Les Cahiers du GERAD ISSN : 0711–2440

Ordonnancement et routage intégrés d'une flotte de chariots dans une mine souterraine

A. Insa Corréa, A. Langevin, L.-M. Rousseau

G–2005–58

Août 2005

Ordonnancement et routage intégrés d'une flotte de chariots dans une mine souterraine

Ayoub Insa Corréa
André Langevin*
Louis-Martin Rousseau

Département de mathématiques et de génie industriel
École Polytechnique de Montréal
C.P. 6079, Succ. Centre-ville
Montréal (Québec) Canada H3C 3A7
{iacorrea, andre.langevin, louis-martin.rousseau}@polymtl.ca

* *et GERAD*

Août 2005

Les Cahiers du GERAD
G–2005–58

Résumé

Dans cet article, nous résolvons un problème intégré d'ordonnancement et de routage sans conflits d'une flotte de chariots dans une mine souterraine. Nous présentons un algorithme de décomposition hybride qui combine la programmation par contraintes (PC) et la programmation linéaire en nombre entiers (PLNE). Notre approche décompose le problème en deux parties : la première partie consiste en l'ordonnancement des tâches de chargement et déchargement de minerai tandis que la seconde partie s'occupe du routage sans conflits des chariots avec prise en compte de l'orientation des pelles mécaniques des chariots. Le réseau de galeries de la mine souterraine est en forme d'arbre. La partie ordonnancement est modélisée en PC avec un traitement préventif des conflits tandis que le routage sans conflits est modélisé en PLNE. En outre, le modèle de PLNE assure une orientation cohérente des chariots aussi bien aux points de chargement/déchargement que pendant leur routage. Nous avons testé notre algorithme sur trois réseaux de galeries différents pour analyser certains scénarios comme la croissance de la mine et la non disponibilité partielle ou totale de galeries suite à des événements imprévus (bris de chariots, chutes de roches, fuites d'eau etc.). Notre méthode de décomposition constitue un outil de dimensionnement de flotte de chariots dans une mine souterraine.

Mots clés : Méthodes hybrides ; programmation par contraintes ; ordonnancement et routage sans conflits ; orientation dans une mine souterraine.

Abstract

In this paper, we solve an integrated scheduling and conflict free routing problem in the context of an underground mine. We present a hybrid decomposition algorithm combining Constraint Programming (CP) and Mixed Integer Linear Programming (MILP). Our approach decomposes the problem in two parts : the first part consists in scheduling loading/dumping tasks while the second one deals with conflict free routing of vehicles that take into account the orientation of the vehicles. The mine roads networks considered are tree-shaped. The scheduling part is modeled with CP and a preventive treatment of conflicts is performed. The conflict free routing of vehicles is modeled with MILP. Furthermore, the MILP model ensures during the routing a coherent orientation of the vehicles either at loading or dumping points. We tested our algorithm on three different networks to analyze some growth scenarios of the underground mine. The total or partial non availability of mine roads due to some events (like vehicles break downs, rock falls, water leakage etc...) can be taken into account with our approach. This decomposition method is a good tool for fleet sizing in the underground mine context.

1 Introduction

La tendance actuelle dans l'industrie minière est une automatisation de plus en plus poussée des opérations dans les mines souterraines (or, fer, charbon, cuivre, uranium etc.). Elle est due d'une part à un manque de mineurs (moins d'intérêt pour ce dur métier, problèmes de santé et sécurité au travail) et, d'autre part, à un réel besoin d'alimenter plus efficace les hauts fourneaux et usines de traitement situés hors de la mine. Dans cet article, nous présentons une méthode de décomposition hybride pour résoudre un problème intégré d'ordonnancement et de routage sans conflits d'une flotte de chariots dans une mine souterraine. Il y a conflit quand deux chariots sont en compétition pour l'occupation d'un point ou pour parcourir un segment de galerie en même temps dans des directions convergentes. Les chariots sont soumis à des contraintes d'orientation de leurs pelles mécaniques pour accomplir leurs tâches de chargement et déchargement : chaque chariot doit toujours se présenter à son point de tâche avec une orientation appropriée de sa pelle mécanique. En outre, les réseaux utilisés sont bidirectionnels avec des galeries à voie unique. Notre approche de décomposition combine la programmation par contraintes (pour la partie ordonnancement) et la programmation linéaire en nombres entiers (pour la partie routage orienté sans conflits). La partie PC est un modèle combiné d'ordonnancement et transport qui permet un traitement préventif presque complet des conflits. Tandis que la partie PLNE est un modèle espace-temps qui permet le routage sans collisions des chariots. L'orientation convenable des chariots y est assurée pour l'exécution des tâches de chargement et déchargement de minerai.

Le plan de l'article est le suivant : la section 2 présente une description du problème, suivie d'une revue de littérature à la section 3. Notre approche hybride est décrite en détail à la section 4. À la section 5, les données utilisées et les résultats obtenus sont présentés. La conclusion suit à la section 6.

2 Description du problème

La mine est présentée sous forme de trois réseaux de galeries en forme d'arbre (voir la figure 1). Les points de chargement et déchargement sont situés aux extrémités des galeries (feuilles de l'arbre). Chaque point de chargement de minerai est représenté par un cercle tandis le point de déchargement est illustré par un rectangle. Dans tous les réseaux de galeries considérés, il y a un seul point de déchargement. Les réseaux sont tous bidirectionnels avec des galeries à voie unique. Les chariots peuvent être automatiques ou avec chauffeurs. L'arrêt d'un chariot n'est permis que sur les points de chargement/déchargement et d'autres points spécifiques prédéterminés. En particulier, l'arrêt n'est pas permis sur aux points d'intersection des galeries. La tâche de chaque chariot consiste à satisfaire des requêtes de transport. Chaque requête de transport consiste en un chargement de mine-

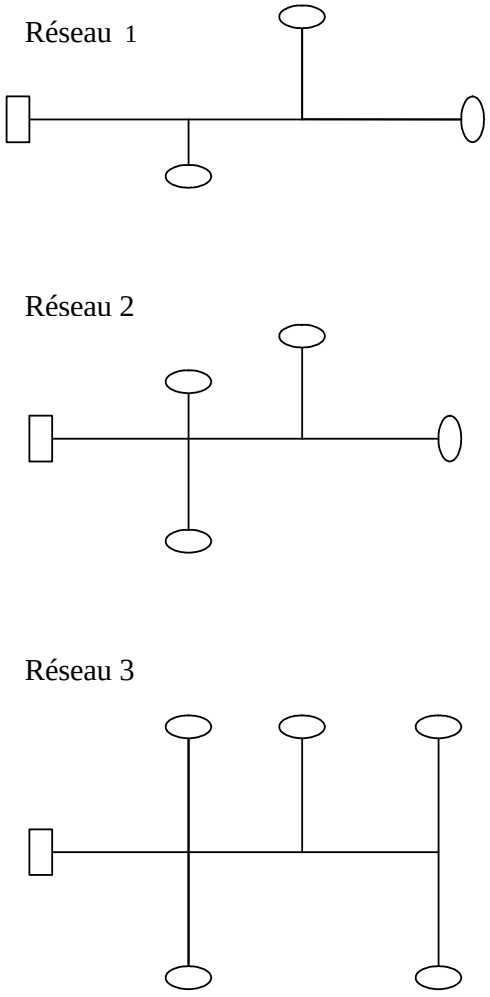

Figure 1 – Réseaux utilisés

rai, son transport et son déchargement. La quantité de minerai à collecter à chaque point de chargement est fixée *a priori* suite à des contraintes sur la teneur en minerai à obtenir au point de déchargement avant le traitement à l'usine. Cette quantité de minerai à collecter est exprimée en termes de nombre de voyages à faire à partir de chaque point de chargement. L'ordonnancement des tâches de chargement et déchargement est fait en tenant compte du routage des chariots. Puisque les chariots partagent le même réseau, des collisions peuvent survenir et doivent être évitées. Chaque chariot qui doit effectuer un chargement ou déchargement doit arriver à son point de tâche avec sa pelle orientée dans la direction appropriée c-à-d vers le fond de la galerie. Notre problème est défini de la façon suivante :

> *Étant donnés une flotte de chariots disponibles (avec leur position de départ associée) et le nombre de voyages à faire à partir de chaque point de chargement, il faut trouver l'affectation des requêtes de transport aux chariots et les routes sans conflits qui minimisent le temps de fin du dernier voyage (makespan) tout en assurant une orientation appropriée des pelles des chariots à chaque point de chargement ou déchargement.*

Ce problème peut être vu comme un problème de tournées de véhicules qui présente les caractéristiques suivantes :

- Pour chaque noeud de tâche (chargement/déchargement), il y a un nombre obligatoire de visites à faire.
- Pour chaque visite obligatoire à faire sur un noeud, le chariot utilisé doit y séjourner pendant une durée fixée.
- Il existe des contraintes de priorité entre les visites faites sur un même noeud.
- Deux chariots différents ne peuvent visiter un même noeud à la même période (contraintes d'évitement de collisions sur les noeuds).
- Deux chariots différents ne peuvent commencer en même temps la visite de certains sous ensembles d'arcs (contraintes d'évitement de collisions sur les arcs).
- Chaque chariot qui visite un noeud de tâche doit se présenter avec l'orientation appropriée.
- Entre deux visites consécutives d'un noeud de tâche, certains arcs ne peuvent être visités pendant une certaine période temps (gel temporaire de galerie pour les tâches).

3 Revue de littérature

Cet article touche au problème d'ordonnancement et de routage intégrés, aux approches hybrides exactes combinant la PC et la PLNE et au traitement de l'orientation des chariots dans une mine souterraine. Aussi, cette revue porte d'une part sur les problèmes intégrés d'ordonnancement et de routage dans les contextes d'atelier flexible et mines souterraines.

D'autre part, nous nous intéresserons au thème spécifique (et récent) des méthodes hybrides qui combinent la programmation par contraintes et la programmation linéaire en nombres entiers. Le lecteur est référé à Co et Tanchoco (1991), King et Wilson (1991), Ganesharajah et al. (1998) et Qiu et al. (2002) pour une revue générale sur les chariots automatiques. D'après la classification de Qiu et al. (2002), les algorithmes sur les chariots automatiques peuvent être classés en trois grands groupes :

- les algorithmes d'optimisation pour des réseaux quelconques.
- les algorithmes pour la conception des réseaux.
- les algorithmes d'optimisation pour les réseaux à topologie spécifique.

Peu de travaux ont été faits dans le domaine des problèmes d'ordonnancement et de routage intégrés résolus par des méthodes exactes. Langevin et al. (1996) ont proposé une méthode basée sur la programmation dynamique pour résoudre de façon exacte des instances à deux chariots dans le contexte d'un atelier flexible. Ensuite, Desaulniers *et al.* (2003) ont proposé une méthode exacte qui permet de résoudre des instances du problème de Langevin et al. (1996) en utilisant jusqu'à quatre chariots. Leur approche combine un algorithme glouton (pour trouver une solution réalisable et fixer des bornes sur les délais), la méthode de génération de colonnes et une procédure de Branch and Cut. Toutefois, leur méthode présente des limites car son efficacité dépend beaucoup de la performance de l'heuristique gloutonne de départ ; en effet si aucune solution de départ n'est trouvée alors il n'y aura pas de solution optimale. En outre, l'heuristique gloutonne perd beaucoup de son efficacité au fur et à mesure que le niveau de congestion augmente. Jusqu'en 2005, leurs résultats obtenus (en termes de nombre de chariots utilisés) étaient les meilleurs. Par la suite Corréa et al. (2005) ont résolu de façon exacte un problème apparenté à celui de Desaulniers et al. (2003). Ils ont utilisé une méthode de décomposition hybride combinant la PC et la PLNE pour résoudre des problèmes contenant jusqu'à six chariots automatiques. Les limites principales de leur approche se trouvent au niveau de leur modèle de PC et de la nature des coupes utilisées. En effet, leur modèle de PC a besoin d'être amélioré pour prévenir les conflits en amont et pour plus d'efficacité quand les requêtes de transport sont très diversifiées. Les coupes logiques ralentissent le modèle de PC du fait de leur nature disjonctive (beaucoup de noeuds de branchement de l'arbre de recherche sont créés). D'après la classification de Qiu et al. (2002), les travaux de Desaulniers et al. (2003) et Corréa et al. (2005) peuvent être classés dans la première catégorie d'algorithmes. Vu la complexité des problèmes intégrés d'ordonnancement et routage sans conflits, la tendance actuelle est à la conception d'algorithme de la troisième catégorie. Notre approche est une méthode de décomposition hybride qui fait partie de ce troisième groupe.

Dans le contexte des mines souterraines, peu de travaux ont été faits sur les problèmes intégrés d'ordonnancement et de routage sans conflits avec prise en compte de l'orientation des chariots. Parmi les rares auteurs qui se sont attaqués à l'orientation des chariots, il faut citer Gamache et al. (2004) et Bigras et Gamache (2005). Toutefois l'approche de

Gamache et al. (2004) n'est pas exacte et son graphe ne tient pas totalement en compte l'orientation des chariots sur les arcs d'intersection. Les travaux de Bigras et Gamache (2005) portent essentiellement sur le traitement exact de l'orientation pendant le calcul des plus courts chemins et pour le routage des chariots. L'orientation des chariots est importante car dans le contexte d'une mine souterraine, le réseau est si exigu qu'un chariot ne peut pas se repositionner une fois arrivé à son point de tâche. Comme Bigras et Gamache (2005), notre modélisation de la partie routage orienté sans conflits s'inspire des travaux de Krishnamurthy et al. (1993) et Vagenas (1991). Ces auteurs ont introduit des modèles qui éclatent les points d'intersection en trois ou quatre noeuds, ce qui permet de bien identifier l'orientation des chariots. Notons aussi que Beaulieu et Gamache (2004) ont proposé une stratégie globale basée sur la programmation dynamique et qui prend en compte l'orientation des chariots pour résoudre en temps réel des problèmes contenant jusqu'à quatre chariots. Deux réseaux utilisés sur trois contiennent des cycles. Toutefois leur approche donne des résultats plus satisfaisants que lorsque la stratégie chariot par chariot, implantée par Bigras et Gamache (2002), est utilisée en amont. Même avec cette approche en deux étapes, des efforts supplémentaires doivent être faits pour améliorer la qualité de la solution obtenue. Les travaux de Bigras et Gamache (2002), Beaulieu et Gamache (2004) et Gamache et al. (2004) sont des approches en temps réel. Toutefois le traitement de l'orientation qu'ils ont fait dépasse le cadre opérationnel : ils ont montré qu'il est possible de construire des réseaux qui assurent une orientation appropriée des chariots. Dans leurs approches, les tâches sont considérées selon leur ordre d'arrivée et il faut construire des routes sans conflits qui donnent une orientation correcte des chariots aux points de tâches.

L'approche proposé dans cet article est une méthode de décomposition hybride basée sur la coopération de solveurs de la PC et la PLNE. Le lecteur est référé aux manuels de Hooker (2000), Hooker et Ottosson (2003) et Milano (2004) pour une revue détaillée des différents types de méthodes hybrides et des perspectives dans ce domaine. Les méthodes hybrides de la PC et la PLNE constituent un domaine de recherche récent où diverses directions de recherche sont actuellement explorées. Ces approches visent surtout à renforcer les outils classiques de la recherche opérationnelle mais aussi à tester les cadres théoriques proposés.

4 Une approche hybride PC / PLNE

Nous présentons ici une méthode de décomposition essentiellement basée sur la coopération de solveurs de la PC (Ilog Solver et Scheduler) et de la PLNE (CPLEX). Dans cette décomposition, la PC détermine l'affectation des requêtes de transport aux chariots et les temps de début des chargements et déchargements. Aussi, le modèle de PC évite certains conflits dans les galeries de chargement ou déchargement. Pour chaque solution trouvée dans la partie PC, la PLNE cherche un routage sans conflits des chariots avec orientation

appropriée. Quand il n'y a pas de routage, une solution alternative du modèle de PC est testée. Quand il n'y a plus de solutions alternatives, la borne inférieure de la durée totale des tâches (makespan) est augmentée d'une unité et le processus reprend. Il faut préciser que le réseau mathématique est généré une seule fois avant la première utilisation du modèle de PC et les plus courts chemins entre les noeuds sont précalculés et utilisés comme des données d'entrée par le modèle de PC. Les plus courts chemins prennent en compte le changement d'orientation du chariot entre l'origine et la destination. La méthode est illustrée à la figure 2. Le modèle de PC est décrit à la section 4.1 tandis que le modèle de routage est décrit à la section 4.2. Les parties PC et PLNE sont riches sur le plan de la modélisation ; avec l'utilisation de Ilog Scheduler, la partie ordonnancement s'occupe de prévenir une grande quantité de conflits qui peuvent survenir dans la partie routage. Le traitement de l'orientation des chariots est assuré grâce à un étiquetage spécifique des noeuds du graphe mathématique associé au réseau physique.

4.1 Le modèle de PC

Le modèle de PC est un modèle intégré de transport et ordonnancement résolu avec Ilog Scheduler 6.0. Il offre l'avantage de faire un traitement préventif des conflits avec la notion de gel temporaire de galeries. Il donne une affectation ordonnée des requêtes de transport aux chariots et les temps de début des chargements et déchargement. L'utilisation de Ilog Scheduler permet une modélisation du problème avec les caractéristiques suivantes :

- Chaque requête de transport est vue comme une activité composée de trois sous activités : le chargement de minerai, son transport et son déchargement.
- Les temps de transitions minimaux entre deux chargements consécutifs au même point mais aussi entre deux déchargements consécutifs (voyages à vide) sont modélisés efficacement.
- Les contraintes de priorité entre certaines activités sont modélisées facilement.
- Une procédure de recherche qui utilise des algorithmes très efficaces d'ordonnancement par la PC et des connaissances empiriques basées sur l'analyse de l'application est disponible.
- La galerie de déchargement est modélisée comme une *ressource unaire renouvelable*. Les chariots sont des *ressources unaires renouvelables alternatives*. Une ressource unaire est une ressource qui ne peut être utilisée en même temps par deux tâches. Elle est dite renouvelable si à la fin d'une utilisation, elle devient disponible pour une autre utilisation. Des ressources sont dites alternatives si chacune d'elles peut exécuter la même tâche.
- Grâce à l'utilisation des temps de transition minimaux, toutes les galeries (chargement/déchargement) sont modélisées comme des ressources unaires renouvelables. Ainsi, des contraintes globales de la PC comme *alldifferent*, *requires*, *precedes* et *activityhasSelectedResource* sont utilisées. Une contrainte globale est une contrainte qui

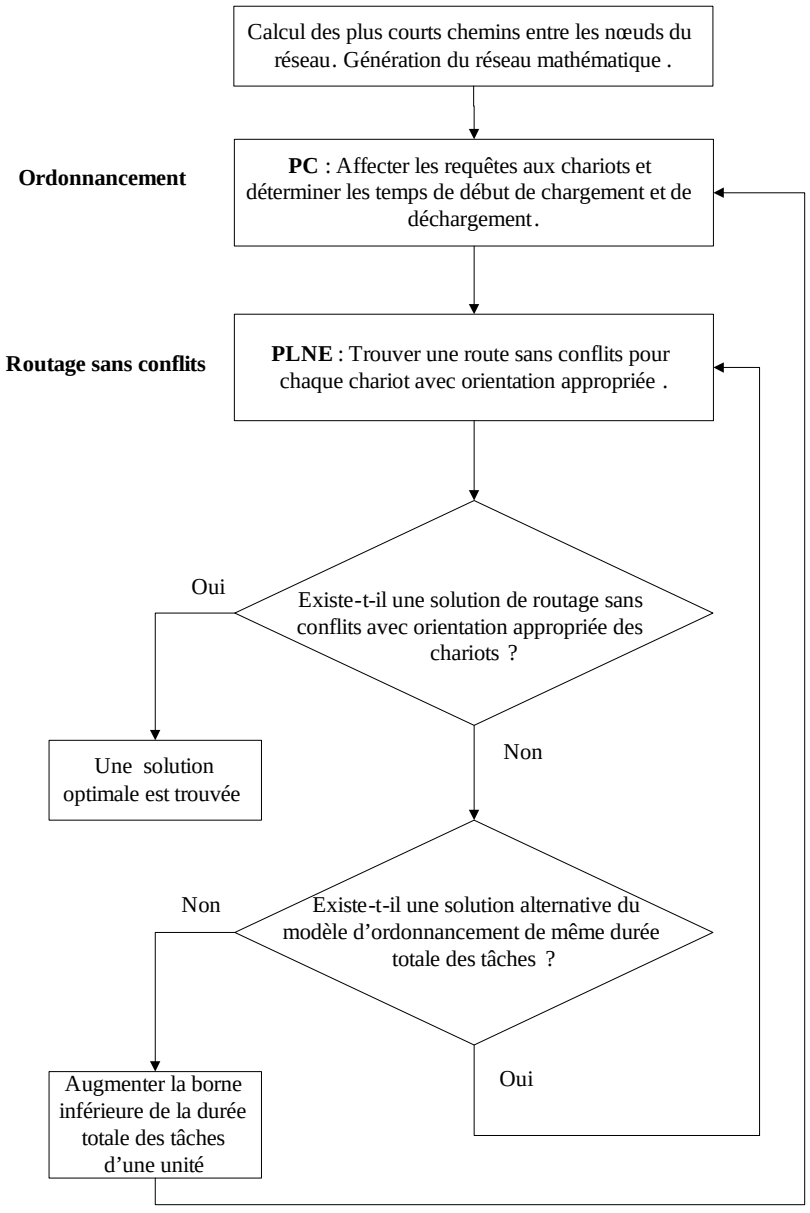

Figure 2 – Algorithme hybride

encapsule un ensemble de contraintes simples. Cet ensemble de contraintes simples est traité de façon simultanée par un algorithme spécialisé de réduction de domaines.

Le lecteur est referé à Baptiste et al. (2001) et Ilog OPL Studio (2004) pour plus de détails sur l'utilisation de Ilog Scheduler.

Le gel temporaire de galeries est fait pour les tâches de chargement ou déchargement seulement. Cela veut dire que dans la partie routage, rien n'empêche un chariot d'entrer dans une galerie gelée temporairement pour céder la place à un autre chariot dans une autre galerie et éviter ainsi un conflit.

Dans le modèle de PC, la première requête de chaque chariot est une requête fictive associée au point de départ du chariot. Les points de chargement/déchargement sont tous numérotés. En particulier, le point de déchargement est toujours numéroté 1. Chaque activité est associée à trois variables : son temps de début, sa durée (si elle n'est pas fixée comme dans notre problème) et son temps de fin. Nous avons déduit de l'analyse du problème une borne supérieure et une borne inférieure du temps de fin de toutes les tâches de chargement et déchargement. La borne supérieure (M) est la somme des longueurs des plus courts chemins entre tous les points de chargement et le point de déchargement (voyage à vide de retour pris en compte) si toutes ces tâches étaient exécutées de façon séquentielle par un seul chariot. La borne inférieure (B) est la somme de deux termes. Le premier terme correspond au temps minimal entre deux déchargements. Ce temps minimal est le temps nécessaire au premier chariot pour parcourir la galerie, faire son déchargement et sortir de la galerie avant qu'un autre chariot puisse y entrer. Le deuxième terme correspond au temps que met le chariot le plus proche de son premier point de déchargement pour parcourir le plus court chemin entre le premier point de chargement et le point de déchargement. L'orientation des chariots a été prise en compte dans l'étiquetage des noeuds et le calcul (fait en amont) des plus courts chemins entre les noeuds.

Ensembles et paramètres du modèle de PC

P	ensemble des noeuds de chargement.
V	ensemble des chariots.
n^v	noeud de départ du chariot v.
$D(\cdot,\cdot)$	matrice des plus courts chemins entre tout couple de noeuds.
R	ensemble des requêtes réelles de transport.
I	ensemble des requêtes réelles de transport et des requêtes fictives de départ.
h	durée de chaque chargement.

d	durée de chaque déchargement.
delai	délai minimal entre 2 chargements consécutifs.
l_1	longueur de la galerie de déchargement.
S	ensemble de toutes les tâches de déchargement.
N_c^p	nombre de chargements à faire au noeud de chargement p.
M	borne supérieure du début de chaque chargement avec $M = 2 * \sum_{p \in P} (D(p,1) * N_c^p)$
$t^r(\cdot,\cdot)$	matrice des temps de transitions minimaux entre 2 requêtes consécutives.
$t^l(\cdot,\cdot)$	matrice des temps de transitions minimaux entre 2 livraisons consécutives.
B	borne inférieure du temps de fin du dernier déchargement avec $B = (\|S\| - 1) * (2 * l_1) + \min_{k \in V, p \in P}(D(n^k, p) + h + D(p,1) + d)$
$tr\,[\cdot]$	tableau du nombre de voyages à faire sur chaque point de chargement.
$UnaryResource\ c[V](t^r)$	chariots munis de leur matrice m de transitions (voyages à vide).
$AlternativeResources\ vehicle\ (c)$	chariots définis comme des ressources unaires alternatives.
$UnaryResource\ g(t^l)$	la galerie de déchargement est une ressource unaire munie de la matrice de transitions t^l.

Variables du modèle de PC

$Activity\ Makespan(0)$	le temps de fin du dernier déchargement.
$Activity\ Request[r \in R]\ (h + d + D(n_r, 1))$	requête de transport (chargement, transport, déchargement).
$Activity\ Pick[r \in R]\ (h)$	sous-activité de chargement de durée h associée à la requête r.

Activity $Del[r \in R]$ (d)	sous-activité de déchargement de durée d associée à la requête r.
$v_r \in V$	variable représentant le chariot affecté à la requête r.
$s_r \in O$	variable représentant la requête satisfaite immédiatement après la requête r sur le même chariot.

La fonction objectif du modèle de PC consiste en la minimisation du temps de fin du dernier déchargement. Le modèle de PC s'écrit comme suit :

$$
\begin{aligned}
Min \quad & Makespan.end \\
s.t. \quad & B \leq Makespan.end \leq M && && (1)\\
& v_r = v_{s_r} && \forall\, r \in I && (2)\\
& alldifferent(s) && && (3)\\
& s_{r_1} = r_2 \Rightarrow Del[r1].end + D(1, n_{r2}) \\
& \qquad \leq Pick[r2].start && \forall\, r_1,\ r_2 \in R && (4)\\
& Request[r]\ requires\ vehicle && \forall\, r \in R && (5)\\
& Request[r].start = Pick[r].start && \forall\, r \in R && (6)\\
& Request[r].end = Del[r].end && \forall\, r \in R && (7)\\
& s_r \neq r && \forall\, r \in R && (8)\\
& Pick[r].start \geq \max(D(n^{v_r}, n_r), (tr[r] - 1) \\
& \qquad * (delai + h)) && \forall\, r \in R && (9)\\
& Del[r].start \geq (tr[r] - 1) \\
& \qquad * (delai + h) + D(1, n_r) && \forall\, r \in R && (10)\\
& Pick[r]\ precedes\ Del[r] && \forall\, r \in R && (11)\\
& Pick[r].end + D(1, n_r) \leq Del[r].start && \forall\, r \in R && (12)\\
& Del[r]\ requires\ g && \forall\, r \in R && (13)\\
& Request[r]\ precedes\ Makespan && \forall\, r \in R && (14)\\
& Pick[r]\ precedes\ Pick[r+1] && \forall\, r \in [1..(|S| - 1)] : n_r = n_{r+1} && (15)
\end{aligned}
$$

$$Pick[r].end + delai \leq Pick[r+1].start \quad \forall\, r \in [1..(|S|-1)] : n_r = n_{r+1} \qquad (16)$$

$$Del[r] \text{ precedes } Del[r+1] \quad \forall\, r \in [1..(|S|-1)] : n_r = n_{r+1} \qquad (17)$$

$$Del[r].end + 2 * l_1 + 1 \leq Del[r+1].start \quad \forall\, r \in [1..(|S|-1)] : n_r = n_{r+1} \qquad (18)$$

$$Request[r] \text{ precedes } Request[r+1] \quad \forall\, r \in [1..(|S|-1)] : n_r = n_{r+1} \qquad (19)$$

$$activityHasSelectedResource$$

$$(Request[r], vehicle, c[k]) \leftrightarrow v_r = k \quad \forall\, k \in V, r \in R \qquad (20)$$

La contrainte (1) assure que la solution trouvée sera dans l'horizon défini. Les contraintes (2) signifient que toute requête de transport et sa successeure doivent être satisfaites par le même chariot. Les contraintes (3) imposent que chaque requête de transport est satisfaite par un unique chariot c-à-d tous les successeurs sont différents. Les contraintes (4) déclarent que si deux requêtes se succèdent sur un même chariot alors ce chariot doit avoir assez de temps pour faire le voyage à vide. Les contraintes (5) assurent que chaque requête requiert un chariot (ressource unaire renouvelable). Les contraintes (6) signifient que le temps de début d'une requête est égal au temps de début de son chargement associé. Les contraintes (7) signifient que le temps de fin d'une requête est égal au temps de fin de son déchargement associé. Les contraintes (8) précisent qu'aucune requête ne peut être successeure d'elle même. Les contraintes (9) signifient que pour qu'un chargement puisse être fait, il faut que le chariot utilisé ait assez de temps pour arriver à son point de chargement et que les chargements antérieurs soient d'abord faits. **Ceci nous permet d'éviter d'introduire des ressources unaires supplémentaires pour les galeries de chargement**. Les contraintes (10) signifient que pour qu'un déchargement puisse être fait à partir d'un point de collecte, il faut que les précédents déchargements soient au préalable effectués et qu'il y ait assez de temps pour le transport. Les contraintes (11) précisent que tout chargement doit précéder son déchargement associé. Les contraintes (12) assurent que, entre la fin d'un chargement et le début de son déchargement associé, il y a assez de temps pour le transport. Les contraintes (13) signifient que tout déchargement requiert l'utilisation de la galerie de déchargement (ressource unaire). Les contraintes (14) imposent que toute requête soit satisfaite avant la fin du dernier déchargement. Les contraintes (15) sont des contraintes de priorité entre les chargements. Les contraintes (16) imposent une période de transition minimale entre deux chargements consécutifs. Les contraintes (17) sont des contraintes de priorité entre les chargements. Les contraintes (18) imposent une période de transition minimale entre deux déchargements. Les contraintes (19) sont des contraintes

de priorité entre deux requêtes consécutives. Les contraintes (20) sont des contraintes de liaison entre la partie transport (variables v) et la partie ordonnancement du modèle de PC. Pour des raisons de concision de la présentation nous n'avons pas inclus les contraintes touchant les requêtes fictives de départ et arrivée.

La notion de gel de galeries est prise en compte de façon implicite dans les déclarations des ressources unaires c, *vehicle* et g. Ces ressources unaires sont munies de paramètres t^r et t^l qui spécifient la période de transition minimale entre deux utilisations consécutives de la même ressource. L'activité *Del* utilise la ressource unaire g tandis que l'activité *Request* utilise l'ensemble de ressources unaires alternatives *vehicle*. En fait, avec l'activité *Request*, on a modélisé de façon implicite la contrainte logique qui impose que c'est le même chariot qui fait le chargement, le transport et le déchargement d'une même requête de transport.

Stratégie de recherche (PC)
Une stratégie de recherche en profondeur combinée à une heuristique pour spécifier la façon de parcourir l'arbre de recherche est utilisée. Dans l'heuristique, le choix des variables et activités est fait dans l'ordre suivant : v (affectation des requêtes de transport aux chariots), puis *Pick* (chargement de minerai), ensuite *Del* (déchargement de minerai). Les valeurs de v sont choisies en fonction de la proximité aux points de chargement. Les chargements et déchargements doivent commencer le plus tôt possible (on prend toujours la valeur minimale du domaine courant de chaque variable associée à ces activités). En outre, sur chaque point de chargement on essaie d'abord d'affecter deux chargements consécutifs à des chariots différents (les voyages sont longs et on veut rapprocher le plus possible les chargements pour obtenir un haut débit dans la mine et minimiser le makespan). Toutes les activités sont ordonnées sur les ressources unaires (galerie de déchargement, chariots) qu'elles requièrent. Les critères utilisés pour ordonner les sous ensembles d'activités sont leurs temps au plus tôt de début, temps au plus tard de fin et la durée de toutes les activités. Les ressources les plus contraintes sont choisies en premier.

4.2 Le modèle de PLNE

À partir d'une solution du programme de PC précédent qui fournit l'affectation des requêtes aux chariots et le temps de début des chargements et déchargement, le modèle de PLNE présenté dans cette section trouve, si elle existe, une solution de routage sans conflits ainsi que l'orientation exacte des chariots. À partir d'un réseau physique, un réseau mathématique est construit, puis utilisé dans un graphe espace-temps. Un graphe espace-temps est nécessaire pour pouvoir localiser la position de chaque chariot à tout moment. Dans ce qui suit, nous expliquons la construction du réseau mathématique pour le réseau physique numéro 2 (voir figure 1). Dans ce réseau physique, les galeries sont subdivisées en segments d'égale longueur et chaque segment est modélisé par deux arêtes. Chaque arête a deux noeuds qui définissent la même orientation des chariots (noté + si le chariot est

orienté dans une direction, noté − dans l'autre direction). Ainsi, deux réseaux parallèles sont obtenus, l'un exclusivement avec des noeuds +, l'autre uniquement avec des noeuds − (voir figure 3). Les deux réseaux sont liés uniquement au niveau des points d'intersection. Dans chaque réseau, un mode de déplacement unique est considéré (chariot orienté + ou chariot orienté −). Pour chacun de ces deux réseaux (+ et −), si quatre (resp. trois) galeries se rencontrent en un point alors ce point d'intersection est éclaté en quatre (resp. trois) noeuds adjacents. L'arrêt sur un des noeuds des intersections n'est pas permis et des points d'attente spécifiques sont choisis (ces points sont soulignés et en gras sur la figure 3). Entre deux tâches consécutives (chargement/déchargement ou déchargement/chargement), tout chariot doit changer d'orientation en passant au travers d'une intersection. La figure 4 illustre comment un chariot change d'orientation. Ainsi un chariot qui quitte le point A doit passer par le point B pour se retrouver en C avec l'orientation contraire de celle qu'il avait au point A. Ceci correspond, sur le réseau mathématique associé au chemin $A, I_1, I_2, B, I_2, I_3, C$.

Le graphe espace-temps est construit en créant pour chaque période de temps un ensemble de noeuds correspondant à tous les noeuds du réseau mathématique (voir figure 5). Dans le graphe, il existe les types de noeuds suivants :

- Les noeuds terminaux (feuilles de l'arbre). Exemple : 1^+ et 1^-.
- Les noeuds d'intersection. Exemple : 4^+, 4^-, 13^+ et 13^-.
- Les noeuds internes. Ce sont les noeuds qui ne sont ni des noeuds d'intersection ni des noeuds terminaux. Exemple : 2^+ et 2^-.

Chaque arête du réseau mathématique est transformée en deux arcs dans le graphe espace-temps pour chaque couple de périodes consécutives. Les arcs d'une période à l'autre correspondent au déplacement sur un segment du réseau physique ou à une attente en un noeud spécifique (voir figures 6 et 7 ; les arcs horizontaux correspondent à l'attente en un noeud). D'autre part, pour modéliser le passage à une intersection, des arcs de longueur nulle entre deux noeuds d'une intersection sont créés. Les figures 8 et 9 illustrent quelques arcs d'intersections du réseau 2. Des contraintes dans les modèle mathématique empêchent l'utilisation par un chariot de plus d'un arc d'intersection à chaque période, ce qui élimine la possibilité d'attente à une intersection.

Le modèle mathématique suivant est un modèle multiflux avec un type de flux par chariot. Le passage à certains noeuds du graphe espace-temps sont fixés (pour l'exécution des tâches de chargement et déchargement). Tous les arcs de galerie sont de longueur 1 tandis que les arcs d'intersection sont de longueur 0. Les chariots peuvent partir de n'importe quel noeud à la période initiale (exception faite des points d'intersection). Les données nécessaires à la compréhension du modèle sont présentées ci-dessous. Certaines données sont générés automatiquement une seule fois au début de l'algorithme hybride pour créer le réseau physique.

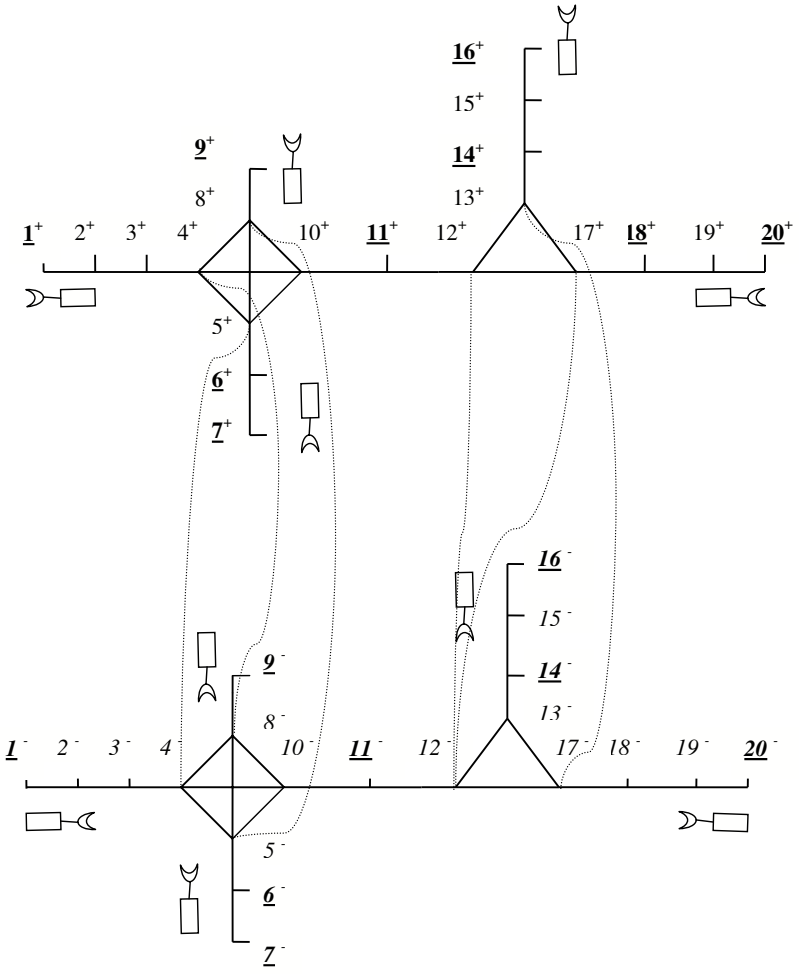

Figure 3 – Version mathématique du réseau 2

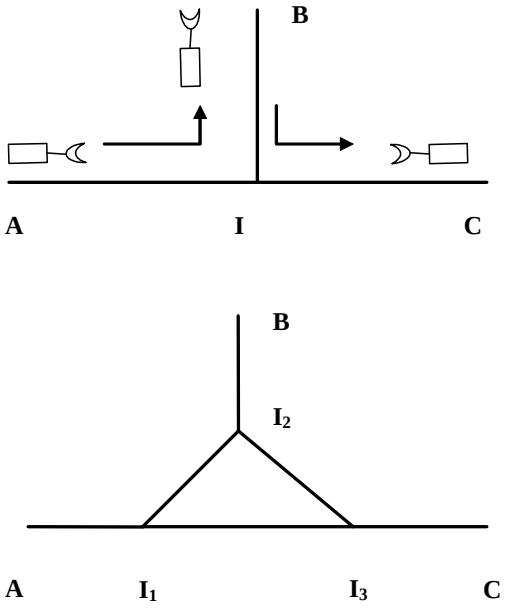

Figure 4 – Changement d'orientation

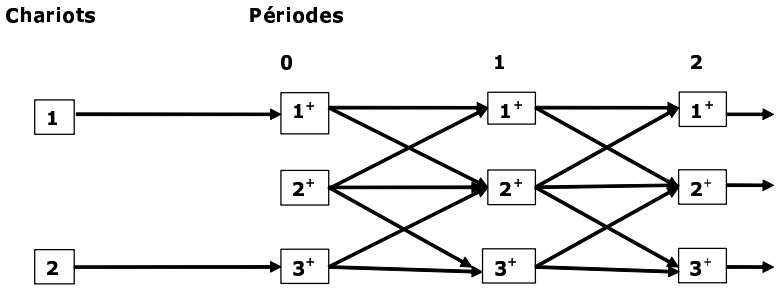

Figure 5 – Graphe espace-temps utilisé (réseau +)

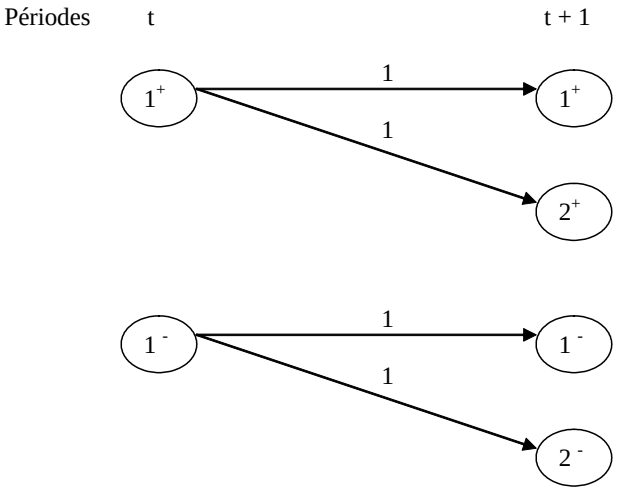

Figure 6 – Arcs issus d'un noeud terminal

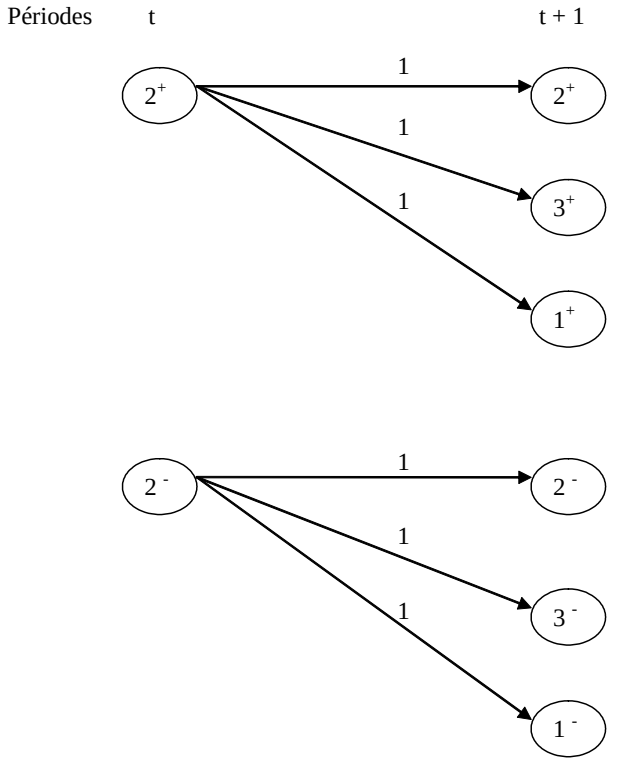

Figure 7 – arcs issus d'un noeud interne

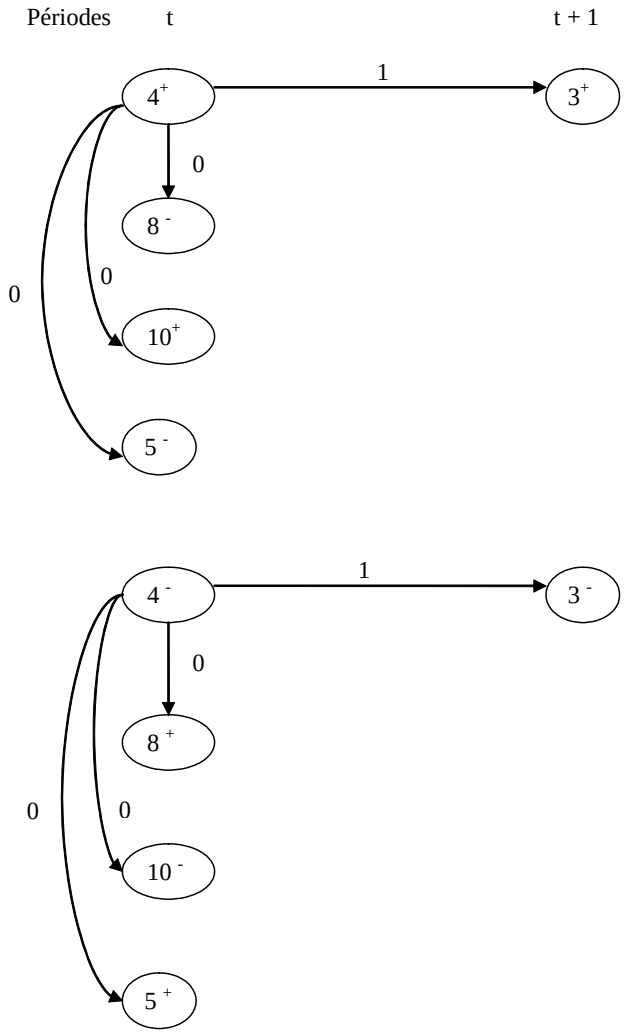

Figure 8 – arcs issus d'une intersection en carré

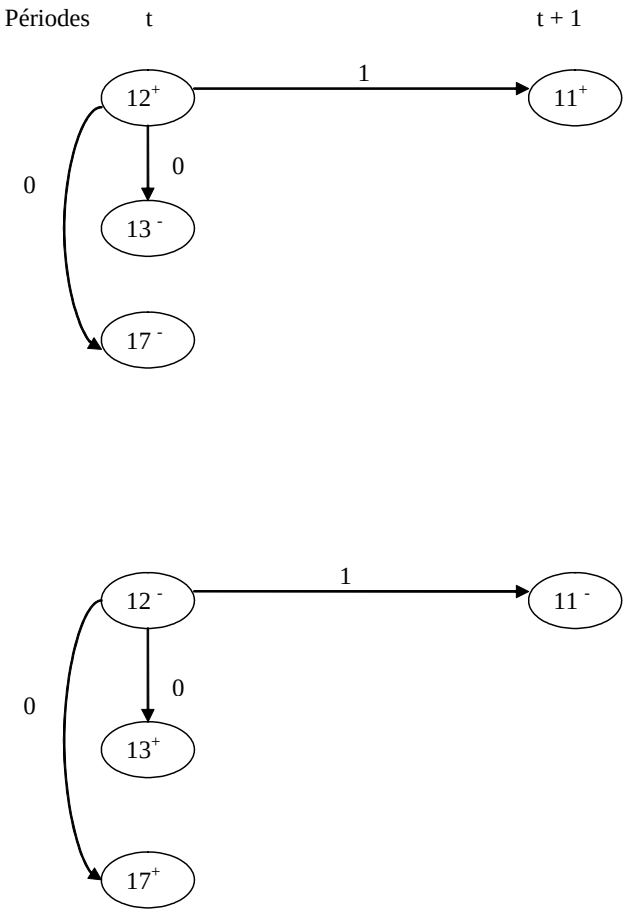

Figure 9 – arcs issus d'une intersection en triangle

Ensembles et paramètres du modèle de PLNE

U	durée totale des tâches (importée du modèle de PC).
$t^c[\cdot]$	temps de début de chargement (importé du modèle de PC).
$t^d[\cdot]$	temps de début de déchargement (importé du modèle de PC).
v_r	chariot affecté à la requête r (donnée importée du modèle de PC).
V	ensemble des chariots.
n^v	noeud de départ du chariot v.
R	ensemble des requêtes de transport.
N^g	ensemble des noeuds de galerie.
n^+	nombre de noeud étiquetés +.
N	ensemble de tous les noeuds.
N_k^i	ensemble des noeuds de l'intersection k.
A^i	ensemble des arcs de toutes les intersections.
A_k^i	ensemble des arcs de l'intersection k.
A	ensemble de tous les arcs de galerie.
$A^f[i]$	ensemble de tous les arcs sortant du noeud i.
$A_g^f[i]$	ensemble de tous les arcs de galerie sortant du noeud i.
$A_i^f[i]$	ensemble de tous les arcs d'intersection sortant du noeud i.
$A^t[i]$	ensemble de tous les arcs entrant au noeud i.
$A_g^t[i]$	ensemble de tous les arcs de galerie entrant du noeud i.
$A_i^t[i]$	ensemble de tous les arcs d'intersection entrant du noeud i.
$A^w[i]$	arc d'attente associé au noeud i.
$A_+^o[a]$	opposé de l'arc a avec origine et destination de a orientés +.
$A_-^o[a]$	opposé de l'arc a avec origine et destination de a orientés -.
$p[a]$	arc parallèle à a avec origine et destination orientés -.
$o[a]$	noeud origine de l'arc a.
$d[a]$	noeud destination de l'arc a.
$n[r]$	noeud de chargement associé à la requête r.

Variables du modèle de PLNE

$X^t_{k,a}$	variable booléenne = 1 si le chariot k débute le parcours de l'arc de galerie a à la période t et 0 sinon.
$Y^t_{k,a}$	variable booléenne = 1 si le chariot k débute le parcours de l'arc d'intersection a à la période t et 0 sinon.

Le modèle de PLNE est décrit ci-dessous. La fonction objectif du modèle de PLNE consiste en la minimisation des déplacements improductifs pour obtenir des déplacements cohérents et économiser de l'énergie. Toutefois, dans la mesure où une seule solution réalisable à ce modèle est cherchée, la solution de routage trouvée essaie d'éviter le plus possible les déplacements improductifs.

$$Min \quad \sum_{k\in V,\, t\in Pd,\, a\in Arcs:\, o[a]\neq d[a]} X^t_{k,a}$$

$$s.t. \quad \sum_{a\in A^f[n^k]} X^o_{k,a} = 1 \qquad \forall\, k \in V \tag{21}$$

$$\sum_{a\,\in\, A^t_g[i]} X^{t-1}_{k,a} + \sum_{a\,\in\, A^t_i[i]} Y^t_{k,a} = \sum_{a\,\in\, A^f_g[i]} X^t_{k,a} + \sum_{a\,\in\, A^f_i[i]} Y^t_{k,a} \qquad \forall\, i \in N, k \in V, t \in [1..(U-1)] \tag{22}$$

$$\sum_{a\,\in\, A} X^t_{k,a} = 1 \qquad \forall\, t \in [0..(U-1)], k \in V \tag{23}$$

$$\sum_{k\,\in\, V, a\,\in\, A^f_g[i]} X^t_{k,a} \leq 1 \qquad \forall\, t \in [0..(U-1)], i \in N^g \tag{24}$$

$$\sum_{k\,\in\, V, a\,\in\, A^f_g[i]} X^t_{k,a} + \sum_{k\,\in\, V, a\,\in\, A^f_i[i]} Y^t_{k,a} \leq 1 \qquad \forall\, t \in [1..(U-1)], i \in N^i_k, k \in [1..2] \tag{25}$$

$$X^{t^c[r]}_{v_r, A^w[n[r]]} = 1 \qquad \forall\, r \in R \tag{26}$$

$$X^{t^c[r]+1}_{v_r, A^w[n[r]]} = 1 \qquad \forall\, r \in R \tag{27}$$

$$X^{t^d[r]}_{v_r, A^w[1]} = 1 \qquad \forall\, r \in R \tag{28}$$

$$\sum_{k \in V, a \in A_k^i} Y_{k,a}^t \leq 1 \qquad \forall\, t \in [1..(U-1)], k \in [1..2] \tag{29}$$

$$\sum_{k \in V, a \in Arcs : d[a] \in N_k^i} X_{k,a}^t \leq 1 \qquad \forall\, t \in [1..(U-1)], k \in [1..2] \tag{30}$$

$$\sum_{k \in V} (X_{k,u}^t + X_{k,A_+^o[u]}^t + X_{k,p[u]}^t + X_{k,A_-^o[p[u]]}^t) \leq 1$$
$$\forall\, t \in [0..(U-1)], u \in G : \left(o[u] + 1 \leq d[u] \leq n^+\right) \wedge \left(o[u] \leq \left(n^+ - 1\right)\right) \tag{31}$$

$$\sum_{k \in V, a \in A_g^t[i]} X_{k,a}^t \leq 1 \qquad \forall\, t \in [0..(U-1)], i \in N^g \tag{32}$$

Les contraintes (21) spécifient la position initiale de chaque chariot. Les contraintes (22) sont des contraintes de conservation de flux. Les contraintes (23) signifient que chaque chariot doit se trouver à un endroit unique à chaque période (et aussi un chariot ne peut être dans une intersection pendant deux périodes consécutives). Les contraintes (24) imposent la présence d'au plus un chariot sur chaque arc à chaque période. Les contraintes (25) assurent qu'il a au plus un chariot sur un noeud intersection à chaque période. Les contraintes (26) et (27) signifient que chaque chariot doit être à son point de chargement au bon moment puis y rester deux périodes pour effectuer son chargement de minerai. Les contraintes (28) imposent que les chariots qui doivent décharger du minerai doivent se présenter au bon endroit au bon moment et y passer une période. Les contraintes (29) assurent qu'au plus un chariot peut être présent sur chaque intersection à chaque période tandis que les contraintes (30) spécifient que deux chariots ne peuvent entrer en même temps dans une même intersection. Les contraintes (31) sont des contraintes d'évitement de collisions sur les arcs de galerie tandis que les contraintes (32) sont des contraintes d'évitement de collisions sur les noeuds de galerie.

Stratégie de recherche (PLNE)
Dans cette partie, la stratégie de recherche utilisée est essentiellement basée sur la fixation de variables liées à des événements précis (position initiale de chaque chariot, début des chargements et déchargements). Pour toutes les variables, on se branche d'abord à droite (variable binaire fixée à 1). Nous avons aussi utilisé le simplexe dual sur le noeud racine de l'arbre de recherche suivi du simplexe réseau dans les autres noeuds tout en mettant l'emphase sur les solutions réalisables cachées (option de CPLEX 9.0).

5 Tests

Les réseaux utilisés pour les tests sont présentés en 5.1 et les résultats en 5.2 en arrière de ces choix.

5.1 Réseaux et données utilisés

Nous avons utilisé trois réseaux qui peuvent être considérés comme des agrandissements successifs d'un réseau de galeries. Le troisième réseau possède les caractéristiques suivantes qui permettent de terster les limites aussi bien du modèle de PC que du modèle de PLNE :

- Les galeries de chargement sont toutes de même longueur ce qui crée des symétries dans le modèle de PC ;
- Les galeries sont très longues, ce qui augmente la taille du modèle de PLNE ;
- Il y a un nombre accru de chariots et de requêtes de transport à satisfaire.

Le tableau 1 présente la description détaillée d'instance pour chaque réseau. Aux extrémités de chaque réseau se trouvent le noeud de déchargement (toujours numéroté 1) et les autres noeuds de chargement. Sur chaque noeud de chargement, il y a un nombre de collectes de minerai à faire, nombre évalué à l'avance (en tenant en compte du rythme de travail des mineurs). On veut faire la planification pour au plus trois collectes par noeud de chargement. Il y a un délai minimal entre deux déchargements ou chargements consécutifs à respecter. **En effet, lorsque deux chariots se suivent sur un même point de tâche (chargement/déchargement), le premier chariot doit avoir assez de temps pour exécuter sa tâche et sortir complèment de la galerie avant que le second chariot ne parcourt la galerie pour commencer sa tâche.** La transition minimale entre deux déchargements consécutifs est de cinq périodes tandis que celle entre

Tableau 1 – Description détaillée de quelques instances

Réseau utilisé	Numéro problème	Noeud et chargements	Nombre chariots	Noeud initial
1	7	6 : 2	4	Veh1 : 12
		12 : 3		Veh2 : 8
		16 : 2		Veh3 : 6
				Veh4 : 14
				Veh5 : 11
2	4	7 : 2	5	Veh1 : 15
		9 : 3		Veh2 : 38
		16 : 3		Veh3 : 2
		20 : 2		Veh4 : 6
				Veh5 : 19
3	1	10 : 3	6	Veh1 : 12
		15 : 3		Veh2 : 3
		25 : 3		Veh3 : 37
		35 : 3		Veh4 : 18
		40 : 3		Veh5 : 7
				Veh6 : 28

deux chargements consécutifs est de 15 périodes. Chaque déchargement dure une période tandis que tout chargement dure deux périodes (le chargement est habituellement plus long que le déchargement). Nous supposons qu'aucun chariot ne peut se trouver initialement sur une intersection. L'horizon choisi est variable : il dépend du nombre de chargements contenus dans chaque instance. Aussi pour chaque réseau, le nombre de chariots varie d'une instance à l'autre. Le nombre total de chariots considérés peut dépasser le nombre de noeuds de chargement d'une unité. Ces instances permettent d'analyser le cas où il existerait un chariot dédié à l'augmentation de la performance de la mine souterraine (ce chariot est surnommé flottant). Nous n'avons pas testé les instances à deux chariots car dans ce cas l'utilisation d'un système automatisé pour gérer la mine souterraine n'est pas pertinente.

5.2 Résultats obtenus

Les tests sont faits avec OPLScript 3.7 sur un ordinateur de type Pentium 4, 2.4 GHz, 512 Mo (RAM). Ilog Scheduler 6.0 est utilisé pour résoudre la partie PC (ordonnancement) et CPLEX 9.0 pour la partie PLNE (routage orienté sans conflits). 115 instances ont été générés aléatoirement et testé. Pour le réseau 1, treize problèmes avec un nombre de chariots variant de trois à quatre ont été testés. Pour le réseau 2, quinze problèmes avec un nombre de chariots variant de trois à cinq chariots tandis que pour le réseau 3, onze problèmes avec avec un nombre de chariots variant de trois à six chariots. Les résultats obtenus sont affichés dans les tableaux deux à six. Pour chaque instance et pour un nombre de chariots fixé, le temps de calcul du modèle de PC (en secondes) est présenté dans la première colonne (TPC). Celui du modèle de PLNE se trouve sur la deuxième colonne ($TPLNE$). *Duree* est la durée totale de toutes les tâches (makespan) est affiché dans la troisième colonne tandis que *nbIter* c-à-d le nombre de fois qu'au moins un conflit a été détecté dans la partie routage et est affichée dans la quatrième colonne. Un trait (-) signifie qu'aucune solution au modèle de PC n'a été trouvée au bout de 20 minutes. Tandis qu'un double trait (- -) signifie qu'aucune solution au modèle de PLNE n'a été trouvée au bout de 20 minutes.

Dans le réseau 1 (voir le tableau 2), 96 % des instances sont résolus, chacune en moins de 40 secondes. Le modèle de PC est ici particulièrement efficace car il anticipe bien sur les conflits et qu'à l'exception d'une, toutes les autres instances ont été résolues en moins d'une seconde. Dans le problème 3 avec trois chariots, seule une solution avec conflits du modèle de PC n'est évitée. Le problème 3 avec quatre chariots n'a pas été résolu car la borne inférieure du makespan est trop éloignée de sa valeur optimale, ce qui a engendré beaucoup d'itérations. L'utilisation d'un chariot flottant (problèmes avec quatre chariots) ne crée pas de congestion supplémentaire significative dans la mine. En ce qui concerne le réseau 2 (voir les tableaux 3 et 4), toutes les instances ont été en moins d'une minute et demie. Le modèle de PC est assez efficace même s'il y a sept instances où des conflits

Tableau 2 – Réseau 1 (3 et 4 chariots)

	3 chariots				**4 chariots**			
Instance	**TPC**	**TPLNE**	**Durée**	**nbIter**	**TPC**	**TPLNE**	**Durée**	**nbIter**
1	0.05	2.63	75	0	0.03	7.88	75	0
2	0.03	2.95	59	0	0.16	7.05	59	0
3	0.02	1.92	51	1	-	-		-
4	0.02	2.36	59	0	0.02	3.31	59	0
5	0.03	2.53	57	0	0.01	5.78	57	0
6	0.01	1.44	43	0	0.02	3.02	43	0
7	0.03	1.78	51	0	0.03	4.41	51	0
8	0.08	2.99	75	0	0.03	12.57	75	0
9	0.08	2.17	59	0	0.14	2.87	59	0
10	0.01	2.23	53	0	0.03	35.46	53	0
11	0.02	2.08	59	0	0.02	2.99	59	0
12	0.05	1.91	59	0	0.05	5.84	59	0
13	0.02	1.37	43	0	0.01	2.31	43	0

n'ont pas été totalement anticipés. Il s'agit des problèmes 10 et 13 avec quatre chariots, des problèmes 9–10–11–12–13 avec cinq chariots. L'utilisation du chariot flottant engendre au passage une légère augmentation de la congestion dans la mine mais toutes les instances associées sont résolues. Dans le réseau 3 (voir les tableaux 5 et 6), 47 % des problèmes avec trois chariots n'ont pas été résolus. les symétries générées par la nature spécifique du réseau 3 font qu'il est difficile de trouver un ordonnancement pour ces problèmes. Lorsque de chariots est augmenté à quatre alors tous les problèmes sont résolus avec une anticipation complète des conflits dans la partie PC. Lorsque le nombre de chariots disponibles passe à cinq alors 72 % des problèmes sont résolus. En ce qui concerne les deux problèmes non résolus (problèmes 2 et 11 avec cinq chariots), un routage réalisable n'a pas été trouvé car on est dans la situation où une solution à la relaxation linéaire est trouvée sans qu'une solution entière ne soit disponible. Notons que le modèle de PC tourne relativement vite. Il faut aussi mentionner que le modèle de PC n'a pas pu anticiper sur tous les conflits. Enfin lorsque le chariot flottant est introduit (problèmes avec six chariots), le pourcentage de problèmes résolus est très bas (environ 27%). Ici le réseau est très congestionné (solution de routage très difficile à trouver). En outre dans les instances résolues, tous les conflits n'ont pas été anticipés. Ainsi dans le réseau 3, l'utilisation du chariot flottant devrait être évitée car son impact est négatif. Le temps de calcul est arbitrairement fixé à 20 minutes. Un temps de calcul plus élevé aurait pu nous permettre de trouver des solutions à des instances déclarées non résolues. Par exemple, le problème 3 avec trois chariots du réseau 3 serait classé résolu.

Tableau 3 – Réseau 2 (3 et 4 chariots)

	3 chariots				**4 chariots**			
Instance	**TPC**	**TPLNE**	**Durée**	**nbIter**	**TPC**	**TPLNE**	**Durée**	**nbIter**
1	0.78	5.58	99	0	0.20	81.72	99	0
2	0.08	3.67	75	0	0.05	13.54	75	0
3	0.44	4.02	85	0	23.26	12.33	84	0
4	0.05	4.27	85	0	0.69	13.66	83	0
5	0.03	3.47	69	0	0.03	5.67	67	0
6	0.03	3.2	67	0	0.01	7.19	67	0
7	0.13	3.23	60	0	0.27	6.46	60	0
8	0.11	3.63	67	0	0.03	8.44	75	0
9	0.11	5.52	99	0	0.08	15.11	99	0
10	0.08	11.79	99	0	0.03	11.19	75	1
11	0.08	3.68	83	0	0.05	9.63	83	0
12	0.05	4.35	83	0	0.06	6.78	83	0
13	0.05	3.90	67	0	0.19	9.28	69	6
14	0.05	3.66	67	0	0.03	4.85	67	0
15	0.02	2.93	59	0	0.03	8.86	59	0

Tableau 4 – Réseau 2 (5 chariots)

Instance	**TPC**	**TPLNE**	**Durée**	**nbIter**
1	0.03	32.72	99	0
2	0.11	44.20	75	0
3	0.03	47.24	83	0
4	18.01	81.70	83	0
5	0.03	13.44	67	0
6	0.02	22.60	67	0
7	0.02	18.55	59	0
8	0.03	13.15	67	0
9	0.08	11.79	99	1
10	0.03	14.46	75	1
11	0.06	62.14	83	1
12	0.05	18.73	83	1
13	0.15	10.58	69	5
14	0.03	9.10	67	0
15	0.02	38.13	59	0

Tableau 5 – Réseau 3 (3 et 4 chariots)

	3 chariots				**4 chariots**			
Instance	**TPC**	**TPLNE**	**Durée**	**nbIter**	**TPC**	**TPLNE**	**Durée**	**nbIter**
1	-	-	-	-	3.35	102.62	156	0
2	251.92	346.66	147	0	0.14	79.63	135	0
3	-	-	-	-	0.22	91.46	115	0
4	15.43	30.63	127	0	0.11	73.39	115	0
5	93.20	32.44	113	6	0.23	221.54	104	0
6	-	-	-	-	0.19	50.20	109	0
7	9.79	24.94	117	0	0.27	6.46	109	0
8	-	-	-	-	8.49	96.48	155	0
9	25.88	25.10	128	0	0.17	47.72	115	0
10	3.69	20.75	112	0	0.61	57.27	115	0
11	14	21.98	117	0	4.62	61.41	109	0

Tableau 6 – Réseau 3 (5 et 6 chariots)

	5 chariots				**6 chariots**			
Instance	**TPC**	**TPLNE**	**Durée**	**nbIter**	**TPC**	**TPLNE**	**Durée**	**nbIter**
1	6.95	530.20	156	1	0.42	- -	- -	- -
2	0.094	- -	- -	- -	0.11	500.61	135	1
3	0.44	108.10	115	1	0.11	- -	–	- -
4	0.08	83.74	115	1	0.08	176.48	115	1
5	-	-	-	-	7.73	- -	- -	- -
6	0.25	468.34	109	1	9.79	- -	- -	–
7	0.44	334.52	109	1	4.92	- -	- -	- -
8	0.97	220.17	155	1	1.41	- -	- -	- -
9	0.09	114.78	115	1	0.08	163.64	115	1
10	0.92	224.59	115	1	0.11	- -	- -	- -
11	2.86	- -	- -	- -	20.93	- -	- -	- -

Les résultats obtenus en termes de durée totale de toutes les activités sont en partie inattendus, ce qui explique leur discussion séparée. Dans le réseau 1, l'augmentation du nombre de chariots disponibles n'a pas d'impact sur la durée totale de toutes les tâches. Puisque tous les problèmes sont résolus avec trois chariots, il n'est pas nécessaire de passer à quatre chariots. Ces résultats s'expliquent par la taille du réseau, les temps de transition minimaux et la qualité de la borne inférieure du modèle de PC. Dans le réseau 2, le passage de trois à quatre chariots disponibles a un impact limité sur la durée totale de toutes les tâches à l'exception des problèmes 8 et 13 à quatre chariots où la durée totale de toutes les tâches augmente à cause de la congestion. L'ajout d'un chariot avec un certain point de départ et une orientation donnée peut causer du retard. Pour les problèmes 3, 4, 5 et 10 la durée totale des tâches baisse du fait de l'ajout d'un chariot proche d'un point de chargement. Lorsque le nombre de chariots disponibles passe à cinq, la durée totale des tâches augmente d'une période dans certains cas (problèmes 3 et 7). Pour le problème 8, le makespan augmente lorsqu'on passe de trois à quatre chariots puis avec cinq chariots, il y a un retour à une durée totale de tâches obtenue avec trois chariots. Ce qui est dû à la proximité du point de départ du chariot ajouté d'un point de chargement. Dans le réseau 3, les résolus obtenus sont conformes à nos attentes. À l'exception du problème 10, la durée totale des tâches baisse lorsque le nombre de chariots disponibles passe de trois à quatre. Lorsque le nombre de chariots disponibles est égal à six, il n'y a pas de gain de temps d'exécution des tâches à l'exception du problème 2.

En résumé, lorsque le nombre de chariots disponibles augmente de un, c'est la position de départ et l'orientation du chariot supplémentaire qui ont un impact sur la durée totale de toutes les tâches. Cet impact peut être positif (proximité d'un point de chargement) ou négatif (retard engendré dû à une orientation inappropriée). D'où l'importance d'un bon choix des points d'attente (qui sont en fait des points de départ potentiels des chariots).

L'anticipation sur les conflits dans le modèle de PC (grâce à l'implantation de la notion de gel de galeries) a donné des résultats très satisfaisants. Toutefois, elle n'est pas complète car les galeries intermédiaires n'ont pas été prises en compte. Les conflits résiduels peuvent survenir de deux façons :

- soit il y a un conflit dans une galerie parce qu'un chariot inoccupé bloque le passage à un autre chariot qui doit aller exécuter sa tâche au bout de la galerie ; ce chariot inoccupé n'a pas assez de temps pour sortir de la galerie et céder la place (ceci se passe dans la partie PLNE alors que l'ordonnancement est fait dans la partie PC) ;
- soit il y a un conflit dans une galerie intermédiaire c-à-d une galerie située entre deux intersections. Il s'agit par exemple de la galerie 10–11–12 du réseau 2. Un conflit peut survenir quand un chariot revient de la galerie de déchargement (voyage à vide pour aller effectuer un nouveau chargement) tandis qu'un autre chariot est déjà chargé et se dirige vers la galerie de déchargement ;

Pour éviter ces deux types de conflits, une solution tout à fait acceptable serait de créer dans chacune des galeries (en priorité les galeries intermédiaires) une ou deux petites voies latérales de dégagement (petites impasses) pour permettre à un chariot de céder la place à d'autres chariots.

Le modèle de PC est très efficace pour trois raisons principales : (1), l'analyse du problème permet de générer automatiquement des bornes inférieure et superieure très utiles pour l'aspect optimisation du modèle de PC. C'est la bonne qualité de la borne inférieure qui a plus d'impact sur le modèle de PC ; (2) la stratégie de recherche qui permet d'ordonner les activités sur les ressources unaires qu'elles doivent utiliser : (3) les contraintes globales dont les algorithmes de réduction de domaine sont très efficaces. Le modèle de PLNE a donné des résultats relativement bons vu sa grande taille, la difficulté d'éviter les conflits, de préciser l'orientation des chariots et le temps de solution est limité à vingt minutes.

6 Conclusion

Dans cet article, nous avons résolu un problème d'ordonnancement et routage intégrés d'une flotte de chariots dans une mine souterraine grâce une méthode de décomposition hybride qui combine la programmation par contraintes et la programmation linéaire en nombres entiers. Notre approche décompose le problème en deux parties : la première partie consiste en l'ordonnancement des tâches de chargement et déchargement de minerai tandis que la seconde partie résout le routage sans conflits des chariots avec prise en compte de l'orientation des chariots. Les problèmes considérés contiennent jusqu'à six chariots. Un effort substantiel est fait dans l'anticipation des conflits (modèle de PC) et la cohérence de l'orientation des chariots (modèle de PLNE). Le traitement de l'orientation des chariots dans la littérature est souvent négligé ou fait de façon inexacte. Notre approche traite de façon exacte l'orientation des chariots. En outre, une analyse de scénarios d'agrandissements est faite. Notre méthode pourrait servir au design d'une mine. Notre algorithme constitue un bon outil d'aide au dimensionnement de la flotte de chariots et peut être utilisé pour rapidement replanifer le travail en fonction de modifications aux galeries (bris de chariots, chutes de roches, fuites d'eaux souterraines). Il serait intéressant de tester notre approche dans d'autres contextes où il y aurait plus de tâches et de chariots, de développer des stratégies d'évitement de solutions symétriques dans le modèle de PC, de considérer le cas où deux points de déchargements coexistent. Une mine contient souvent deux points de déchargement : l'un servant à décharger le minerai utile, l'autre les rebuts (dans cette application, la maximisation du plan de production de minerai utile est visée). Ce qui permettrait d'éprouver un peu plus une version modifiée du modèle de PC. L'introduction de deux points de déchargements permettrait aussi de résoudre des problèmes dans d'autres contextes (terminaux à conteneurs par exemple).

References

Baptiste, P., Le Pape, C., Nuitjen, W. 2001. Constraint based Scheduling : Applying Constraint Programming to Scheduling Problems. *Operations Research/Management Science*. Kluwer Academic Publishers, Dordrecht.

Beaulieu, M., Gamache, C. 2004. An Enumeration Algorithm for Solving the Fleet Management Problem in Underground Mines. *Les Cahiers du GERAD* G–2004–52. Technical Report. HEC Montréal.

Bigras, L-P., Gamache, M. 2002. A solution approach for real-time fleet management system : an application to underground mining. *Les Cahiers du GERAD* G–2002–66. Technical Report. HEC Montréal.

Bigras, L-P., Gamache, M. 2005. Considering displacement modes in the fleet management problem. *International Journal of Production Research* **43** 1171–1184.

Co, C.G., Tanchoco, J.M.A. 1991. A Review of Research on AGVs Vehicle Management. *Engineering Costs and Production Economics* **21** 35–42.

Corréa, A.I., Langevin, A., Rousseau, L-M. 2004. Dispatching and Conflict-Free Routing of Automated Guided Vehicles : A Hybrid Approach Combining Constraint Programming and Mixed Integer Programming. Régin, J-C. and Rueher, M. Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. First International Conference, CPAIOR. Lecture Notes in Computer Science. LNCS 3011, 2004. Springer.

Corréa, A.I., Langevin, A., Rousseau, L-M. 2005. Scheduling and Routing of Automated Guided Vehicles : a Hybrid Approach. *Computers and Operations Research* To appear.

Desaulniers, G., Langevin, A., Riopel, D., Villeneuve, B. 2003. Dispatching and Conflict-Free Routing of Automated Guided Vehicles : An Exact Approach. *The International Journal of Flexible Manufacturing Systems* **15** 309–331.

Dincbas, M., Van Hentenryck, P., Simonis, H. 1990. Solving Large Combinatorial Problems in Logic Programming. *Journal of Logic Programming* **8** 75–93.

Fuji, S., Sandok, H., Hohzaki, R. 1988. Routing control of automated guided vehicles in FMS. *Proceedings of the USA-Japan Symposium on Flexible Automation*, Minneapolis, MN. 130–136.

Gamache, M., Grimard, R., Cohen, P. 2004. A shortest-path algorithm for solving the fleet management problem in underground mines. *European Journal of Operational Research* **166** 497–506.

Ganesharajah, T., Hall, N.G., Sriskandarajah, C. 1998. Design and Operational Issues in AGV-Served Manufacturing Systems. *Annals of Operations Research* **76** 109–154.

Hooker, J. N. 2000. *Logic-Based Methods for Optimisation*. Wiley, New York.

Hooker, J.N., Ottosson, G. 2003. Logic-based Benders Decomposition. *Mathematical Programming* **96** 33–61.

Ilog OPL Studio 2004. *Ilog OPL Studio 3.7* Language Manual, 2004.

Jaffar, J. and Maher, M. J. 1994. Constraint Logic Programming : A Survey.*Journal of Logic Programming* **19/20** 503–581.

King, R.E., Wilson, C. 1991. A review of Automated Guided Vehicle System Design and Scheduling. *Production Planning and Control* **2** 44–51.

Krishnamurthy, N.N., Batta, R., Karwan, K. 1993. Developing conflict-free routes for automated guided vehicles. *Operations Research* **4** 1077–1090.

Langevin, A., Lauzon, D., 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.

Mariott, K., Stuckey, P.J. 1998. *Programming with Constraints*. The MIT Press.

Milano, M. 2004. Constraint and Integer Programming : Toward a Unified Methodology. Ramesh Sharda and Stefan Vob, eds. *Operations Research/Computer Science Interfaces Series*. Kluwer Academic Publishers. 33–53.

Qiu, L., Hsu, W.-J., Wang, H. 2002. Scheduling and Routing Algorithms for AGVs : A Survey. *International Journal of Production Research* **40** 745–760.

Vagenas, N. 1991. Dispatch control of a fleet of remote-controlled/automatic load-haul-dump vehicles in underground mines. *International Journal of Production Research* **29** 2347–2363.

Van Hentenryck, P. 1989. *Constraint Satisfaction in Logic Programming*. MIT Press, Cambridge, Mass.

Wallace, M. 1996. Practical Applications of Constraint Programming. *Constraints* **1** 139–168.