois, notre heuristique peut être appliquée sur des problèmes à plus grand nombre de véhicules.

En rapport avec cette problématique, le présent mémoire est organisé comme suit. Dans le premier chapitre, nous présentons une revue de la littérature sur l'utilisation de véhicules autoguidés. Le chapitre 2 introduit ensuite le problème d'ordonnancement des tâches et de routage de véhicules sans conflit dans un atelier de production flexible. Dans le chapitre 3, l'approche heuristique qui a été développée pour résoudre ce problème est présentée. Au chapitre 4, des résultats sont rapportés sur un ensemble de problèmes tests et des comparaisons sont établies avec une méthode exacte. Enfin, le tout se termine par une conclusion.

## CHAPITRE 1

### REVUE DE LITTÉRATURE

Ce chapitre situe le problème qui nous intéresse dans l'ensemble des travaux portant sur l'exploitation des véhicules autoguidés.

Une revue récente sur le sujet se trouve dans Tuan et de DeKoster (2006). Dans ce travail, les auteurs ont classé les travaux portant sur la modélisation et le contrôle de systèmes faisant appel à des véhicules autoguidés selon les grandes catégories suivantes : conception de réseaux, détermination de la taille de la flotte de véhicules, localisation des véhicules en attente, gestion des batteries, ordonnancement des tâches et routage sans conflit des véhicules.

Il est important de signaler que l'une ou l'autre de ces problématiques peut intervenir dans un système industriel, selon l'objectif visé et le niveau de décision considéré (court, moyen ou long terme).

#### 1.1 Niveau de décision stratégique

La conception de réseaux est l'aspect le plus important au niveau de décision stratégique. La majorité des travaux publiés considèrent que les stations de travail sont connues et localisées a priori dans l'atelier.

Les algorithmes développés visent généralement à déterminer les connections les plus appropriées entre les stations afin de minimiser la distance parcourue par les véhicules.

Les trois catégories de réseaux les plus fréquentes sont : les réseaux conventionnels,

les réseaux à une seule boucle et les réseaux à transfert indirect.

### **Les réseaux conventionnels**

Il s'agit de réseaux affichant une topologie relativement complexe, comportant souvent des segments parallèles liant les noeuds du réseau, des raccourcis, des voies de dégagement et des intersections. Ces réseaux peuvent être unidirectionnels ou bidirectionnels.

### **Les réseaux à une seule boucle**

Il s'agit d'une forme primitive de réseau conventionnel où les véhicules doivent circuler sur une boucle unique, sans voies de dégagement ou de raccourcis.

### **Les réseaux à transfert indirect**

Ce type de réseau a été introduit pour la première fois dans Bozer et Srinivasan (1991). Il contient plusieurs zones, chacune étant desservie par un seul véhicule, et ces zones sont connectées entre elles par des stations de transfert.

## **1.2 Niveau de décision tactique**

Au niveau tactique, plusieurs problématiques peuvent être abordées, comme l'estimation de la taille de la flotte de véhicules, la localisation des véhicules en attente et la gestion des batteries.

### **L'estimation de la taille de la flotte**

Le nombre de véhicules utilisés influence beaucoup la performance du système ainsi que les coûts. On distingue deux types de véhicules.

- Les véhicules à charge unique :  
Ce sont des véhicules qui doivent obligatoirement livrer leur charge à destination, une fois qu'ils l'ont recueillie. L'ordonnement sur de tels véhicules est constitué d'une séquence de paires (collecte, livraison).
- Les véhicules à charges multiples :  
Ces véhicules peuvent faire plusieurs collectes successives avant de procéder aux livraisons.

### La localisation des véhicules en attente

On distingue ici deux types de stratégies.

- Les stratégies statiques  
Les auteurs dans Egbelu (1993), Ven der Meer et DeKoster (2000) suggèrent quatre approches.
  - Stationnement central :  
Une zone dans le réseau est réservée aux véhicules en attente.
  - Stationnement sur des boucles circulaires :  
Une boucle est réservée pour les véhicules en attente.
  - Stationnement dans les stations :  
Le véhicule demeure là où il a réalisé sa dernière activité.
  - Stationnement multiple :  
Différents noeuds dans le réseau sont réservés pour le stationnement des véhicules.
- Les stratégies dynamiques  
Ici, les zones de stationnement changent au cours de l'exécution des tâches. Il n'y a donc pas de zones ou de noeuds fixés à l'avance pour l'attente.

## La gestion des batteries

Bien que la problématique liée au chargement des batteries est un facteur important à considérer, il est omis dans la plupart des études. Selon MacHaney, cet aspect ne devrait être ignoré que dans certaines circonstances particulières, comme dans le cas d'un système à *chargement opportuniste* où le chargement est fait hors des heures de travail.

### 1.3 Niveau de décision opérationnel

L'affectation des tâches et le routage des véhicules entre les différentes stations sont des problématiques qui se posent sur le plan opérationnel. L'un des objectifs les plus importants à ce niveau est la minimisation du temps total de production.

Tel que mentionné précédemment, ces deux problématiques sont étroitement liées. Alors que l'affectation permet de savoir quelles sont les tâches qui sont affectées à chacun des véhicules et dans quel ordre ces tâches sont réalisées, le routage produit un cheminement exact dans le réseau qui n'induit pas de conflit avec les cheminements des autres véhicules.

Dans cette section, nous donnons d'abord un bref aperçu des travaux portant sur l'affectation et l'ordonnancement des tâches puis sur le routage de véhicules. Finalement, nous abordons les méthodologies proposées dans la littérature qui s'attaquent simultanément aux deux problématiques.

#### L'affectation et l'ordonnancement

On parle souvent de répartition lorsqu'une nouvelle tâche est affectée à un véhicule. On s'intéresse alors aux règles de choix heuristiques utilisées pour

effectuer cette affectation.

Dans la littérature, différentes règles sont proposées. Par exemple, une *répartition initiée par les stations* survient quand des produits doivent être transportés à partir d'une station donnée. Par ailleurs, une *répartition initiée par les véhicules* survient quand un véhicule termine sa tâche courante et attend qu'on lui en affecte une autre.

L'ordonnancement des tâches est obtenu de façon implicite par la règle d'affectation choisie, puisque la tâche courante est simplement ajoutée à la fin des tâches déjà affectées à un véhicule.

Cet ordonnancement, couplé au routage, permet d'associer une date d'arrivée et de départ pour un véhicule à chaque station, ainsi que sur chacun des noeuds ou segments intermédiaires.

## Le routage

Le routage consiste à déterminer le cheminement exact du véhicule dans le réseau entre les différentes stations qu'il doit desservir.

Selon Qiu et al. (2002), les travaux portant sur le routage se divisent en deux grandes familles selon qu'on considère un routage au sein d'un réseau quelconque ou un routage dans un réseau spécifique.

On s'intéresse dans ce qui suit au routage dans un réseau quelconque, qui ne dépend pas des caractéristiques spécifiques du réseau. Il s'agit d'identifier un routage réalisable (sans conflits). C'est à dire, que deux véhicules ne doivent pas se retrouver au même instant, sur un même segment ou un même noeud. Deux types de collisions peuvent se produire.

- Une collision frontale qui se produit quand deux véhicules partent de deux noeuds adjacents (reliés par un même segment), en même temps et en sens

opposé l'un à l'autre.

- Une collision tamponnage survient quand deux véhicules aboutissent en même temps, à un même noeud.

Le problème de routage des véhicules autoguidés a été abordé de trois façons différentes : de façon statique, en faisant appel à des fenêtres de temps ou de façon dynamique.

- Méthodes de résolution statiques :

Les méthodes statiques sont généralement appliquées quand la taille de la flotte ainsi que le nombre de tâches à effectuer est assez petit.

Elles ont été introduites pour la première fois dans des réseaux unidirectionnels par Brodbent et al. (1985). Pour éviter les conflits frontaux, les auteurs excluent le segment conflictuel et recherchent un autre chemin plus court pour l'un des deux véhicules. Pour les conflits tamponnage, ils retardent le départ d'un véhicule pour que l'autre ait le temps de terminer ses activités.

Dans Egbelu et Tanchoco (1986), les auteurs ont appliqué ces mêmes techniques dans des réseaux bidirectionnels.

Un des premiers algorithmes pour le routage de véhicules dans des réseaux bidirectionnels est rapporté dans Daniels (1988). Cet algorithme ne permet toutefois pas une exploitation optimale du réseau puisque les segments formant la route d'un véhicule entre deux stations restent inutilisables jusqu'à ce que le véhicule atteigne sa destination.

C'est d'ailleurs là l'une des limites des méthodes statiques. Les méthodes basées sur les fenêtres de temps permettent de remédier à cette lacune.

- Méthodes de résolution basées sur les fenêtres de temps :

Ces méthodes permettent d'utiliser un même segment par différents véhicules en autant que les fenêtres d'utilisation soient différentes. Dans Kim et Tanchoco (1991), les auteurs transforment leur problème original en un problème sur un graphe auxiliaire. Dans ce graphe, les noeuds correspondent aux fenêtres de temps disponibles. Deux noeuds sont liés par un segment si et seulement s'il y a deux stations qui disposent de ces deux fenêtres de temps et qui sont liées dans le réseau. Il semble que cet algorithme demande beaucoup de temps de calcul, même pour de petits problèmes.

Les mêmes auteurs Kim et Tanchoco (1993) ont travaillé plus tard sur le contrôle opérationnel des véhicules en utilisant l'algorithme décrit dans Kim et Tanchoco (1991). Ils ont considéré le routage d'un seul véhicule à la fois, en respectant le routage préétabli pour les autres véhicules.

- Méthodes de résolution dynamiques :

Contrairement aux méthodes statiques qui sont centralisées (c.-à.-d., le contrôle se fait au niveau du centre de contrôle de l'atelier et le plan de routage d'un véhicule est connu à son entrée dans le système), la méthode dynamique proposée par Taghaboni (1995) est de nature plus locale. L'utilisation de chaque segment est déterminée dynamiquement pendant le routage. La méthode est basée sur une technique de planification incrémentale. Ce n'est qu'une fois arrivé à un noeud, qu'une décision sur le prochain noeud à visiter est prise, et ce jusqu'à ce que le véhicule atteigne sa destination.

L'objectif local de l'algorithme est de minimiser le temps de parcours

entre deux noeuds adjacents, ce qui ne garantit toutefois pas l'optimalité au niveau global.

Cette constatation a incité les auteurs à inclure un critère de sélection additionnel qui tient compte de l'état global du réseau.

Cette méthode est plus rapide comparée aux méthodes centralisées mais elle ne garantit pas l'optimalité et a tendance à moins bien fonctionner quand le nombre de véhicules augmente.

### **Ordonnancement et routage intégrés**

Plusieurs travaux ont porté seulement sur un seul des deux aspects d'affectation et ordonnancement des tâches et de routage. Cependant, leur combinaison a rarement été étudiée. Les travaux qui ont intégré les deux aspects sont ou bien très coûteuse en terme de temps de calcul ou elles se limitent à un très petit nombre de chariots.

En faisant appel à la programmation dynamique, les auteurs dans Langevin et al. 1996 ont développé un algorithme qui permet de produire une solution optimale pour l'affectation, l'ordonnancement et le routage sans conflits de deux véhicules. Bien qu'une extension de l'algorithme est proposée pour plus de deux véhicules, cette nouvelle version ne garantit pas l'optimalité.

Dans Desaulniers et al. (2003), les auteurs présentent une méthode exacte basée sur la génération de colonnes. Elle a été testée sur des problèmes contenant jusqu'à quatre véhicules. L'efficacité de leur méthode dépend toutefois de la qualité de la solution produite par l'heuristique gloutonne de départ. Il peut même arriver qu'aucune solution réalisable ne puisse être identifiée par l'heuristique gloutonne, empêchant ainsi l'application de la méthode exacte.

Dans Correa et al. (2005), les auteurs proposent une approche hybride de programmation par contraintes et de programmation linéaire en nombres entiers pour aborder l'intégration des problématiques d'affectation des tâches et de routage. Ils ont pu résoudre des problèmes avec six véhicules. Leur méthode est plus flexible que celle décrite dans Desaulniers et al. (2003), mais s'avère très coûteuse en temps de calcul.

Afin de comparer nos résultats à ceux de Correa et al. (2005), nous avons gardé la même description du problème sur lequel il a travaillé. L'atelier de production flexible étudié dans ce projet est muni d'un *réseau conventionnel bidirectionnel*. Il dispose de *véhicules à charge unique*. Le système qui nous intéresse dans cette étude est à *chargement opportuniste*, le chargement des batteries des véhicules n'est donc pas considéré. Il adopte la troisième stratégie statique de stationnement citée ci-dessus, soit le *stationnement dans les stations*.

## CHAPITRE 2

### DÉFINITION DU PROBLÈME

Dans ce chapitre, nous présentons une description du réseau dans l'atelier de production flexible, des ressources utilisées, des contraintes opérationnelles ainsi que l'objectif visé.

#### 2.1 Description de l'atelier

La figure 2.1 illustre l'atelier de production automatisé que nous étudions. Ce système comporte plusieurs unités de travail où s'effectuent des fonctions spécifiques telles que le lavage, le fraisage, l'assemblage, etc.

Chaque unité est connectée au réseau par une station où ont lieu les collectes et les livraisons des produits à transporter. Chacun des segments entre un noeud de départ et un noeud d'arrivée (petits points noirs sur la figure) est considéré de longueur identique. Par ailleurs, le réseau est bidirectionnel ce qui signifie que les véhicules peuvent circuler dans un sens ou dans l'autre sur un segment. Pour fins de résolution, l'horizon de planification est discrétisé en périodes de 15 secondes (qui correspond au temps de parcours des véhicules sur un segment).

#### 2.2 Données du problème

Les données du problème peuvent être subdivisées en données relatives aux ressources et aux tâches.

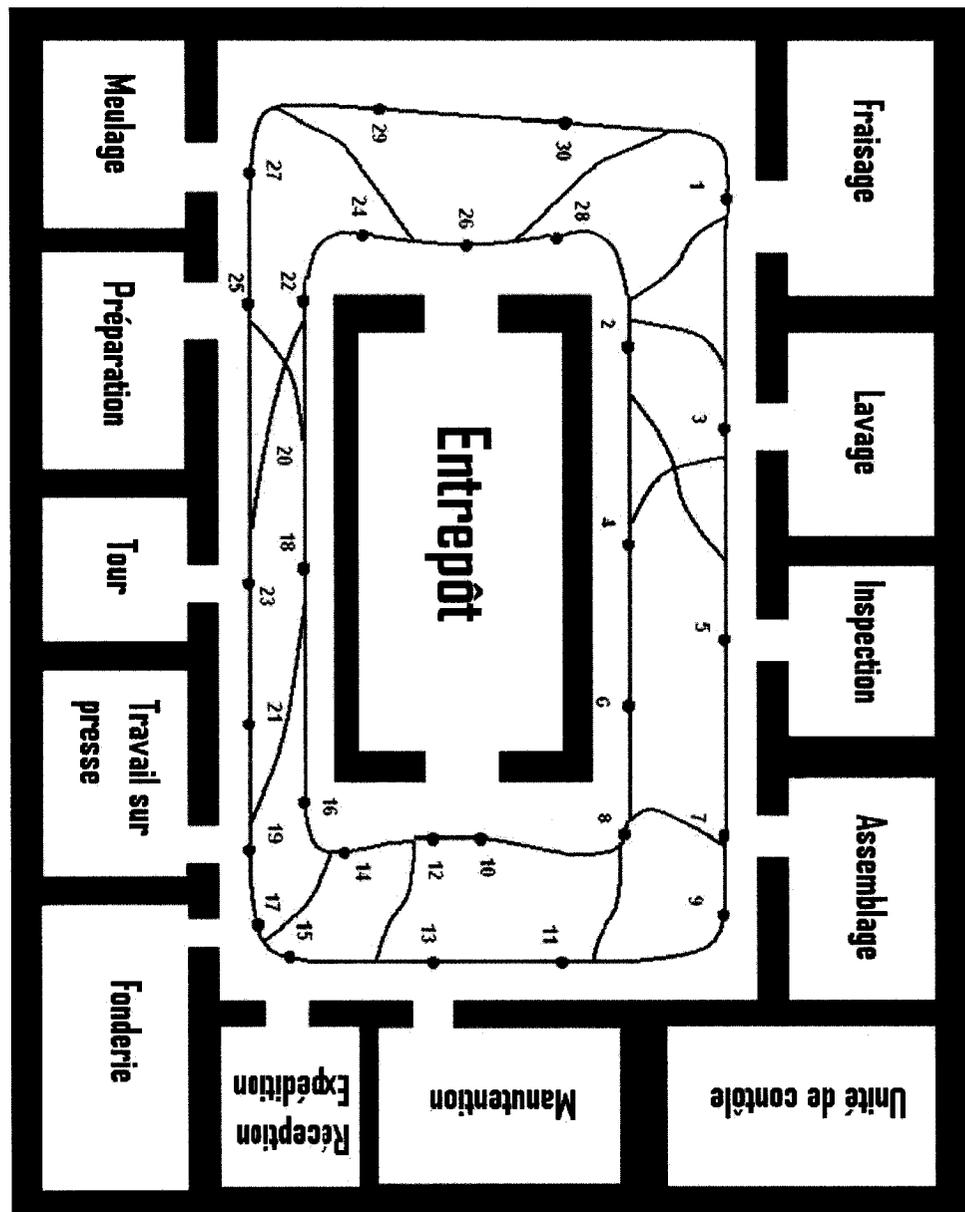


Figure 2.1 L'atelier de production flexible

## Ressources

Les ressources correspondent essentiellement aux véhicules autoguidés. Ce sont les équipements de manutention qui se déplacent sur le réseau en transportant des palettes de produits d'une station à une autre.

Leurs caractéristiques principales sont les suivantes.

- Leur nombre ainsi que leur localisation sur un noeud du réseau est fixé au début de l'horizon de planification.
- Tous les véhicules sont identiques et ont la même vitesse.
- La charge de chaque véhicule est d'une seule palette de produit.

## Tâches

Une tâche se compose de deux activités, soit une collecte et une livraison. Pour chaque activité, on dispose des données suivantes: le noeud (station), le temps de début au plus tôt et le temps de service des machines pour effectuer le traitement. Dans le cas d'une livraison, on a également un temps de début au plus tôt.

Par ailleurs, la durée de chargement et de déchargement est fixée à une période (15 secondes).

### 2.3 Contraintes

Les contraintes dépendent beaucoup de la nature de l'atelier et de son réseau de transport. Dans notre cas, on retrouve :

## Contraintes sur les véhicules

Ces contraintes portent sur le positionnement, le déplacement et la charge des véhicules autoguidés.

- La capacité des véhicules est unitaire (c.-à.-d., une palette).
- Chaque véhicule doit être présent à un noeud au début de l'horizon (période 0).
- Chaque véhicule doit séjourner à une station pendant au moins une période.
- Chaque véhicule doit être positionné à un noeud donné à chaque période (en effet, le temps de parcours sur chacun des segments correspond à une période).
- Si un véhicule effectue une collecte, ce même véhicule doit effectuer la livraison associée.

## Contraintes d'antériorité

Il y a deux types de contraintes d'antériorité.

- Contraintes intra-tâches  
Il existe une contrainte d'antériorité entre la collecte et la livraison associée à une même tâche. Les deux activités ont lieu à des noeuds différents et sont réalisées par le même véhicule.
- Contraintes inter-tâches  
Il peut y avoir des contraintes d'antériorité entre deux tâches associées à un même produit, qui proviennent de l'ordre d'exécution des opérations sur les machines ainsi que du besoin en manutention des outillages requis par les tâches de transformation (outils de coupe et montages d'usinage).

Les deux tâches peuvent alors être exécutées par des véhicules différents.

### **Contraintes opérationnelles**

Quand un véhicule livre un produit  $A$  à une station, cette dernière demeure inaccessible tant que le traitement du produit  $A$  n'est pas terminé et que sa collecte n'a pas été réalisée.

### **Routage sans conflit**

Il ne peut y avoir plus d'un véhicule sur un même segment ou un même noeud à une période donnée, sinon il y a conflit de routage. On distingue deux types de conflits.

- Des conflits tamponnage sur les noeuds, quand deux véhicules aboutissent au même noeud à la même période.
- Des conflits frontaux sur les segments, quand deux véhicules circulent en sens opposé sur un même segment à la même période.

## **2.4 Fonctionnement du système**

Une liste de commandes est présentée quotidiennement. Chaque commande correspond à une ou plusieurs unités d'un produit à fabriquer.

Cette fabrication correspond à une succession pré-établie d'opérations sur les diverses stations de travail. Les véhicules, partant d'un certain noeud de départ, doivent assurer le transport des produits entre les différentes stations. À la fin de la journée, le véhicule s'arrête là où il a complété sa dernière tâche.

Il y a un retard lorsqu'une palette est livrée après l'heure de début au plus tôt de la livraison.

### **Fonction objectif**

Le problème consiste à trouver une affectation des tâches aux véhicules et un routage sans conflit de ces véhicules de façon à minimiser la somme des retards de livraison.

## CHAPITRE 3

### APPROCHE DE RÉOLUTION

Ce chapitre décrit l'approche heuristique que nous avons développée. On aborde le problème en considérant à la fois l'affectation des tâches pour chacun des véhicules, de façon à respecter les contraintes d'antériorité entre les tâches, et le routage des véhicules.

Le mécanisme de génération du cheminement des véhicules peut être appliqué à des réseaux bidirectionnels à topologie quelconque.

L'algorithme de routage appartient à la famille des méthodes basées sur les intervalles de temps. Contrairement aux méthodes statiques, qui condamnent l'utilisation d'un segment dès le début du parcours du véhicule jusqu'à la fin, notre approche permet l'utilisation d'un même segment par plusieurs véhicules dans des périodes différentes.

C'est aussi un routage complet, contrairement à la planification incrémentale qui se fait segment par segment. Il ne se situe pas dans une perspective préventive des conflits mais accepte plutôt les conflits et les répare par la suite.

Donc, après chaque affectation, nous aurons une borne inférieure sur la somme des retards de production qui pourra s'améliorer ou se détériorer si des conflits apparaissent et doivent être réparés.

L'algorithme global est décrit à la figure 3.1. Cet algorithme dans sa version finale est obtenu en appliquant progressivement plusieurs stratégies de résolution. Certaines de ces stratégies ont été utilisées dans la littérature, donc nous les avons adaptées à notre problème. D'autres sont nouvelles, nous les avons établies après plusieurs essais de résolution des conflits qui surviennent lors du routage.

L'idée de base qui a guidé nos choix est de partir d'une borne inférieure sur les retards de la production et essayer de la garder, au lieu de l'idée classique de partir d'une borne supérieure et essayer de l'améliorer.

### 3.1 Affectation des tâches aux véhicules

La première étape consiste à affecter les tâches aux différents véhicules. L'ordre d'affectation des tâches est obtenu en triant d'abord les tâches selon leur date de livraison au plus tôt.

Puis, on vérifie les contraintes d'antériorité entre les tâches. Si l'ordre courant viole la contrainte d'antériorité pour deux tâches  $T$  et  $T'$ , alors  $T'$  prend la place de  $T$ ,  $T$  est placée immédiatement après  $T'$  et toutes les autres tâches qui suivent sont décalées. Une description de ces étapes est présentée dans l'algorithme 3.1.

---

#### Algorithme 3.1 Ordonnancement des tâches [Étape 1. de l'algorithme 3.1]

---

```

trier les tâches selon leur temps de début de livraison croissant
i ← 1
tant que i < nbTâches faire
  pour j = i à nbTâches faire
    si (tempsDébutLivraison( $T_{i-1}$ ) < tempsDébutLivraison( $T_j$ )) et
     $\exists$ (ContraintePrécédence( $T_{i-1}, T_j$ )) alors
      Placer  $T_j$  à la place de  $T_{i-1}$ 
      Placer  $T_{i-1}$  juste après  $T_j$ 
      Décaler les autres tâches
    si (i - 1 > 1) alors
      i ← i - 1
    sinon
      i ← i + 1
    finsi
  finsi
finpour
i ← i + 1
fantant que

```

---

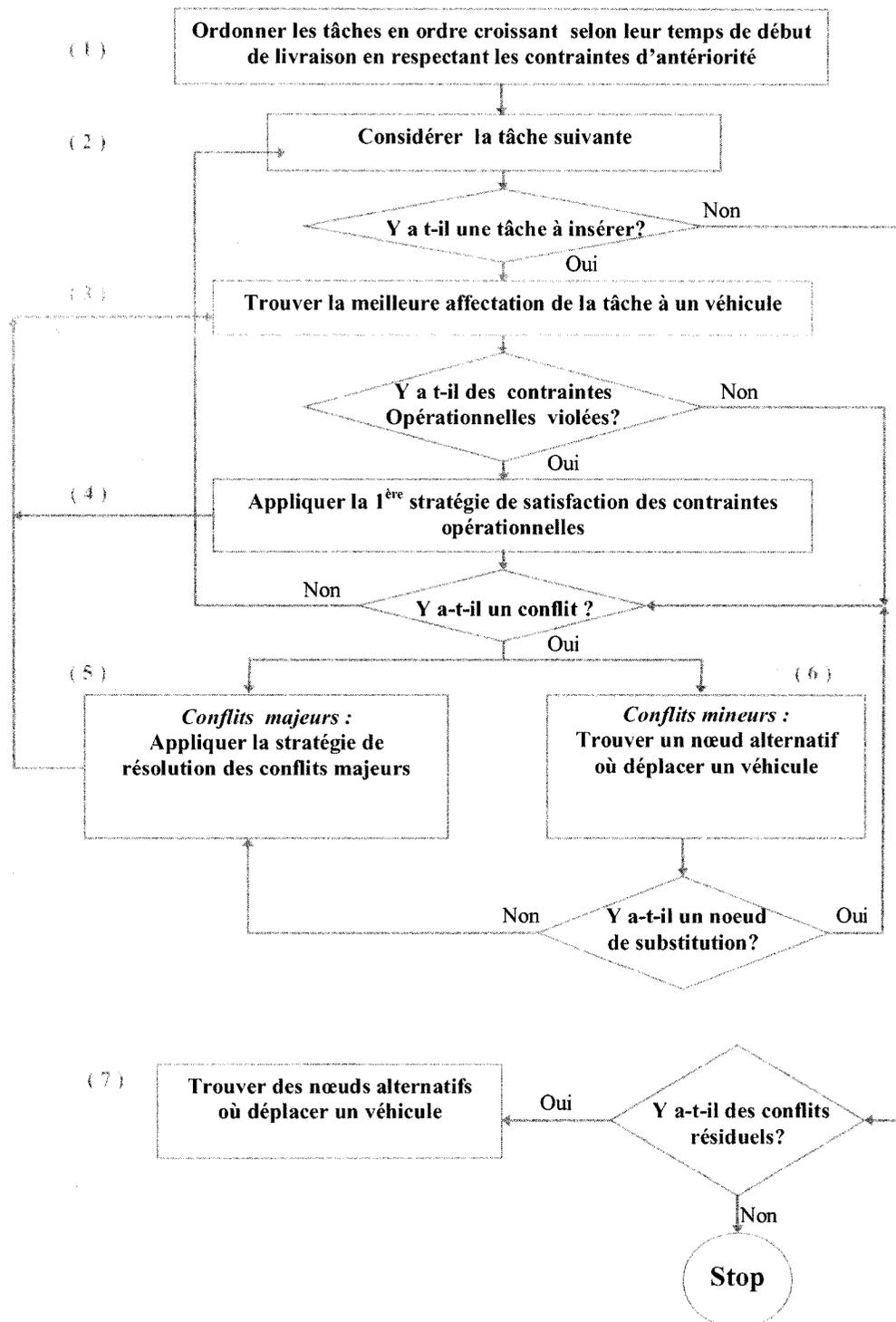


Figure 3.1 L'algorithme complet

Après avoir traité toutes les violations de contraintes d'antériorité, l'ordre final pour l'affectation des tâches est obtenu.

Pour ce qui est de l'insertion elle-même, il s'agit essentiellement de décider à quel véhicule sera affectée la tâche courante, puisque l'insertion est toujours faite à la fin de la séquence de tâches affectée à un véhicule. Le critère retenu pour l'affectation est la minimisation du temps d'arrivée du véhicule à la station de livraison ce qui, implicitement, minimise le retard de livraison, comme indiqué dans l'algorithme 3.2.

Si deux véhicules ont le même coût d'affectation, on choisit celui qui est le plus près de la station de collecte de la tâche courante.

Et s'il y a toujours égalité, on choisit le véhicule le plus occupé. Le véhicule affecté à la tâche courante est nommé véhicule courant.

Le temps d'arrivée du véhicule au noeud de livraison est obtenu de la façon suivante:

- On considère d'abord le temps de parcours sur le plus court chemin menant du noeud courant du véhicule au noeud de collecte afin d'obtenir le temps d'arrivée du véhicule au noeud de collecte.
- On prend le maximum entre ce temps d'arrivée et le temps de début de service au plus tôt au noeud de collecte.
- On additionne ensuite le temps de chargement.
- On additionne enfin le temps de parcours sur le plus court chemin menant du noeud de collecte au noeud de livraison.

Si le véhicule arrive avant le temps de début d'une activité à une station, il doit donc attendre jusqu'à ce temps. Dépendant des circonstances, un véhicule attendra ou bien à son noeud courant ou bien à la station elle-même (par exemple, il peut être avantageux pour le véhicule de quitter immédiatement son noeud courant et

---

**Algorithme 3.2** Affectation des tâches aux véhicules [Étape 3. de l'algorithme 3.1]
 

---

Initialiser  $minimum$  à une très grande valeur  
**pour**  $i = 1$  à  $nbVéhicules$  **faire**  
    $cheminC_i \leftarrow$  la longueur du chemin du noeud source du véhicule  $i$  vers le noeud  
   collecte  
    $cheminL \leftarrow$  la longueur du chemin du noeud collecte vers le noeud livraison  
    $tempsDébutRéal_i = \max tempsDébut_i + cheminC_i, tempsDébutCollecte$   
    $Cout_i = tempsDébutRéal_i + tempsChargement + cheminL$   
   **si** ( $Cout_i < minimum$ ) **alors**  
      $minimum \leftarrow Cout_i$   
     affecter la tâche courante au véhicule  $i$   
**fin**  
**finpour**

---

attendre à la station, si ce noeud courant est requis par un autre véhicule).

Il faut aussi noter qu'on ne tient pas compte des conflits possibles à ce stade. On a donc une borne inférieure sur le retard accumulé. Après chaque affectation, nous devons mettre à jour :

- Les temps de début des tâches non encore affectées qui sont liées par des contraintes d'antériorité avec la tâche courante qui vient d'être ajoutée.
- Le noeud courant du véhicule ainsi que son temps de départ (ou temps de disponibilité) courant.

### 3.2 Vérification des contraintes opérationnelles

On doit ensuite vérifier la satisfaction des contraintes opérationnelles (l'étape 4. de l'algorithme 3.1) qui portent sur les temps de début de certaines activités ayant lieu aux mêmes noeuds.

### Détection des violations des contraintes opérationnelles

Supposons que la livraison  $L_1$  pour une tâche donnée  $T_1$  et la collecte  $C_2$  pour une autre tâche  $T_2$ , toutes deux associées au même produit, sont liées par une contrainte d'antériorité à un noeud. Alors, si la collecte  $C$  ou la livraison  $L$  de la tâche courante  $T$  a lieu au même noeud et que son temps planifié de début se situe entre les temps planifiés de début  $L_1$  et  $C_2$ , une contrainte opérationnelle est violée. Les étapes de la détection des violations des contraintes opérationnelles sont données dans l'algorithme 3.3.

---

#### Algorithme 3.3 Détection des contraintes opérationnelles

---

Soit la tâche courante  $T$  à insérer dont les deux activités associées sont la livraison  $L$  et la collecte  $C$

```

pour  $i = 1$  à  $nbTâches$  faire
  pour  $j = 1$  à  $nbTâches$  faire
    si ( ( $i \neq j$ ) et  $\exists(ContraintePrécédence(L_i, C_j))$  ) alors
      si ( ( $Noeud(L) = Noeud(L_i) = Noeud(C_j)$ ) et ( $tempsDébut(L_i) <$ 
         $tempsDébut(L) < tempsDébut(C_j)$ ) ) alors
        ContrainteOpérationnelle( $L_i, C_j, L$ ) est violée
      sinonsi ( ( $Noeud(C) = Noeud(L_i) = Noeud(C_j)$ ) et ( $tempsDébut(L_i) <$ 
         $tempsDébut(C) < tempsDébut(C_j)$ ) ) alors
        ContrainteOpérationnelle( $L_i, C_j, C$ ) est violée
      finsi
    finsi
  finpour
finpour

```

---

### Satisfaction des contraintes opérationnelles

Dans le cas évoqué plus haut, la collecte  $C$  ou la livraison  $L$  de la tâche courante  $T$  sera retardée. Le temps planifié de début de l'activité sera alors fixé au temps planifié de début de collecte de la tâche  $T_2$  plus le temps de chargement plus une période. Cette période correspond à l'unité de temps nécessaire pour parcourir le segment qui correspond à la variable *const* dans l'algorithme 3.4.

---

**Algorithme 3.4** Satisfaction des contraintes opérationnelles: première stratégie
 

---

Soit la tâche courante  $T$  à insérer dont les deux activités associées sont la livraison  $L$  et la collecte  $C$

$const \leftarrow 1$

**pour**  $i = 1$  à  $nbTâches$  **faire**

**pour**  $j = 1$  à  $nbTâches$  **faire**

**si**  $contrainteOpérationnelle(L_i, C_j, L)$  violée **alors**

$tempsDébut(L) = tempsDébut(C_j) + tempsDéchargement + const$

**sinon si**  $contrainteOpérationnelle(L_i, C_j, C)$  violée **alors**

$tempsDébut(C) = tempsDébut(C_j) + tempsChargement + const$

**fin si**

**fin pour**

**fin pour**

---

Toutefois, la résolution des conflits et la disponibilité des véhicules peuvent retarder le temps actuel de début de  $T_2$ , causant à nouveau une violation de la contrainte opérationnelle.

Une deuxième stratégie est appliquée dans ce cas, pour éviter que  $T$  soit entre  $T_1$  et  $T_2$ . Il s'agit de placer  $T_2$  juste après  $T_1$  et de décaler toutes les autres tâches en respectant les contraintes d'antériorité (voir algorithme 3.5). Cette deuxième stratégie est fructueuse si toutes les contraintes d'antériorité sont respectées et qu'au plus deux tâches sont liées par des contraintes opérationnelles sur une même station. Elle est appliquée à l'étape 1. de l'algorithme 3.1.

### 3.3 Vérification et résolution des conflits de routage

La détection des conflits de routage se fait après chaque affectation de tâche.

---

**Algorithme 3.5** Satisfaction des contraintes opérationnelles: deuxième stratégie
 

---

```

pour  $i = 1$  à  $nbTâches$  faire
  pour  $j = 1$  à  $nbTâches$  faire
    si  $(T_i < T < T_j)$  et  $((contrainteOpérationnelle(L_i, C_j, L)$  violée) ou
       $(contrainteOpérationnelle(L_i, C_j, C)$  violée) alors
        si  $(tempsDébutRéel(T_j) > tempsDébutPrévu(T_j))$  alors
          placer  $T_j$  juste après  $T_i$ 
          décaler les autres tâches
        finsi
      finsi
    finpour
  finpour

```

---

### 3.3.1 Modélisation des conflits

On utilise une stratégie de détection de conflits qui fait appel à la position du véhicule à chaque période et à son état.

#### La notion d'état

On suppose qu'un véhicule peut être dans deux états possibles: l'état d'attente (attente d'être affecté ou attente en cours d'exécution de tâche) et l'état d'action (le véhicule parcourt un chemin d'une station à une autre, ou bien il est en train de réaliser une collecte ou une livraison).

Cette notion d'état facilite le contrôle et l'exploitation des véhicules autoguidés. En effet, elle permet non seulement d'établir une distinction entre les stratégies de résolution de conflits, mais pourrait aussi permettre d'introduire des activités comme l'entretien et la recharge des batteries lorsque les véhicules sont en attente, un aspect souvent ignoré dans les travaux portant sur le contrôle de véhicules autoguidés.

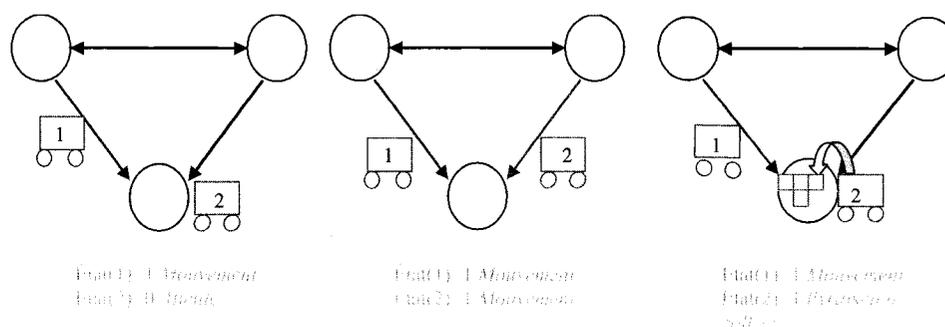


Figure 3.2 Différents états possibles d'un véhicule

### La nature des conflits

Dans la littérature, on retrouve deux grandes familles de conflits selon leur occurrence sur un noeud ou sur un segment.

Nous distinguons les conflits mineurs (type m) sur un noeud lorsqu'ils n'entraînent pas une remise en question de l'affectation courante et une réévaluation du coût de la solution et les conflits majeurs (type M) sur un noeud ou un segment dont la résolution peut induire un changement d'affectation des tâches et une réévaluation du coût de la solution.

Dans tout ce qui suit, nous entendons par N, un conflit sur un noeud, et par S, un conflit sur un segment. Les deux parenthèses utilisées pour distinguer les différents types de conflits comportent comme premier terme l'état du véhicule courant et comme deuxième terme l'état de l'autre véhicule impliqué dans ce conflit.

### Conflit sur un noeud

Un tel conflit se produit lorsque deux véhicules se trouvent sur un même noeud à une même période. Selon l'état des véhicules, on distingue les

cas suivants:

1. Conflit mN1(1-0)

Le véhicule affecté à la tâche courante est en action (état 1), alors que le véhicule qui entre en conflit avec ce dernier est en attente (état 0). Ce conflit est illustré à la figure 3.3.

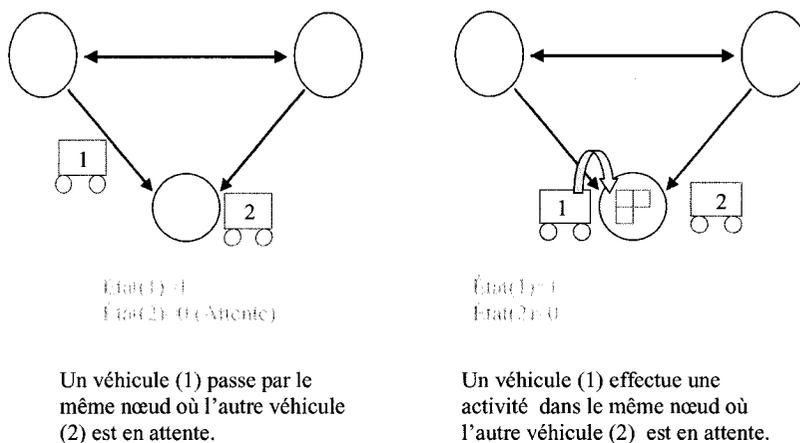


Figure 3.3 Conflit mN1

2. Conflit mN2(0-0)

Les deux véhicules qui entrent en conflit sont en attente (état 0). Ce conflit est illustré à la figure 3.4.

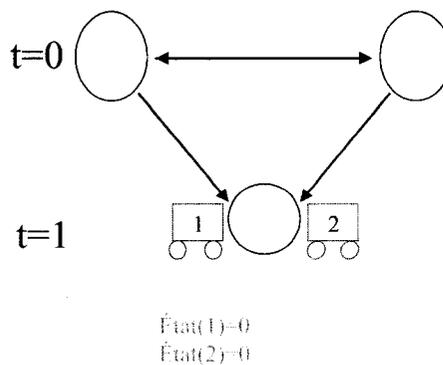
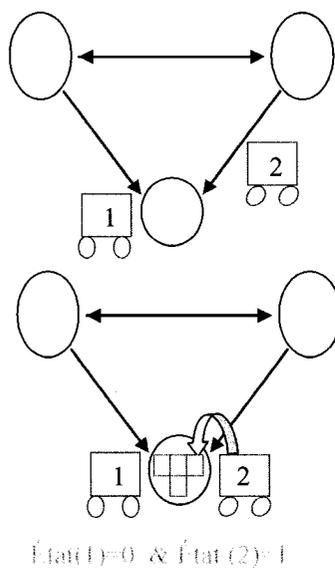


Figure 3.4 Conflit mN2

### 3. Conflit mN3(0-1)

C'est l'opposé du conflit mN1. Il se produit quand le véhicule affecté à la tâche courante est en attente (état 0), alors que le véhicule qui entre en conflit avec ce dernier est en mouvement (état 1). Ce conflit est illustré à la figure 3.5.



Le véhicule courant (1) est en attente.  
Le véhicule (2) est en marche

Figure 3.5 Conflit mN3

### 4. Conflit MN1(1-1)

C'est le premier type de conflit majeur. Il se produit quand les deux véhicules sont en action. La figure 3.6 illustre ce conflit.

### Conflit sur un segment MS2(1-1)

On retrouve un seul type de conflit sur un segment, illustré à la figure 3.7. Il est majeur et se produit quand deux véhicules se trouvent au même instant, en sens opposé l'un à l'autre sur le même segment.

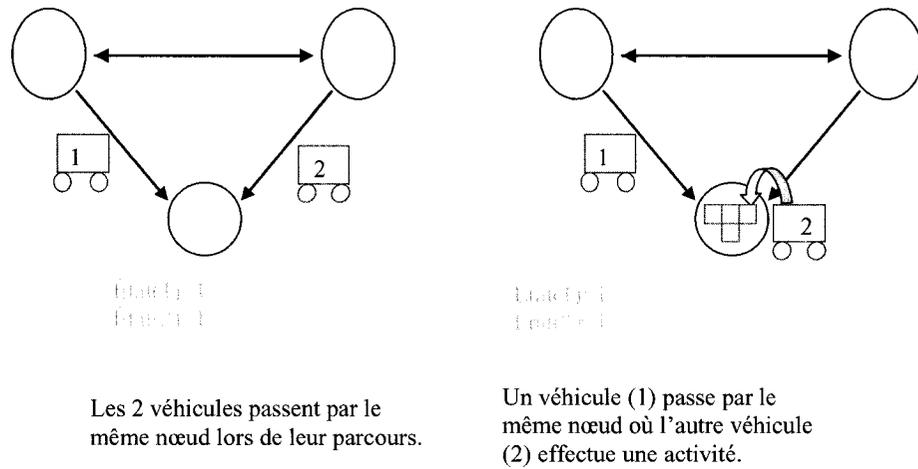


Figure 3.6 Conflit MN1

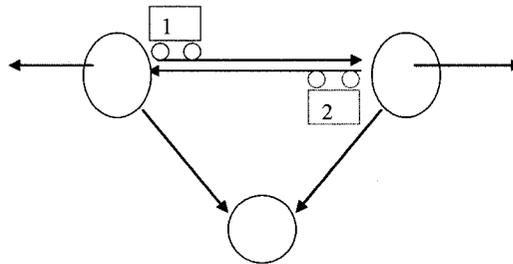


Figure 3.7 Conflit MS2

### 3.3.2 Résolution des conflits

Une stratégie de résolution spécifique, de nature locale, est appliquée selon la nature du conflit. Ces stratégies permettent d'exploiter au mieux la capacité du réseau puisqu'elles ne condamnent pas les segments utilisés par un véhicule tout au long de son déplacement.

La position du véhicule à une période donnée peut changer suite à l'application d'une stratégie donnée, ce qui peut engendrer d'autres conflits. On doit donc détecter les conflits pour chaque période qui suit le départ du véhicule de son nœud courant.

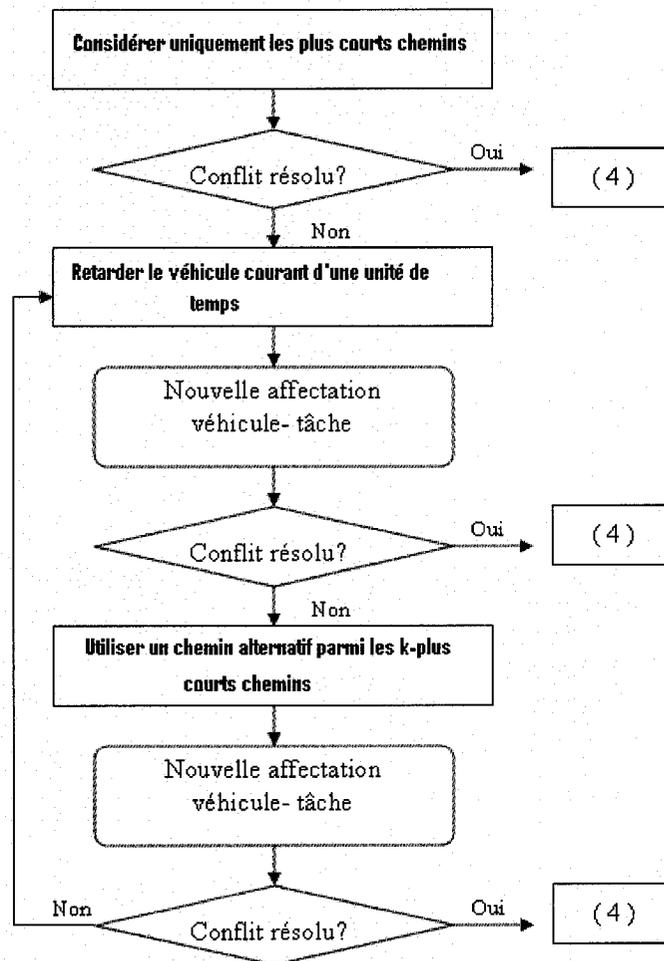
## Résolution des conflits majeurs MN1 et MS2

La séquence d'actions à exécuter dans le cas d'un conflit majeur est donnée à la figure 3.8. Cette séquence s'interrompt dès que le conflit est résolu.

1. Considérer tout autre plus court chemin(algorithme de Dijkstra), s'il en existe, entre le noeud courant du véhicule et la station de collecte ou entre les stations de collecte et de livraison.
2. Retarder le départ du véhicule courant d'une période. Réévaluer l'affectation de la tâche courante au véhicule. En effet, suite au décalage d'une période, il se peut qu'il soit plus avantageux d'affecter la tâche à un autre véhicule.
3. Appliquer un algorithme des  $k$ -plus courts chemins afin de trouver des cheminements alternatifs entre le noeud courant du véhicule et la station de la collecte ou entre les stations de collecte et de livraison. Pour chacun des chemins considérés, réévaluer l'affectation de la tâche courante au véhicule. En effet, suite à l'utilisation d'un chemin plus long, il se peut qu'il soit plus avantageux d'affecter la tâche à un autre véhicule.
4. Retourner en 2.

Il faut noter qu'il faut assez souvent itérer entre les étapes 2. et 4. car le réseau est peu dense.

Comme nous n'avons pas besoin d'une énumération exhaustive de tous les  $k$ -plus courts chemins entre toutes les sources et les destinations ( l'aspect glouton de notre heuristique ne l'exige pas), on a élaboré une méthode heuristique pour calculer certains des  $k$  plus chemins les moins longs entre chaque source et chaque destination. Notre heuristique est adaptée de l'algorithme de Yen (1971). Elle permet d'obtenir ces chemins plus simplement (facile à implémenter), en des temps de calcul pas très importants.



Note: (4) correspond à la quatrième étape de la figure 3.1

Figure 3.8 Stratégie de résolution des conflits majeurs

Notre méthode est donnée par l'algorithme 3.6.

---

**Algorithme 3.6** Recherche des k-plus courts chemins

---

Soit  $PathVector$  l'ensemble des k-plus courts chemins qu'on calculera entre une source  $s$  et une destination  $d$ .

$K=5$

$PathsVector = \emptyset$

Trouver le plus court chemin  $SP$  entre une source  $s$  et une destination  $d$  en utilisant l'algorithme de Dijkstra

$PathsVector = PathsVector \cup SP$

$n = 1$

$indice = 0$

**tant que** ( $n < K$ ) et ( $indice \neq$  la taille de  $PathsVector$  **faire**

prendre le premier chemin  $p$  dans  $PathsVector$

Rechercher l'ensemble des arcs  $L(p)$  qui composent le chemin  $p$

$taille =$  nombre d'arcs dans  $L(p)$

**pour**  $a = 0$  à  $taille$  **faire**

supprimer l'arc  $a$  du réseau  $p$  et chercher un nouveau plus court chemin  $SP_1$  entre  $s$  et  $d$ (Dijkstra)

$PathsVector = PathsVector \cup SP_1$

**pour**  $b = a + 1$  à  $taille$  **faire**

supprimer l'arc  $b$  du réseau et chercher un nouveau plus court chemin soit  $SP_2$  entre  $s$  et  $d$ (Dijkstra)

$PathsVector = PathsVector \cup SP_2$

réinsérer l'arc  $b$  dans le réseau

**finpour**

réinsérer l'arc  $a$

**finpour**

$indice \leftarrow indice + 1$

$n \leftarrow n + 1$

**fantant que**

---

Un premier exemple est donné à la figure 3.9 pour un conflit de type MN1.

Nous supposons que le véhicule 1 est affecté à la tâche courante  $T$  alors que le véhicule 2 a déjà été affecté à une tâche précédente. Lors de son cheminement dans le réseau, le véhicule 1 croise le véhicule 2 au noeud 4 au temps  $t=3$ , causant un conflit.

Nous remplaçons alors le chemin (7-14-4-8-9) par le chemin (7-15-4-12-9), qui

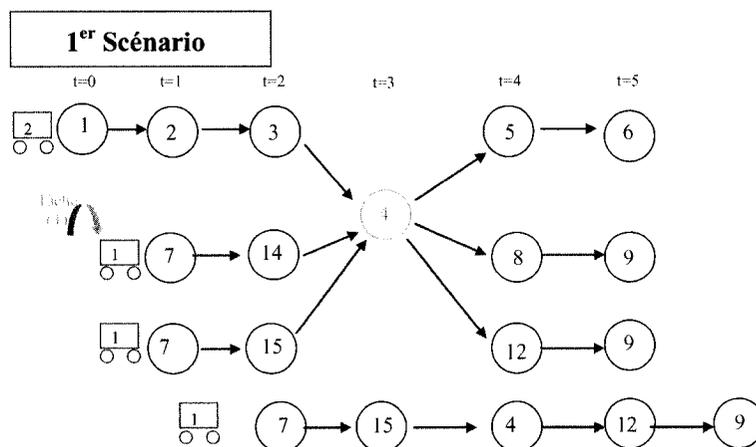


Figure 3.9 Stratégie de résolution des conflits MN1

est également un chemin plus court de longueur 4.

Le conflit n'est toutefois pas résolu puisque le véhicule 1 passe toujours par le noeud 4 au temps  $t=3$ .

On retarde alors le départ du véhicule 1 d'une période afin de résoudre le conflit (son temps de début devient  $t=2$ ).

Dans le second exemple à la figure 3.10, le conflit qui se produit entre le véhicule 1, affecté à la tâche courante, et le véhicule 2 ne peut pas être résolu en retardant le départ du véhicule 1 d'une période. En effet, le conflit observé au noeud 2 au temps  $t=3$  persiste.

Pour résoudre le conflit, il faut remplacer le plus court chemin (1-2) par un autre chemin parmi les  $k$  plus courts, en l'occurrence (1-0-3-5-2). Il faut noter que ce chemin est plus long que le chemin original (1-2).

### Résolution des conflits mineurs

L'approche générale de résolution de ces conflits mineurs se fait en deux étapes.

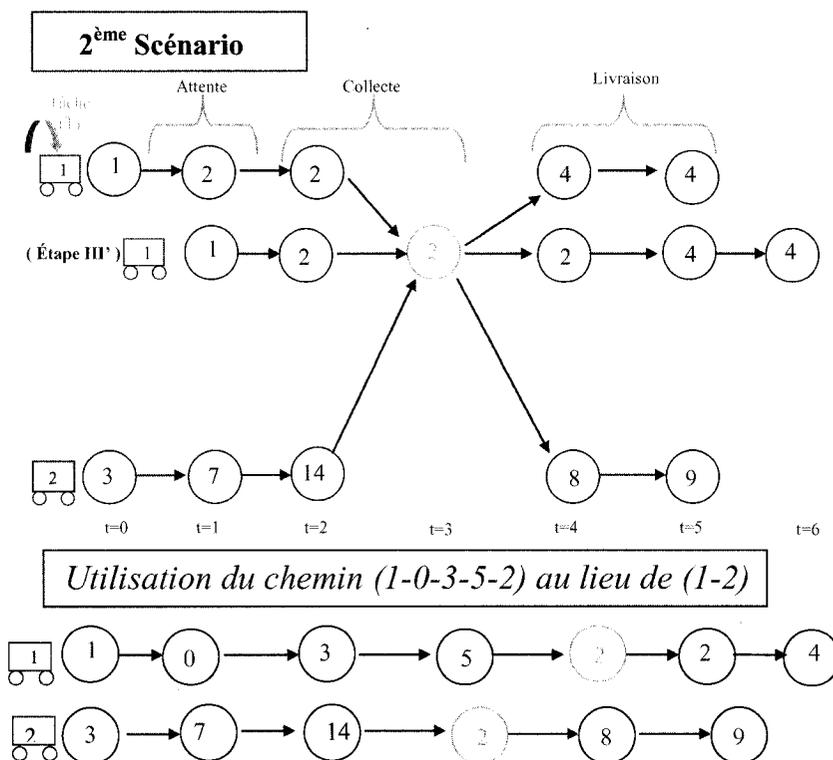


Figure 3.10 Stratégie de résolution des conflits MN1(2)

a) Première étape

On tente un simple déplacement de l'un des véhicules en attente à un autre noeud afin de laisser passer l'autre véhicule. Ce noeud ne doit pas engendrer de conflits avec d'autres véhicules (voir l'algorithme 3.7).

b) Deuxième étape

Si la première étape est infructueuse, alors la stratégie plus lourde développée pour les conflits majeurs (voir l'algorithme 3.8) est appliquée au véhicule auquel est affectée la tâche courante.

L'algorithme 3.9 présente ces étapes sur les trois types de conflits mineurs.

1. Résolution des conflits mN1(1-0)

La stratégie de résolution est appliquée au véhicule dont l'état est égal à

---

**Algorithme 3.7** Rechercher Noeud Alternatif (Véhicule, MomentConflit  $t$ , Noeud-Conflit)
 

---

```

substitution= Faux
Différent = Vrai
tant que ( Il y a des noeuds adjacents (NoeudAlternatif) à NoeudVéhicule( $t-1$ )
faire
  pour tout les positions Pos des autres véhicules à l'instant  $t$  d'occurrence du
  conflit faire
    si (NoeudAlternatif = Pos) alors
      Différent= Faux
    sinon
      Incréments Pos
    finsi
  finpour
  si (Différent = Vrai) et (NoeudAlternatif est adjacent à NoeudVéhicule( $t+1$ )
  ) et (NoeudAlternatif n'engendre pas un conflit sur un segment à la position
   $t-1$  et  $t$ ) et (NoeudAlternatif n'engendre pas un conflit sur un segment à la
  position  $t$  et  $t+1$ ) alors
    NoeudConflit = NoeudAlternatif
    substitution = Vrai
  sinon
    Passer à un autre NoeudAlternatif
  finsi
fintant que
retourner(substitution)

```

---

zéro, soit celui qui n'est pas affectée à la tâche courante.

Un exemple est donné à la figure 3.11. Dans cette figure, le véhicule 1 est affecté à la tâche courante et entre en conflit avec le véhicule 2 à l'instant  $t = 3$  au noeud 2. À cet instant, le véhicule 1 effectue une collecte (état=1) alors que le véhicule 2 est en attente afin de commencer une activité (état=0). On substitue alors le noeud conflictuel 2 par le noeud 3 dans le cheminement du véhicule 2 tout en s'assurant que cette substitution n'induit pas un autre conflit.

Le véhicule 2 passera donc maintenant par le noeud 3 et son cheminement devient (7-14-3-2-9-9) plutôt que (7-14-2-2-9-9). Ce changement permet de laisser passer le véhicule 1 au noeud 2.

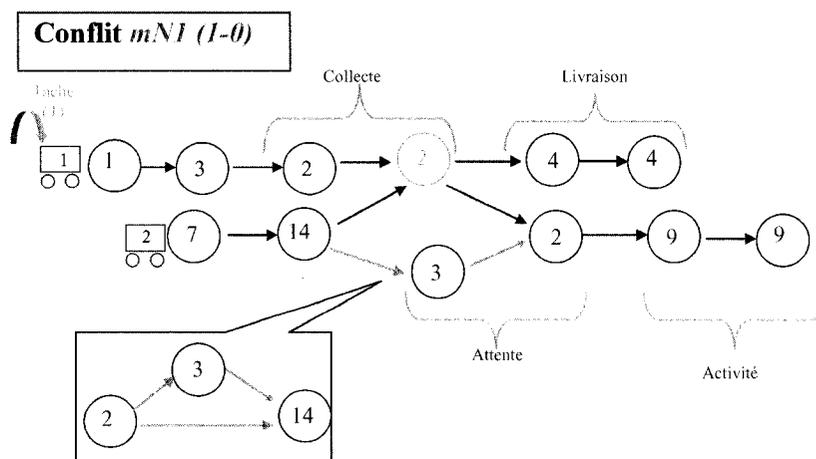


Figure 3.11 Stratégie de résolution des conflits  $mN1$

## 2. Résolution des conflits $mN2(0-0)$

Ici, on commence par chercher un autre noeud pour le véhicule auquel est affectée la tâche courante. Si cette tentative échoue, on fait une tentative avec l'autre véhicule. Enfin, en cas d'échec, on applique la stratégie de résolution pour les conflits majeurs au véhicule auquel est affectée la tâche courante.

Un exemple est donné à la figure 3.12. Dans cet exemple, le cheminement (1-3-2-2-4-4) est remplacé par (1-3-2-5-4-4) dans le cas du véhicule 1 et le cheminement (7-14-2-2-9-9) est remplacé par (7-14-3-2-9-9) dans le cas du véhicule 2.

On commence d'abord par remplacer le noeud conflictuel 2 par un noeud de substitution pour le véhicule 1, soit le noeud 5.

Le conflit n'étant pas résolu, on remplace alors le noeud 2 par le noeud 3 dans le cheminement du véhicule 2. Ceci est possible puisque les deux véhicules sont en attente au noeud conflictuel (état=0).

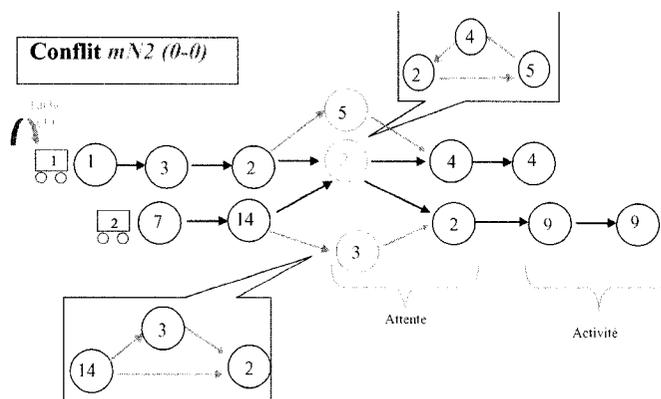


Figure 3.12 Stratégie de résolution des conflits mN2

### 3. Résolution des conflits mN3(0-1)

La stratégie est appliquée au véhicule dont l'état est égal à zéro, soit celui auquel est affectée la tâche courante.

Un exemple est donné à la figure 3.13. Le véhicule 1, qui est affecté à la tâche courante, est en attente au noeud 2 ce qui cause un conflit avec le véhicule 2 qui passe par ce noeud. Comme le véhicule 1 est en attente (état=0), on peut le déplacer sur un noeud adjacent pour résoudre le conflit, soit le noeud 3. Ainsi, le cheminement du véhicule 1 devient (1-2-2-3-4-4) plutôt que (1-2-2-2-4-4), tandis que le véhicule 2 conserve le même cheminement.

### 3.4 Stratégie de résolution des conflits résiduels

Les stratégies de résolution des conflits présentées précédemment ne concernent que les véhicules affectés à une tâche.

Mais si un véhicule *AGV1* entre en conflit avec un autre véhicule *AGV2* qui est en

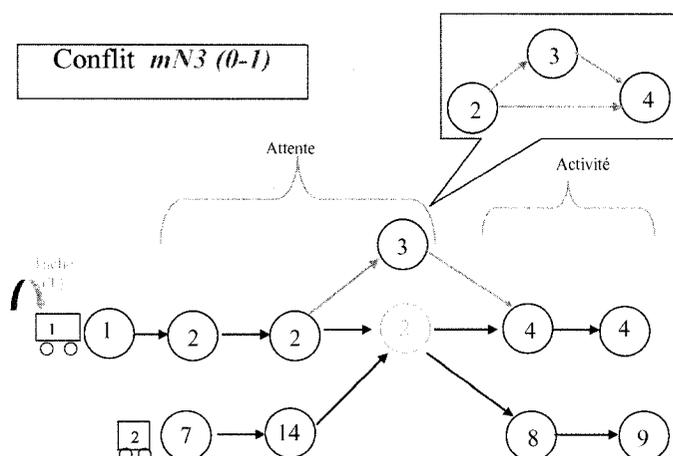


Figure 3.13 Stratégie de résolution des conflits mN3

attente d'être affecté, alors le conflit n'est pas encore résolu mais pourrait l'être un peu plus tard puisque la position du véhicule *AGV2* est temporaire et est susceptible de changer lors de la prochaine affectation.

Dans certains cas, toutefois, cette position ne changera pas. Le conflit demeurera donc non résolu jusqu'à la fin de l'affectation.

Dans la figure 3.14, nous procédons à l'insertion de quatre tâches, de façon séquentielle. Nous disposons de trois véhicules *AGV1*, *AGV2* et *AGV3* dont les noeuds de départ sont les noeuds 10, 14 et 18, respectivement.

Dans chaque colonne, nous avons la position du véhicule à chaque période. Par exemple, le véhicule *AGV1* effectue la collecte associée à la première tâche au noeud 10 au temps  $t=0$  (en rouge). Ensuite, le véhicule attend 6 unités de temps au noeud 12, et commence la livraison au temps  $t=8$  à ce même noeud (en rouge).

Les autres véhicules demeurent à leur noeud de départ durant tout ce temps.

Ensuite, la deuxième tâche sera affectée au véhicule *AGV2*, la troisième au véhicule *AGV3* et la quatrième au véhicule *AGV2*.

Lors de l'affectation de la deuxième tâche au véhicule *AGV2*, un conflit se produit au noeud 12 avec le véhicule *AGV1* au temps  $t=11$ .

Ce conflit peut être résolu si le véhicule *AGV1*, qui attend d'être affecté à une tâche, est effectivement affecté à une autre tâche et change de position. Sinon, le conflit demeure et on parle alors de conflit résiduel.

Ces conflits résiduels sont présents à la fin de l'affectation de l'ensemble des tâches. Ils sont nécessairement de type *mN1*, *mN2* ou *mN3* puisqu'au moins un des deux véhicules impliqués est en attente.

La stratégie adoptée pour résoudre ces conflits se limite toutefois à la première étape de résolution des conflits mineurs sur un noeud. On ne considère donc pas une réaffectation des tâches aux véhicules vu qu'aucun retour arrière n'est permis une fois que toutes les tâches sont insérées (voir l'étape 7. de l'algorithme 3.1).



---

**Algorithme 3.8** Résoudre Conflits Majeurs (VéhiculeCourant AGV, NoeudCollecte, NoeudLivraison) [Étape 5. de l'algorithme 3.1]

---

```

i ← 1
j ← 1
compteur ← 0
n ← 0
m ← 0
tant que (ConflitMajeur = Vrai) faire
  si (i < Nombre PlusCourtsChemins (noeudDépart(AGV), noeudCollecte))
  alors
    substituer le ie plus court chemin entre (noeudDépart(AGV), noeudCollecte)
    Incréments i
  sinon si (j < Nombre PlusCourtsChemins(noeudCollecte, noeudLivraison))
  alors
    substituer le je plus court chemin entre (noeudCollecte, noeudLivraison)
    Incréments j
  sinon si (compteur = 0) alors
    Incréments tempsDébut(AGV) d'une unité de temps
    Incréments compteur
  sinon si (n < Nombre k-PlusCourtChemin (noeudDépart(AGV), noeudCollecte)) alors
    substituer le ne k-PlusCourtChemin entre (noeudDépart(AGV), noeudCollecte)
    Incréments n
  sinon si (m < Nombre k-PlusCourtChemin (noeudCollecte, noeudLivraison))
  alors
    substituer le me k-plus court chemin entre (noeudCollecte, noeudLivraison)
    Incréments m
  sinon
    Incréments tempsDébut(AGV) d'une unité de temps
  fin si
  Recalculer le coût
  détecter les conflits
fin tant que

```

---

---

**Algorithme 3.9** Résolution des conflits mineurs [Étape 6. de l'algorithme 3.1]
 

---

```

si ( $mN1 = \text{Vrai}$ ) alors
  si ( Rechercher Noeud Alternatif (VéhiculeNonCourant, MomentConflit t,
  NoeudConflit) = Vrai) alors
     $mN1 = \text{Faux}$ 
  sinon
    Résoudre Conflits Majeurs (VéhiculeCourant, NoeudCollecte, NoeudLivrai-
    son)
  finsi
finsi
si ( $mN2 = \text{Vrai}$ ) alors
  si ( Rechercher Noeud Alternatif (VéhiculeCourant, MomentConflit t, Noeud-
  Conflit) = Vrai) alors
     $mN2 = \text{Faux}$ 
  sinon si ( Rechercher Noeud Alternatif (VéhiculeNonCourant, MomentConflit
  t, NoeudConflit) = Vrai) alors
     $mN2 = \text{Faux}$ 
  sinon
    Résoudre Conflits Majeurs (VéhiculeCourant, NoeudCollecte, NoeudLivrai-
    son)
  finsi
finsi
si ( $mN3 = \text{Vrai}$ ) alors
  si (Rechercher Noeud Alternatif (VéhiculeCourant, MomentConflit t, Noeud-
  Conflit) = Vrai) alors
     $mN2 = \text{Faux}$ 
  sinon
    Résoudre Conflits Majeurs(VéhiculeCourant, NoeudCollecte, NoeudLivrai-
    son)
  finsi
finsi

```

---

## CHAPITRE 4

### APPLICATIONS ET RÉSULTATS

Dans ce chapitre, nous présentons les problèmes sur lesquels ont été réalisés nos tests et nous rapportons les résultats obtenus.

Les 15 problèmes utilisés proviennent de Correa et al. (2005). Le nombre de tâches varie entre 7 et 12 et le nombre de véhicules varie entre 2 et 6.

On distingue trois catégories de problèmes selon la densité spatiale des tâches (voir tableau 4.1). Plus précisément:

- Les problèmes compacts: ce sont les problèmes où toutes les tâches sont concentrées dans la même région de l'atelier de production.
- Les problèmes dispersés: ce sont les problèmes où les tâches sont éloignées les unes des autres et sont dispersées dans tout l'atelier de production.
- Les problèmes mixtes: il s'agit de problèmes présentant à la fois les caractéristiques des problèmes compacts et dispersés.

Le tableau 4.2 présente un problème pour chacune des trois catégories. La première colonne est le numéro du problème, la deuxième colonne contient la liste des tâches, les troisième et quatrième colonnes correspondent aux noeuds de collecte et de livraison, respectivement (où la paire  $(x, y)$  représente le numéro du noeud  $x$  et son temps de début au plus tôt  $y$ ). La cinquième colonne contient le temps de service au noeud de livraison. Les relations de précedence sont exprimées dans la sixième colonne. Ainsi,  $Dx-Py$  signifie que la livraison (D) associée à la tâche  $x$  doit être réalisée avant la collecte (P) associée à la tâche  $y$ . La dernière colonne donne le noeud de départ de chacun des six véhicules.

Chaque problème a été résolu en utilisant de 2 à 6 véhicules.

Tableau 4.1 Description des problèmes

Problème	Catégorie	Nombre de tâches	Nombre de relation de précedence
1	Compacte	8	7
2	Compacte	10	9
3	Compacte	8	7
4	Compacte	9	7
5	Compacte	10	4
6	Dispersé	8	1
7	Dispersé	10	2
8	Dispersé	8	3
9	Dispersé	8	5
10	Dispersé	8	9
11	Dispersé	8	9
12	Mixte	7	8
13	Mixte	12	9
14	Mixte	12	6
15	Mixte	13	7

Les tests ont été réalisés sur un ordinateur avec un processeur Dual Core 2,5 GHz. Notre heuristique est implémentée en C++. Les résultats sont présentés dans les tableaux 4.3 à 4.7.

Tous les temps de calcul sont en secondes. Il est à noter qu'un temps de calcul de 30 secondes est nécessaire afin de calculer les  $k$ -plus courts chemins entre les sommets du graphe avant de commencer l'algorithme. Ce temps n'est pas inclus dans les tableaux, puisque ces chemins plus courts ne sont calculés qu'une seule fois pour un atelier donné.

Les résultats sont comparés à ceux de l'algorithme exact décrit dans Correa et al. (2005).

Pour chaque problème  $A$  et pour un nombre de véhicules fixé à  $B$  ( $A:B$ ), le retard cumulatif trouvé par notre méthode heuristique est affiché dans la deuxième colonne. Le résultat de la méthode exacte dans Correa et al. (2005) est rapporté dans la troisième colonne. Le temps de calcul de notre heuristique (en secondes) et celui de la méthode exacte dans Correa et al. (2005) sont présentés dans les quatrième et

Tableau 4.2 Description des problèmes

No du problème	No de tâche	Collecte (noeud, temps)	Livraison (noeud, temps)	Temps de service	Relations de précedence	Noeud initial
3	1	(15,0)	(13,5)	12	D1-P5	AGV1 :15
	2	(15,12)	(13,19)	12	D2-P6	AGV2 :14
	3	(15,24)	(13,33)	12	D3-P7	AGV3 :19
	4	(15,36)	(13,47)	12	D4-P8	AGV4 :27
	5	(13,17)	(25,28)	10	P5-D2	AGV5 :12
	6	(13,31)	(25,41)	10	P6-D3	AGV6 :8
	7	(13,45)	(25,55)	10	P7-D4	
	8	(15,24)	(13,33)	12		
15	1	(15,0)	(13,5)	10	D1 -P3	AGV1:15
	2	(15,10)	(13,17)	10	D2 -P4	AGV2 :14
	3	(13,15)	(19,21)	24	D1 -P3	AGV3 :19
	4	(13,37)	(19,57)	24	D5 -P8	AGV4 :27
	5	(13,0)	(25,18)	40	D3-P9	AGV5 :12
	6	(13,60)	(25,78)	40	P10-D7	AGV6 :25
	7	(25,16)	(1,29)	10	D10-P11	
	8	(25,79)	(1,92)	10	D11-P12	
	9	(19,56)	(7,64)	18		
	10	(1,43)	(27,55)	10		
	11	(27,62)	(3,86)	10		
	12	(3,106)	(5,110)	20		
	13	(3,37)	(5,41)	20		
13	1	(15,0)	(13,5)	10	D1-P5	AGV1 :15
	2	(15,12)	(13,27)	10	D2-P6	AGV2 :14
	3	(15,24)	(13,49)	10	D3-P7	AGV3 :19
	4	(15,36)	(13,71)	10	D4-P8	AGV4 :27
	5	(13,25)	(19,31)	14	D5-P9	AGV5 :12
	6	(13,47)	(19,57)	14	P6-D3	AGV6 :25
	7	(13,69)	(19,81)	14	P7-D4	
	8	(13,91)	(19,107)	14	D10-P11	
	9	(19,55)	(7,100)	18		
	10	(13,0)	(25,18)	18		
	11	(25,0)	(1,18)	10		
	12	(3,30)	(5,38)	10		

Tableau 4.3 Problèmes à deux véhicules

Instance	Retard		Temps	
	Amira et al.	Correa et al. (2005)	Amira et al.	Correa et al. (2005)
1:2	0	0	0.782	17.45
2:2	7	4	1.14	57.25
3:2	26	14	0.953	102.94
4:2**	18	-	0.781	-
5:2**	46	-	0.968	-
6:2**	165	2	0.843	58.26
7:2**	218	2	0.953	57.25
8:2**	233	-	1.297	-
9:2	12	12	0.672	13.65
10:2	35	-	0.625	-
11:2	13	13	0.656	44.97
12:2	6	6	0.547	11.13
13:2**	491	-	1.406	-
14:2**	276	-	2.516	-
15:2	-	-	-	-

cinquième colonnes, respectivement.

Un astérisque (\*) signifie que nous avons trouvé une solution en moins de 3 secondes, alors que la solution dans Correa et al. (2005) a été obtenue soit en plus de 500 secondes ou alors n'a jamais été trouvée après une heure de calcul.

Un tiret (-) indique que notre heuristique n'a pu trouver de solution et une double étoile (\*\*) indique que la solution a été trouvée en faisant appel à la deuxième stratégie d'ordonnancement des tâches, telle que décrite à la section 2 du chapitre 3.

Nous analysons maintenant les résultats pour chaque nombre de véhicules utilisés. Comme on peut le constater au tableau 4.3, on a pu résoudre très rapidement des problèmes à deux véhicules non résolus par la méthode exacte dans Correa et al. (2005) (voir problèmes 2, 5, 8, 13 et 14). En fait, tous les problèmes sauf un ont été résolus en moins de trois secondes. Par contre, nous avons eu de la difficulté pour résoudre certains problèmes à deux véhicules (voir problèmes 6 et 7). Cette difficulté provient surtout de la satisfaction des contraintes opérationnelles, d'où le recours à la deuxième stratégie d'ordonnancement des tâches.

Tableau 4.4 Problèmes à trois véhicules

Instance	Retard		Temps	
	Amira et al.	Correa et al. (2005)	Amira et al.	Correa et al. (2005)
1:3	-	0	-	84.79
2:3	0	0	1.14	57.25
3:3	0	0	1.093	199.48
4:3	0	0	1.28	210.54
5:3	1	1	0.937	140
6:3	2	2	0.734	212.63
7:3	2	2	0.831	229.35
8:3	2	2	0.641	198.84
9:3	8	8	0.688	234.06
10:3	8	8	0.672	84.04
11:3	10	10	0.687	88.35
12:3	6	6	1.172	45.86
13:3*	26	26	0.985	534.73
14:3	-	0	-	358.67
15:3	-	25	-	282.16

En considérant 3 véhicules (voir tableau 4.4), l'optimum est atteint pour 80% des problèmes dans des temps de calcul ne dépassant pas une seconde et demie, alors que le temps moyen dans Correa et al. (2005) est de plus de 170 secondes. En outre, les retards de livraison sont améliorés pour tous les problèmes par rapport aux résultats obtenus pour les problèmes à deux véhicules.

Avec 4 véhicules (voir tableau 4.5), les retards de livraison diminuent encore et on atteint l'optimum pour 73% des problèmes en des temps de calcul très rapides. En particulier, notre heuristique trouve une solution pour les problèmes 7 et 13 en 0,859 seconde et 2,347 secondes, respectivement, alors que la méthode exacte demande près de 1200 secondes de temps de calcul. Il devient toutefois de plus en plus difficile de trouver des cheminements sans conflits, puisque la congestion augmente. Ainsi, dans certains cas, on ne peut plus déplacer un véhicule sur un noeud adjacent afin de céder le passage à un autre véhicule. C'est pourquoi les problèmes 4, 5 et 10 (le problème 1 est déjà non résolu pour le problème à trois véhicules) n'ont pu être résolus par notre heuristique alors qu'on a réussi à les résoudre pour les problèmes à deux et trois véhicules. Le problème est particulièrement aigü pour les problèmes

Tableau 4.5 Problèmes à quatre véhicules

Instance	Retard		Temps	
	Amira et al.	Correa et al. (2005)	Amira et al.	Correa et al. (2005)
1:4	-	0	-	84.79
2:4	0	0	1.422	174.72
3:4	0	0	2.484	182.75
4:4	-	0	-	424.41
5:4	-	1	-	86.05
6:4	2	2	0.75	203
7:4*	2	2	0.859	1117.05
8:4	2	2	0.891	198.84
9:4	5	5	0.734	99.84
10:4	-	8	-	158.04
11:4	9	9	0.589	129.44
12:4	6	6	0.593	75.86
13:4*	26	26	2.437	1290.94
14:4*	0	0	0.953	602.59
15:4	17	17	1.031	632

compacts où le réseau devient très congestionné dans les zones où sont concentrées les tâches.

Avec 5 et 6 véhicules (voir les tableaux 4.6 et 4.7), les retards de livraison ne changent presque plus. Donc, l'augmentation du nombre de véhicules n'améliore plus ces retards. Par contre, la congestion augmente beaucoup et il n'y a plus que 53% des problèmes pour lesquelles les solutions obtenues sont optimales. Plusieurs problèmes ne peuvent plus être résolus par notre heuristique, même en faisant appel à la deuxième stratégie d'ordonnancement des tâches.

De façon générale, les problèmes compacts sont les plus difficiles à résoudre (seulement 33 % de ces problèmes ont été résolus). Il semble que les problèmes mixtes sont quelque peu problématiques quant à la satisfaction des contraintes opérationnelles, et ce, même avec peu de véhicules (voir le problème 15 aux tableaux 4.3 et 4.4). Les problèmes dispersés sont les plus faciles à résoudre étant donné que les tâches sont bien réparties dans tout le réseau, ce qui facilite l'identification de noeuds adjacents permettant de céder le passage en cas de conflit. Les temps de calcul pour ces problèmes sont très petits.

Tableau 4.6 Problèmes à cinq véhicules

Instance	Retard		Temps	
	Amira et al.	Correa et al. (2005)	Amira et al.	Correa et al. (2005)
1:5	-	0	-	90.17
2:5	0	0	1.516	117.52
3:5	0	0	2.453	257.95
4:5	-	0	-	338.83
5:5	-	1	-	140
6:5*	2	2	0.766	1011.13
7:5	2	2	0.875	336.83
8:5	-	-	-	194.46
9:5	5	-	0.89	-
10:5	-	8	-	89.69
11:5	-	8	-	199.41
12:5	6	6	0.672	68.47
13:5	-	26	-	395.85
14:5*	0	0	1.281	625.59
15:5*	17	17	1.219	706.61

Tableau 4.7 Problèmes à six véhicules

Instance	Retard		Temps	
	Amira et al.	Correa et al. (2005)	Amira et al.	Correa et al. (2005)
1:6	-	0	-	84.57
2:6	0	0	1.36	268.11
3:6*	0	0	2.547	937.97
4:6	-	0	-	138.2
5:6	-	1	-	78.69
6:6*	2	2	0.796	833
7:6	2	2	0.844	108.55
8:6	-	2	-	120.45
9:6	3	3	0.875	229.01
10:6	-	8	-	178.03
11:6	-	8	-	93.59
12:6	6	6	0.641	123.5
13:6	-	26	-	509.81
14:6	0	0	1.657	445.27
15:6*	17	17	1.219	605.09

## CONCLUSION

L'intégration de l'affectation des tâches et du routage de véhicules autoguidés dans un atelier de production flexible est un problème fort complexe. L'heuristique que nous avons développée dans ce contexte semble toutefois apporter une réponse intéressante. Ainsi, nous observons des temps de calcul nettement inférieurs à ceux rapportés dans Correa et al. (2005) pour un nombre de véhicules variant entre 2 et 6.

Ce travail a débuté par l'élaboration d'une heuristique permettant une affectation des tâches aux véhicules qui respecte le temps de début de chaque tâche et les contraintes d'antériorité qui existent entre ces tâches. Dans un deuxième temps, l'heuristique permettant de résoudre les conflits de routage a été développée. Cette dernière fait appel, au besoin, à l'heuristique d'affectation afin de modifier l'affectation courante. Différents types de conflits sont distingués selon le lieu d'occurrence et l'état des véhicules, et une stratégie de résolution propre à chaque type a été développée. Enfin, nous nous sommes également intéressés à la satisfaction des contraintes opérationnelles.

Lors de ce travail, nous avons obtenu des solutions optimales pour 71% des instances considérées, dans des temps de calcul nettement inférieurs à ceux rapportés dans Correa et al. (2005). Nous avons aussi obtenu des solutions pour des problèmes non résolus par la méthode exacte dans Correa et al. (2005), surtout dans le cas des problèmes à deux véhicules. En revanche, certains problèmes n'ont pu être résolus par notre heuristique, et ce, pour deux raisons principales.

- l'impossibilité de trouver une affectation des tâches aux véhicules qui respecte à la fois les contraintes opérationnelles et d'antériorité;
- l'impossibilité de trouver un routage sans conflit des véhicules, dû à la nature peu dense du réseau considéré.

Des écarts importants de retard entre certains résultats obtenus par notre méthode (surtout pour les problèmes à deux véhicules) et les résultats obtenus par Correa et al. (2005) sont reportés, surtout en appliquant la deuxième stratégie d'ordonnement. En fait, cette stratégie permet de satisfaire les contraintes opérationnelles en déplaçant la tâche qui viole ces contraintes, ce déplacement permet de retarder cette tâche et donc retarder toutes les autres tâches qui la suivent. Ceci engendre l'accumulation du retard, donc le retard que nous aurons est très loin de l'optimal.

Nous venons de résumer en quelques lignes les principales problématiques abordées dans ce projet de maîtrise. Les avenues de recherche à considérer en rapport avec ce projet sont multiples. Mentionnons simplement:

- Pour les problèmes où notre heuristique n'a pu produire une solution optimale, il serait envisageable d'utiliser cette solution comme solution de départ pour l'algorithme proposé dans Desaulniers et al. (2003), afin d'en accélérer la convergence.
- L'intégration de notre heuristique au sein d'une stratégie de recherche de type métaheuristique. On pourrait ainsi exploiter différents ordonnancements des tâches respectant les contraintes d'antériorité afin de développer une approche de type multi-start. Nous pourrions aussi utiliser notre heuristique comme méthode de construction dans un algorithme de recherche à voisinage élargi.
- L'application de notre heuristique à des problèmes de gestion de terminaux de conteneurs où l'on retrouve davantage de tâches et de véhicules.

## RÉFÉRENCES

BOZER, Y.A. et SRINIVASAN, M.M. (1992). Tandem configurations of autoguided vehicles systems and the analysis of single-vehicle loops. *IIE Transactions*, 23(1), 72-82.

BRODBENT, A.J., BESANT, C.B., PEMI, S.K. et WALKER, S.P., (1985). Free ranging AGV systems: promises, problems, and pathways. *Dans Proceedings of the Second International Conference on Automated Materials Handling, (ISF-Springer)*, 221-237.

CORREA, A.I., LANGEVIN, A. et ROUSSEAU, L.M. (2005a). Scheduling and Routing of Automated Guided Vehicles : A Hybrid Approach. *Computers and Operations Research*, 34, 1688-1707.

DANIELS, S.C. (1988). Real-time conflict resolution in automated guided vehicles scheduling. *PhD thesis, Department of Industrial Engineering, Pennsylvania State University, USA.*

DESAULNIERS, G., LANGEVIN, A., RIOPEL, D. et VILLENEUVE, B. (2003). Dispatching and Conflict-Free Routing of Automated Guided Vehicles : An Exact Approach. *International Journal of Flexible Manufacturing Systems*, 15, 309- 331.

EGBELU, P.J. et TANCHOCO, J.M.A. (1986). Potentials for bidirectional guide-path for automated guided vehicle based systems. *International Journal of Flexible Manufacturing Systems*, 24(5), 1075-1097.

EGBELU, P.J. (1993). Positioning of automated guided vehicle in a loop layout to improve response time. *Advances in Distribution Logistics*, Springer, Berlin, 197-214.

KIM, K.H. et TANCHOCO J.M.A. (1991). Conflict-free shortest-time bi-directional AGV routing. *International Journal of Production Research*, 29, 2377-2391.

KIM, K.H. et TANCHOCO J.M.A. (1993). Operational control of bi-directional autoguided vehicle systems. *International Journal of Production Research*, 31, 2123-2138.

LANGEVIN, A., LAUZON, D. et RIOPEL, D. (1996). Dispatching, Routing and Scheduling of Two Automated Guided Vehicles in a Flexible Manufacturing System. *International Journal of Flexible Manufacturing Systems*, 8, 246-262.

MACHANEY, R. (1995). Modelling battery constraints in discrete event automated vehicles simulations. *European Journal of Operational Research*, 33, 3023-3040.

QIU, L., HSU, W.J. et WANG, H. (2002). Scheduling and Routing Algorithms for AGVS: A Survey. *International Journal of Flexible Manufacturing Systems*, 40, 745-760.

TAGHABONI, F. et TANCHOCO J.M.A. (1995). Comparaison of dynamic routing techniques for automated guided vehicle systems. *International Journal of Production Research*, 33, 2653-2669.

TUAN L.A., M.B.M. et DEKOSTER, R. (2006). A review of design and control of automated guided vehicle systems. *European Journal of Operational Research*, 171, 1-23.

VEN DER MEER, J.R. et DEKOSTER, R. (2000). Using multiple load vehicles for internal transport with arrivals of loads. *International Journal of Flexible Manufacturing Systems*, 71(1), 32-44.

YEN, J.Y. (1971). Finding the k shortest loopless paths in a network. *Management sciences*, 17, 712-716.