

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Optimisation de la planification à court terme avec contraintes de trafic dans
les mines souterraines**

LUCAS BERGEVIN

Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques appliquées

Août 2021

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Optimisation de la planification à court terme avec contraintes de trafic dans
les mines souterraines**

présenté par **Lucas BERGEVIN**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Jean-Marc FRAYRET, président

Michel GAMACHE, membre et directeur de recherche

Amina LAMGHARI, membre

DÉDICACE

À ma famille et mes amis, merci...

REMERCIEMENTS

Je souhaite remercier mon professeur, Michel Gamache, pour son soutien continu envers la réalisation de mon projet. La guidance de Michel à mon égard a sans l'ombre d'un doute contribué grandement à la réussite de mon travail. J'ajouterais par-dessus tout que Michel est très clair et structuré dans ses propos. Sans même savoir quel projet je voulais réaliser, je savais que je devais choisir Michel, mon ancien professeur de recherche opérationnelle.

Louis-Pierre Campeau, que ce soit le chercheur ou chef d'équipe à Newtrax, dans tous ces rôles ta contribution envers mes réalisations a été indispensable à leurs succès. Ta vision de projet, ta motivation et ta patience envers tes apprentis sont toutes des raisons pour lesquelles tu as été un bon leader pour moi. Sans oublier, ton aide pour la partie de programmation par contraintes a été plus que bénéfique.

À tous les membres du département mine chez Agnico Eagle, votre solidarité dans ce coin isolé du monde m'a permis de me sentir comme chez moi. L'environnement minier est un deuxième monde pour moi et se rendre sous-terre pour bien comprendre le processus minier était une expérience inoubliable.

RÉSUMÉ

Ce mémoire s'insère dans l'ensemble des oeuvres qui ont pour cible l'optimisation de la planification dans l'univers complexe des mines souterraines. L'approche empruntée dans ce mémoire est de considérer le trafic des véhicules qui est une conséquence aux décisions prises en amont lors de la planification. L'objectif principal est de déterminer des contraintes de trafic à inclure à la planification à court terme afin d'améliorer la productivité des mines souterraines. Afin d'y arriver, trois sous-objectifs ont été érigés. D'abord, il a été nécessaire de modéliser le trafic généré par les activités de production. Ensuite, des contraintes fidélisant la congestion des routes ont été déterminées. Enfin, une validation de l'ajout de ces contraintes à la planification est faite. Plusieurs modèles de programmation par contraintes ont ainsi été créés pour le volet planification et un modèle de simulation a servi à valider l'efficacité de chaque modèle de planification.

Au travers de la revue de la littérature à ce sujet, nous constatons que l'inclusion du trafic dans un outil de planification d'une mine souterraine n'a jamais été tentée auparavant. D'ailleurs, le domaine de l'optimisation pour ce type de mine a très peu été couvert auparavant. La majorité des modèles utilisent de la programmation linéaire mixte afin de résoudre le problème de planification à court, moyen et long terme. Récemment, des modèles de programmation par contraintes appliqués au problème d'optimisation de la planification ont soulevé beaucoup d'intérêt. Ces modèles sont beaucoup plus rapides et plus simples à résoudre que ceux en programmation linéaire. Également, plusieurs cas d'application de la simulation au domaine des mines souterraines sont présents dans la littérature. Ce type d'outil a fait ses preuves quant à son efficacité à reproduire un environnement complexe comme une mine souterraine sous format numérique.

Dans un premier temps, le modèle de simulation d'une mine souterraine est présenté. Les données de sortie de ce modèle ont été utilisées afin de modéliser le trafic dans la mine souterraine. Le modèle a surtout permis d'utiliser les planifications provenant de modèles de planification afin de réaliser celles-ci dans l'environnement minier numérique. La simulation de ces planifications permet surtout d'éviter toutes perturbations qui pourraient être provoquées par de tels tests sur la mine réelle. Grâce à cet outil, il a été possible de récolter plusieurs résultats en lien avec la réalisation des planifications de 10 scénarios pour chaque modèle de planification afin de valider l'ajout des contraintes de trafic au modèle de programmation par contraintes.

Ensuite, quatre modèles de planification sont présentés. Le modèle CP-M0 sans contraintes de trafic, le modèle CP-M1 avec une somme de trafic en fonction objectif, le modèle CP-M2 avec des contraintes de délais de trafic pour les activités en développement et le modèle CP-M3 avec des contraintes de délais de trafic pour toutes les activités. En simulant les planifications de ces modèles pour tous les scénarios, nous démontrons que le modèle CP-M3 avec des contraintes de trafic permet d'obtenir des planifications plus réalistes. De plus, ce modèle obtient un temps de complétion moyen plus faible que ceux obtenus par le modèle de planification sans contraintes de trafic (CP-M0) sur l'ensemble des scénarios.

ABSTRACT

This master's thesis inserts itself in the global effort being put forth concerning the optimization of the underground mines efficiency. The approach taken in this master thesis is to consider the underground traffic caused by the short-term planning of the operations. The uttermost objective of this thesis is to determine traffic constraints to include in short-term planning tools to improve the productivity of underground mines. In order to reach this goal, three sub-objectives are presented. Firstly, we model the traffic generated by the mining activities. Secondly, traffic constraints were created to represent the congestion of an underground mine. Lastly, we validate the traffic constraints used in short-term planning tools. Overall, four different constraint programming models are presented as short-term planning tools as well as a simulation model used for validating the efficiency of these planning tools.

Throughout the literature on the subject of optimization of short-term planning in underground mines, we note that including traffic in such tools was never done before. Moreover, the number of articles covering optimization for this kind of mine is very little compared to open-pit mines. The majority of models for underground mines use mixed linear programming in order to solve the short-, medium- and long-term planning problem. Recently, a small amount of constraint programming models applied to the optimization problem was shown to improve drastically the resolution of such problems when compared to linear programming. Also, many cases of application using simulation in underground mines are presented in the literature. This kind of tool as proven itself when it comes to reproducing a complex environment such as an underground mine in a digital format.

At first, the simulation model of an underground mine is presented. Output data from this model is used to shape the traffic in the mine. We were able to use the short-term plans from the planning tools as input data for the simulation model and then analyze the performance of such plans in a digital underground mine. Simulating these plans avoids disturbing the real underground mines by testing the generated plans in a digital version of the mine. With this tool, many results linked to the simulation of planning of 10 scenarios were obtained for every planning model. Such results allowed us to validate the addition of traffic constraints to the constraint programming models.

Next, four short-term planning models are presented. The CP-M0 model without any traffic constraints, the CP-M1 model with a traffic sum inserted in the objective function, the CP-M2 model with traffic delays constraints for development activities and then, the CP-M3

model with traffic delays constraints for every activity. By simulating the plans from these models for all the scenarios, we demonstrate that the CP-M3 model with traffic constraints produces more realistic plans. Furthermore, this model obtains the lowest makespan average over all the scenarios when compared to the model with no traffic constraints (CP-M0).

TABLE DES MATIÈRES

DÉDICACE	iii
REMERCIEMENTS	iv
RÉSUMÉ	v
ABSTRACT	vii
TABLE DES MATIÈRES	ix
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
LISTE DES SIGLES ET ABRÉVIATIONS	xv
LISTE DES ANNEXES	xvi
CHAPITRE 1 INTRODUCTION	1
1.1 Définitions et concepts de base	1
1.2 Éléments de la problématique	3
1.3 Objectifs de recherche	6
1.4 Plan du mémoire	6
CHAPITRE 2 REVUE DE LITTÉRATURE	7
2.1 Optimisation de la planification dans les mines souterraines	7
2.1.1 Long terme	7
2.1.2 Moyen terme	8
2.1.3 Court terme	9
2.1.4 Temps réel	11
2.2 Ordonnancement de tâches	12
2.2.1 Gestion de projets avec contraintes de ressources limitées	13
2.2.2 Programmation par contraintes	13
2.3 Simulation dans les mines	14
2.4 Méthodes choisies	15

CHAPITRE 3	MÉTHODOLOGIE	17
3.1	Le modèle de simulation	17
3.1.1	Analyse du processus de transport de minerai	17
3.1.2	Les données d'entrée	19
3.1.3	La construction du simulateur	22
3.1.4	La validation	31
3.1.5	Les données de sortie	31
3.2	L'outil de planification à court terme	32
3.2.1	Les données d'entrée	33
3.2.2	Le modèle de programmation par contraintes	34
3.2.3	Les données de sortie	43
3.2.4	L'ajout des contraintes de trafic	44
CHAPITRE 4	RÉSULTATS EXPÉRIMENTAUX ET DISCUSSION GÉNÉRALE	50
4.1	Planifié vs simulé	51
4.1.1	Écarts entre la planification et la simulation pour le modèle CP-M0	51
4.1.2	Écarts entre la planification et la simulation pour le modèle CP-M1	53
4.1.3	Écarts entre la planification et la simulation pour le modèle CP-M2	55
4.1.4	Écarts entre la planification et la simulation pour le modèle CP-M3	56
4.1.5	Comparaison des écarts entre la planification et la simulation de tous les modèles	57
4.2	Optimalité des outils sans trafic vs avec trafic	60
4.2.1	CP-M0 vs CP-M1	60
4.2.2	CP-M0 vs CP-M2	62
4.2.3	CP-M0 vs CP-M3	64
4.2.4	Comparaison des temps de complétion simulés de tous les modèles	66
CHAPITRE 5	CONCLUSION ET RECOMMANDATIONS	68
RÉFÉRENCES		70
ANNEXES		75

LISTE DES TABLEAUX

Tableau 3.1	Les paramètres attribués aux types de véhicules de la flotte.	20
Tableau 3.2	Les paramètres d'environnement provenant du fichier Excel principal.	21
Tableau 3.3	Les principaux attributs mis à jour lors de l'indexation des tâches.	25
Tableau 3.4	Descriptions des ensembles.	35
Tableau 3.5	Descriptions des paramètres.	37
Tableau 3.6	Descriptions des variables.	38
Tableau 4.1	Le nombre de variables et contraintes pour chaque scénario en fonction du modèle.	52

LISTE DES FIGURES

Figure 1.1	Visualisation de la méthode d'exploitation par sous-niveaux abattus [1].	2
Figure 3.1	Le processus d'acquisition de données de trafic pour la mine étudiée.	18
Figure 3.2	Exemple des paramètres de séquençement des tâches d'un véhicule.	22
Figure 3.3	Exemple du chemin possible entre A et B sur un niveau à un <i>footwall</i>	26
Figure 3.4	Exemple de deux chemins possibles entre A et B sur un niveau à deux <i>footwalls</i>	27
Figure 3.5	Exemple de deux chemins considérés entre A et B par l'heuristique sur un niveau à trois <i>footwalls</i>	28
Figure 3.6	Exemple d'un niveau faisant partie du tableau de bord. Les encadrés avec des zéros sont les valeurs des variables de noeuds. . . .	29
Figure 3.7	Exemple d'un graphique d'utilisation des équipements au travers du temps où chaque couleur représente un équipement.	44
Figure 4.1	Comparaison des temps de complétion des planifications du CP-M0 (10 minutes de résolution) vs les temps de complétion moyens simulés de ces mêmes planifications.	53
Figure 4.2	Comparaison des temps de complétion des planifications du CP-M0 (30 minutes de résolution) vs les temps de complétion moyens simulés de ces mêmes planifications.	54
Figure 4.3	Comparaison des écarts entre les temps de complétion simulés avec les temps de complétion des planifications pour tous les modèles CP-M1.	55
Figure 4.4	Comparaison des temps de complétion des planifications du CP-M2 vs les temps de complétion moyens simulés de ces mêmes planifications.	56
Figure 4.5	Comparaison des temps de complétion des planifications du CP-M3 vs les temps de complétion moyens simulés de ces mêmes planifications.	57

Figure 4.6	Comparaison des écarts moyens (moyenne sur les 10 scénarios) entre les temps de complétion simulés et les temps de complétion planifiés pour tous les modèles.	58
Figure 4.7	Comparaison des écarts moyens absolus (moyenne sur les 10 scénarios) entre les temps de complétion simulés et les temps de complétion planifiés pour tous les modèles.	59
Figure 4.8	Comparaison des temps de complétion moyens simulés provenant du CP-M0 (10 minutes de résolution) avec ceux des modèles CP-M1.	61
Figure 4.9	Comparaison des durées de trafic moyens simulés provenant du CP-M0 (10 minutes de résolution) avec ceux des modèles CP-M1. . .	62
Figure 4.10	Comparaison des temps de complétion moyens simulés provenant du CP-M0 (10 minutes de résolution) avec ceux du CP-M2. . . .	63
Figure 4.11	Comparaison des durées de trafic simulés provenant du CP-M0 (10 minutes de résolution) avec ceux du CP-M2.	64
Figure 4.12	Comparaison des temps de complétion moyens simulés provenant du CP-M0 (30 minutes de résolution) avec ceux du CP-M3. . . .	65
Figure 4.13	Comparaison des durées de trafic simulés provenant du CP-M0 (30 minutes de résolution) avec ceux du CP-M3.	66
Figure 4.14	Comparaison des écarts moyens entre les temps de complétion simulés des modèles de trafic (CP-M1, CP-M2 et CP-M3) avec ceux des modèles sans trafic.	67
Figure A.1	Section 1.0 : Création des entités de la flotte.	75
Figure A.2	Section 1.1 : Comptabilisation du nombre de tâches par entité.	76
Figure A.3	Section 1.2 et 1.3 : Création des entités quart de travail et des entités d'enregistrement des résultats en format CSV.	77
Figure A.4	Section 2.0 : Attribution des tâches pour les chargeuses.	78
Figure A.5	Section 2.1 : Attribution des tâches pour les boulonneuses. . .	79
Figure A.6	Section 2.2 : Attribution des tâches pour les <i>jumbo</i>	80
Figure A.7	Section 2.3 : Attribution des tâches pour les camions à émulsion.	81
Figure A.8	Section 2.4 : Attribution des tâches pour les camions.	82
Figure A.9	Section 3.0 : Aiguillage des entités, qui ont une nouvelle tâche, vers la section de déplacement appropriée.	83
Figure A.10	Section 3.1 : Déplacements d'entrée de niveau ou de rampe. . .	84
Figure A.11	Section 3.2 : Déplacements dans la rampe 2.	85
Figure A.12	Section 3.3 : Déplacements dans la rampe 1.	86
Figure A.13	Section 3.4 Partie 1 : Aiguillage des déplacements dans les niveaux.	87

Figure A.14	Section 3.4 Partie 2 : Déplacements verticaux dans les niveaux.	88
Figure A.15	Section 3.4 Partie 3 : Déplacements horizontaux dans les niveaux.	89
Figure A.16	Section 3.4 Partie 4 : Laissez-passer associé au trafic dans les niveaux.	90
Figure A.17	Section 4.0 Partie 1 : Aiguillage des entités vers les logiques d'opérations appropriées.	91
Figure A.18	Section 4.0 Partie 2 : Toutes les opérations de développement et de remblayage sauf le soutirage.	92
Figure A.19	Section 4.0 Partie 3 : Opérations de soutirage pour le dévelop- pement et la production.	92
Figure A.20	Section 4.1 : Opérations de déchargement de la roche à la surface.	93
Figure A.21	Section 5.0 : Files d'attentes lors des changements de quart. .	94
Figure A.22	Tableau de bord pour les simulations.	95
Figure B.1	Comparaison des temps de complétion moyens simulés prove- nant de tous les modèles.	96

LISTE DES SIGLES ET ABRÉVIATIONS

VAN	Valeur Actuelle Nette
MIP	Mixed Integer Programming
CP	Constraint Programming
LRMS	Low Resolution model with Micro Selectivity
RMSOP	Resource based Mine Scheduling Optimization Problem
LHD	Load Haul Dump
PCC	Plus Courts Chemins
HEFT	Heterogeneous Earliest-Finish-Time
CPOP	Critical-Path-on-a-Processor
CPGA	Critical Path Genetic Algorithm
RCPSP	Resource-Constrained Project Scheduling Problem
DP	Drawpoint
FW	Footwall
API	Application Programming Interface
CPU	Central Processing Unit
RAM	Random-Access Memory

LISTE DES ANNEXES

Annexe A	Modèle de simulation Arena	75
Annexe B	Modèles de planification	96

CHAPITRE 1 INTRODUCTION

Les mines souterraines ont un rôle majeur dans l’approvisionnement des matières premières pour les entreprises autour du globe. Lorsqu’un gisement d’intérêt est trop profond pour l’excavation à ciel ouvert, il faut se tourner vers une méthode d’exploitation souterraine pour atteindre la matière première. Une fois que la méthode d’extraction est déterminée, un plan de durée de vie de la mine (*life of mine*) est mis en place pour planifier chaque étape à franchir de l’ouverture jusqu’à la fermeture de la mine. En se référant aux objectifs du plan, une planification à long terme (*long-term planning*) est créée et mise à jour annuellement afin d’identifier les cibles de développement et de production à atteindre pour les prochaines années. Par la suite, une planification à moyen terme (*medium-term planning*) est utilisée afin d’identifier plus précisément les cibles de production et de développement qui donnent accès aux points de soutirage (*drawpoints*) et les chantiers de production (*stopes*) respectivement sur un horizon de quelques mois. Après vient le niveau de planification à court terme (*short-term planning*) qui donne les cibles sur un horizon d’une à deux semaines. C’est avec cette planification que le département des opérations organise les activités quotidiennes.

1.1 Définitions et concepts de base

Afin de bien comprendre ce mémoire, certaines notions doivent être établies quant aux concepts miniers utilisés. L’exploitation du gisement de la mine est faite à l’aide de la méthode d’exploitation par sous-niveaux abattus (*sublevel stoping*). Plusieurs niveaux de profondeurs différentes sont développés afin de se rendre à diverses parties du gisement. Comme il est possible d’observer sur la figure 1.1, des points de soutirage (*drawpoints*) sont développés à chaque niveau afin d’accéder au minerai. Après, l’idée est de forer de longs trous (*long-hole drilling*) à partir d’un point de soutirage vers un autre point de soutirage du niveau subjacent. Ces trous, qui passent directement dans le gisement, sont ensuite remplis d’explosifs et explosés afin de fragmenter le minerai et le faire tomber au point de soutirage du niveau subjacent. Ce point de soutirage devient alors un chantier de production (*stope*) et permettra ainsi d’extraire un grand volume de minerai.

Les mines souterraines sont composées de rampes d’accès qui commencent à partir de la surface et permettent de se rendre à toutes les galeries des niveaux. La méthode de forage et dynamitage (*drill and blast*) est employée en cycle afin de développer ces rampes et galeries. Ce cycle commence par le chargement de la roche (*mucking*) dans des camions par une chargeuse. Lorsqu’un camion est rempli, il remonte à la surface pour décharger son contenu dans

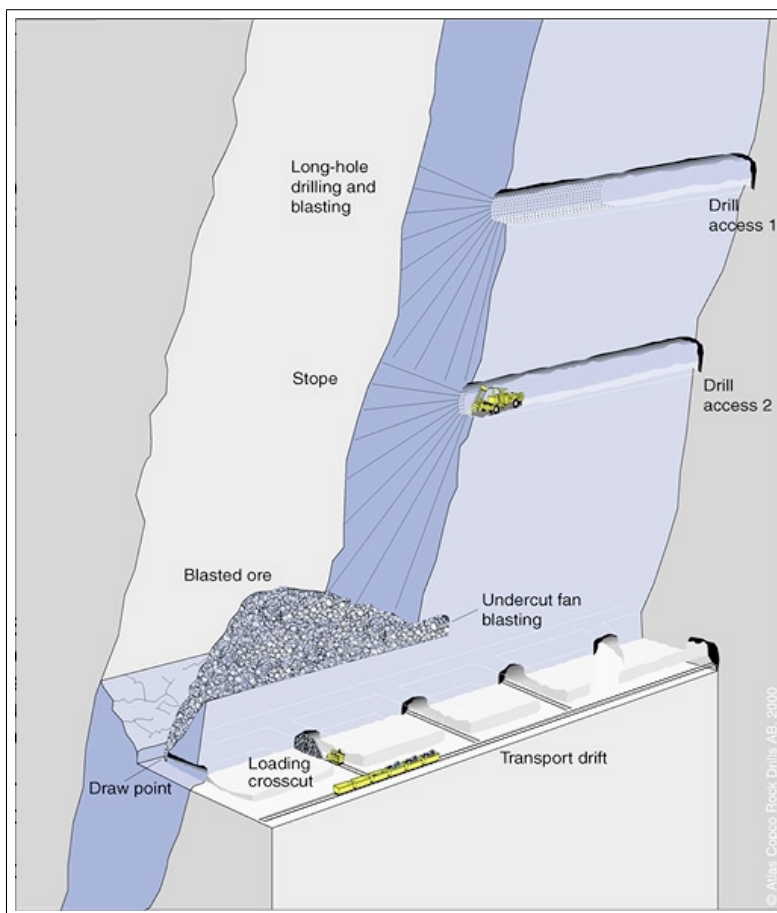


Figure 1.1 Visualisation de la méthode d'exploitation par sous-niveaux abattus [1].

la pile de roche appropriée. Ces camions font des allers-retours entre le lieu de chargement et de déchargement jusqu'à ce que toute la roche dynamitée soit évacuée à la surface. Ensuite, une boulonneuse vient à cet emplacement pour installer des tiges de forages avec du grillage pour assurer le soutien du tunnel et empêcher les chutes de pierres. Plusieurs blocs de roche instables tombent sur le plancher lors de ce boulonnage (*bolting*). La prochaine étape est alors de nettoyer le plancher (*clean face*) à l'aide d'une chargeuse pour libérer la face de travail. Une fois terminé, un jumbo de forage peut se rendre à l'emplacement pour forer les trous utilisés pour le dynamitage. D'ailleurs, c'est après cette étape de forage (*drilling*) qu'un camion à émulsion vient remplir les trous d'un mélange explosif. Une fois que l'emplacement est sécurisé, l'explosion (*blast*) a lieu entre les quarts de travail et le cycle recommence à nouveau avec l'étape de chargement de la roche dynamitée.

Chaque mine possède un plan de durée de vie de la mine qui détaille toutes les étapes nécessaires de l'ouverture de la mine jusqu'à sa fermeture. La phase d'exploitation du plan

est une composante clé qui établit les objectifs d'extraction à atteindre au courant des années. Une planification à long terme qui s'étale sur plusieurs années cherche à atteindre ces objectifs. Étant donné la longue portée de ce plan, les tâches journalières sont accumulées afin de réduire le nombre de variables en jeu et ainsi avoir une vision plus globale de la production. Des périodes annuelles sont souvent utilisées pour ce type de planification. L'objectif derrière une planification à long terme est de maximiser la profitabilité de la mine. Un indicateur comme la valeur actuelle nette (VAN) est souvent utilisé à cet effet pour assurer la rentabilité du projet minier. De plus, ce plan est souvent révisé à chaque année afin d'y inclure la progression de la mine et certaines contraintes autrefois inconnues du planificateur.

La première année du plan à long terme est divisée en plusieurs périodes d'un à trois mois qui sont utilisées pour la planification à moyen terme. Un avantage de cette planification est qu'elle permet de déterminer plus précisément les séquences d'activités à exécuter sur certaines parties du gisement. De plus, l'objectif de cette planification est d'obtenir ces séquences tout en atteignant les cibles de productions imposées par la planification à long terme pour la première année.

Une planification plus précise peut être obtenue en décomposant la première période de la planification à moyen terme en périodes d'une à deux semaines. Pour chacune de ces périodes, les équipes de travail sont affectées aux différentes tâches associées aux parties du gisement identifiées par le plan à moyen terme. L'objectif est de minimiser le temps de complétion total (*makespan*) de ces tâches. Plusieurs contraintes qui étaient négligées dans les planifications à moyen et long terme sont prises en compte ici. D'ailleurs, il y a les contraintes de précédence entre les tâches des cycles de forage et dynamitage qui obligent le début d'une tâche à être après la fin de la tâche prédécesseur. De plus, un équipement mobile peut seulement effectuer une seule tâche à la fois et un emplacement de travail peut accueillir qu'une seule tâche à la fois. Ainsi, des contraintes de non-chevauchement sont utilisées à cet effet. Des contraintes de changement de quart sont utilisées pour obliger les dynamitages à se produire lors des changements de quarts de travail. Ensuite, certaines tâches nécessitent plusieurs types d'équipements. Des contraintes de synchronisation forcent alors ces équipements à être présents au même moment. La planification à court terme permet ainsi de connaître l'ordre optimal dans lequel les tâches journalières doivent être au courant de la semaine.

1.2 Éléments de la problématique

Le transport de la roche représente à lui seul environ 50% des coûts d'opération pour une mine à ciel ouvert [2]. La question d'optimisation de la planification des ressources a largement

été couverte quant à ce type de mine. À l'inverse, pour les mines souterraines, le manque de solutions à ce problème d'optimisation peut être expliqué par plusieurs facteurs, dont le manque en technologies de suivi de l'équipement [3]. Ce manque ne permet pas de faire un suivi en continu des opérations minières et a défavorisé l'élaboration d'outils de planification. Encore aujourd'hui, la planification à court terme est résolue à la main, avec des contraintes de temps limitées, pour plusieurs mines. Cette résolution va grandement dépendre de l'expérience de l'employé en poste, ce qui met en doute l'efficacité de cette façon d'opérer qui n'est pas standardisée. Sans oublier qu'il est très peu probable que la solution trouvée par l'employé soit la plus optimale étant donné l'erreur humaine et le manque de standard dans la planification des opérations.

De récentes avancées ont été effectuées en planification à court terme et moyen terme pour les mines souterraines. D'ailleurs, Campeau et Gamache ont démontré dans [4] qu'il est possible d'obtenir une plus grande rapidité d'exécution (26 contre 3600 secondes) et un écart plus faible avec l'optimalité (0.1 contre 1240 %) en utilisant la programmation par contraintes au lieu de la programmation linéaire mixte (MIP) pour planifier les activités de production sur une année [5]. Pour ce mémoire, l'accent est mis sur la planification à court terme sur un horizon d'une semaine de production puisqu'il s'agit de l'étape qui permet de faire le pont entre la planification à moyen terme et les opérations journalières des mineurs. Ainsi, en améliorant la productivité des opérations hebdomadaires, la durée de vie de la mine est réduite et permet alors d'augmenter la rentabilité de l'entreprise minière.

Lorsqu'il vient temps d'apporter des changements aux opérations, la simulation devient un outil indispensable afin d'évaluer l'impact de telles modifications sur l'organisation du travail. Plus précisément, un modèle de simulation est une réplique la plus fidèle possible du système à l'étude. Le but d'un tel outil est d'évaluer le comportement du système numérique sous plusieurs conditions différentes avant d'apporter des changements au système réel qui pourraient le perturber. Dans ce mémoire, un modèle de simulation à événements discrets de la mine est établi. L'avantage de ce type de modèle est qu'il permet de reproduire en partie la variabilité et le dynamisme en lien avec les opérations et les déplacements des véhicules. Aussi, cet outil permet de tester les affectations des tâches à la flotte de véhicules dans un environnement numérique sans risque de nuire aux opérations en cours dans le monde réel.

En ce qui concerne la planification de la production, les mines se tournent de plus en plus vers des outils d'optimisation afin de résoudre des problèmes qui étaient autrefois résolus à la main. Le modèle d'optimisation de ce mémoire est un programme de programmation par contraintes (CP). Ce type de modèle permet de résoudre des problèmes combinatoires de grande taille.

Ceci s'avère alors idéal pour le problème d'affectation de tâches pour la flotte minière. Le modèle se compose de variables de décisions et de contraintes qui établissent des liens entre les variables. En fonction de ces liens, le domaine des valeurs possibles pour chaque variable présente dans une contrainte est limité. Le but d'un programme CP est l'affectation de valeurs à toutes les variables et satisfaire toutes les contraintes. En ayant les domaines de variables, il serait possible de simplement énumérer toutes les combinaisons possibles afin de trouver une solution optimale. Cependant, pour des problèmes de grande taille, comme l'affectation de tâches sur une semaine pour l'ensemble de la production minière, une simple énumération des solutions possibles devient impraticable. Pour arriver à trouver une solution optimale au problème d'affectation, une recherche arborescente est faite afin d'explorer chaque solution possible à partir de différentes affectations de valeurs de variables. De plus, une propagation des contraintes est faite afin de réduire le nombre de solutions à explorer.

Néanmoins, aucun outil de planification n'incorpore de notions associées au trafic souterrain qui est démontré d'avoir un impact non négligeable sur la productivité [6]. Effectivement, une mine souterraine est un circuit fermé avec un nombre fixe de galeries qui communiquent entre les niveaux. La largeur des galeries permet seulement d'avoir une voie pour le passage des véhicules miniers pour des raisons de coûts de développement élevés et de support de la roche affaiblie. Par conséquent, lorsque deux véhicules dans une même galerie cherchent à aller dans des directions opposées, l'un doit attendre le passage de l'autre. Des priorités de passage sont alors érigées afin de choisir quel conducteur aura le passage avant un autre. Cependant, cela ne règle pas le problème des pertes de temps associés aux dépassements. Cette congestion des galeries s'intensifie au niveau des rampes puisqu'il s'agit que d'une seule voie de déplacement entre les niveaux et la surface. Les camions vides à la surface cherchent à descendre la rampe tandis que les camions pleins cherchent à remonter à la surface pour vider leur contenu en roche. Évidemment, des baies de dépassements sont présentes entre les niveaux et des segments de la rampe, mais elles peuvent accueillir qu'un seul véhicule à la fois.

De plus, les opérateurs de véhicules miniers ont pour tâche de communiquer constamment à la radio afin d'avertir les autres de leur position dans la rampe. Pourtant, malgré cette communication, le trafic persiste dans la mine étant donné la configuration des rampes et le flot de véhicules se dirigeant dans des directions opposées. D'ailleurs, la courbure des rampes et la noirceur présente dans les mines souterraines font en sorte qu'il est fréquent que des véhicules fassent des face-à-face avec d'autres véhicules du réseau de transport. Les camions doivent alors faire marche arrière pour laisser passer un véhicule qui a la priorité de passage. Ainsi, que ce soit l'attente dans une baie ou de faire marche arrière, ces temps morts très fréquents dans la journée diminuent grandement la productivité des opérations. Sans

oublier les impacts en termes de coût d'essence, de risque d'accident lorsqu'un camion est à reculons et de patience de l'opérateur du véhicule qui doit reculer dans une rampe inclinée. L'inclusion de cette composante de trafic souterrain à la planification permettrait ainsi un gain économique significatif.

Les modèles de planification à court terme ignorent les temps perdus en raison de la gestion du trafic par les opérateurs. Cet aspect est ignoré, car il est difficile d'évaluer le temps perdu en raison du trafic et de le prévoir étant donné que celui-ci dépend fortement des chantiers simultanément en opération dans la mine. Ces modèles ont alors tendance à proposer des solutions optimistes où les activités sont réalisées dans une période de temps plus courte que celle observée en réalité. Le but de ce mémoire est de proposer un modèle de planification qui tiendra compte de cet aspect.

1.3 Objectifs de recherche

L'objectif du travail présenté est de proposer un modèle mathématique pour la planification à court terme des opérations de minage en y incluant la congestion souterraine afin d'optimiser la productivité des mines souterraines. Pour valider le modèle de planification proposé, un modèle de simulation sera créé. Le simulateur permettra d'estimer la pertinence d'inclure de nouvelles contraintes de trafic dans les outils de planification.

On propose une méthodologie de résolution du problème en trois étapes :

1. Modéliser le trafic généré par les activités de production.
2. Déterminer des contraintes fidélisant la congestion des routes.
3. Valider l'ajout des contraintes au modèle de planification.

1.4 Plan du mémoire

Les chapitres suivants du mémoire dévoilent les démarches utilisées afin d'obtenir les résultats qui répondent aux objectifs érigés précédemment. Au chapitre 2, une revue de la littérature est établie quant à l'état de l'art de l'optimisation dans le domaine des mines souterraines. Au chapitre 3, nous présenterons la méthodologie employée pour l'outil de simulation et l'outil de planification. Au chapitre 4, les tests effectués seront décrits et les résultats expérimentaux seront présentés et analysés. Finalement, le chapitre 5 propose une conclusion sur les résultats obtenus ainsi que des avenues de recherche afin d'améliorer le modèle proposé.

CHAPITRE 2 REVUE DE LITTÉRATURE

Un survol de la littérature sera présenté ici quant au sujet de l’optimisation de la planification dans les mines souterraines afin de bien situer l’avancée de la recherche dans ce domaine. De plus, une revue de l’ordonnancement de tâches et de la simulation dans les mines sera faite afin d’informer le lecteur quant aux techniques utilisées pour atteindre les objectifs de ce mémoire.

2.1 Optimisation de la planification dans les mines souterraines

Plusieurs niveaux de planification sont nécessaires afin d’assurer que les opérations journalières contribuent à l’avancement des objectifs à long terme imposés par le plan de durée de vie de la mine souterraine. D’ailleurs, trois niveaux de planification sont surtout observés dans la littérature : la planification à long terme, la planification à moyen terme et la planification à court terme. Les horizons de temps pour chacune de ces planifications n’étant pas clairement définis dans la littérature, les sous-sections qui suivent se réfèrent alors aux horizons mentionnés dans [4]. Il est à mentionner que plusieurs travaux ont été réalisés pour les mines à ciel ouvert, mais compte tenu des différences opérationnelles importantes entre les mines à ciel ouvert et souterraines on se concentre ici que sur la littérature portant sur la planification dans les mines souterraines.

2.1.1 Long terme

L’horizon de ce type de planification est au minimum une année et peut s’étaler sur plusieurs dizaines d’années. La maximisation de la rentabilité du projet minier sur des périodes annuelles est utilisée dans la plupart des cas présents dans la littérature. [7] présente un modèle de planification à long terme d’une mine de cuivre souterraine sur un horizon de 25 ans. Cet horizon est composé d’une période d’un an, d’une période de quatre ans et d’une période de 20 ans. Ce modèle de programmation linéaire mixte (MIP) cherche à maximiser la valeur actuelle nette. Étant donné la taille du problème d’optimisation, les contraintes d’intégrité sur certaines variables ont été relaxées et une heuristique pour arrondir les résultats est créée pour retrouver des solutions entières. Un autre modèle MIP est présenté dans [8] pour résoudre le problème de planification à long terme d’une mine de fer souterraine sur un horizon de 3 ans. Contrairement au modèle présenté par [7], l’objectif de celui-ci est de minimiser l’écart entre les quantités de minerai prévues et réelles. De plus, des modèles agrégés sont

résolus afin de réduire le plus possible la taille du modèle original. Cette stratégie (ou agrégation) a permis d’obtenir des solutions réalisables à près de 5% de l’optimalité en moins de 10% du temps requis par le modèle original. Par exemple, pour un des scénarios testés par les auteurs, l’utilisation de modèles agrégés résout le problème en 55 minutes au lieu de 36 heures. En 2016, les auteurs de [9] présentent un modèle de programmation linéaire stochastique pour de la planification à long terme tout en considérant l’incertitude reliée à l’aspect géologique du gisement. Pour une application plus récente d’un modèle MIP au problème de planification à long terme sur un horizon de 20 ans, voir l’article de Collard, Matinelli et Gamache [10].

2.1.2 Moyen terme

Une année planifiée dans le plan à long terme est découpée en périodes d’un à trois mois et une planification à moyen terme de la première période est établie. Le modèle de faible résolution avec microsélectivité (LRMS) dévoilé en [11] permet d’obtenir des planifications optimales pour des périodes mensuelles. Contrairement au modèle d’optimisation de la planification minière basée sur les ressources (RMSOP), celui-ci réussit à trouver des solutions entières pour toutes les tailles de problèmes testées et qui respectent le délai de résolution maximal de deux heures. [12] présente un modèle MIP qui utilise une approche intégrée entre la planification à moyen terme et la planification à court terme. Les auteurs affirment que cette intégration des deux types de planification dans un seul modèle permet d’obtenir un optimum global. Effectivement, un gain de 0.4% sur la valeur actuelle nette du projet est observé en utilisant cette approche intégrée lorsque comparé à une approche séparée. De même, un autre modèle MIP est présenté dans [13] pour la planification à moyen terme d’une mine souterraine en Irlande. Des méthodes exactes et heuristiques sont utilisées pour réduire la taille du modèle de base et ensuite une heuristique de décomposition est conçue pour l’optimisation du problème. Ce nouveau modèle permet de résoudre le problème de planification à moyen terme 98% plus vite, lorsque comparé au modèle MIP de base.

En 2020, une approche heuristique qui relève de la programmation par contraintes est présentée dans [14]. Au lieu de maximiser la valeur actuelle nette, l’objectif est plutôt de minimiser le temps de complétion total du plan de production. De plus, la contribution principale des auteurs est l’utilisation de contraintes de précédences si (*OR-precedences*) qui permettent de commencer des activités avec plusieurs prédécesseurs sans avoir besoin de terminer toutes les activités prédécesseurs. Cette nouveauté dans la planification permet au modèle d’être plus flexible et représente mieux la réalité où il est possible d’atteindre une partie du gisement par plusieurs chemins différents. Les résultats de l’heuristique montrent que plus de mètres

de roche sont creusés lorsque comparés aux résultats réels du plan de la mine dans un même horizon de temps.

2.1.3 Court terme

Cette planification possède généralement un horizon d'une à deux semaines et permet d'affecter les séquences de tâches journalières aux équipements mobiles dans la mine. Ce type de planification pour les mines souterraines est très peu présent dans la littérature. Il y a une dizaine d'années environ, un modèle MIP a été développé par [15] pour résoudre le problème de planification à court terme. En plus de résoudre le problème de planification à court terme, celui-ci effectue l'affectation des tâches aux équipements miniers. La flotte de véhicules présentée dans l'article est composée de cinq LHD et trois camions et seules les tâches de chargement de roche ont été considérées. L'avantage de ce modèle est la possibilité de résoudre en moins d'une minute le problème de planification lorsque des événements imprévus dans la réalité obligent une réoptimisation. Cependant, la simplicité du modèle ne permet pas de prendre en compte les opérations de développement comme le dynamitage, le boulonnage et autres qui permettent d'obtenir les chantiers de production. De plus, la localisation des tâches et des équipements n'est pas prise en compte par le modèle, ce qui rend peu pratique son application à une mine d'envergure cherchant à optimiser l'ensemble des opérations de développement et de production.

Plus tard, un modèle MIP de planification en temps réel pour une mine souterraine de potasse est développé [16]. Le problème est représenté sous la forme d'un problème d'ordonnancement de tâches hybride en *flow-shop*. Les plus petites instances de leur problème sont résolues avec CPLEX sans problème tandis que les problèmes de plus grande taille nécessitent l'utilisation d'heuristiques.

Le modèle MIP le plus récent est décrit en [5]. Contrairement au modèle présenté précédemment, celui-ci inclut les activités de développement et de production. Des contraintes de planification supplémentaires (*planning constraints extensions*) permettent au planificateur d'ajuster manuellement les temps de début et fin des tâches. Ces contraintes permettent d'ajuster le modèle lorsque des événements imprévus se produisent dans la réalité. De plus, l'aspect le plus intéressant présenté dans l'article est la capacité du modèle à interrompre les activités (*preemptive solution*). Par exemple, une activité de chargement de roche d'une durée de 100 heures peut être fragmentée en trois périodes de [10, 40, 50] heures. Cette contribution des auteurs permet ainsi au modèle de produire des planifications beaucoup plus réalistes. En effet, cet ajout permet de fragmenter une tâche dans des périodes libres qui pourraient ne

pas pouvoir accueillir la tâche complète. Les résultats dévoilent que le modèle P (*preemptive model*) obtient des solutions plus optimales que le modèle NP (*non-preemptive model*) dans tous les scénarios. Par contre, comme tous les autres modèles MIP, celui-ci ne peut pas être utilisé comme référence pour notre étude sur le trafic souterrain. Effectivement, ces approches utilisent des périodes d’une durée minimale de quelques heures voire d’une semaine. Donc, les solutions provenant de ces outils ne permettent pas d’obtenir un ordonnancement optimal des tâches pour les équipements mobiles tout en tenant compte de la congestion pouvant être produite à chaque minute.

[17] propose de résoudre le problème de planification à court terme pour les mines souterraines comme un problème d’ordonnancement de type *flow-shop*. L’idée ici est alors de représenter les opérations minières comme des tâches et les équipements mobiles comme des machines. Cette façon de représenter le problème permet de prendre en compte les déplacements des équipements mobiles comme des temps de préparation (*setup*). Toutes les opérations peuvent être prises en compte et la flotte complète des véhicules miniers aussi. La modélisation du problème de planification à court terme s’avère alors plus réaliste. En 2018, les mêmes auteurs présentent un modèle de programmation par contraintes pour résoudre le problème de planification sur un horizon de 36 heures [18]. La fonction objectif est de réduire le temps de complétion total des tâches. Un avantage majeur de l’emploi de la programmation par contraintes (CP) est son efficacité à résoudre un problème avec des contraintes non linéaires. De même, les auteurs ont pu ajouter les contraintes de dynamitage lors des changements de quart. Ces contraintes forcent toute production à arrêter lors des périodes de changement de quart. Ainsi, l’utilisation du CP permet d’obtenir des solutions plus réalistes. Tous les scénarios avec cinq points de soutirages sont résolus en moins de 10 minutes et les scénarios avec 10 points de soutirages en moins de 25 minutes. Cependant, certains points manquants au modèle pourraient grandement influencer les résultats. Tout d’abord, les activités en lien avec les chantiers de production ne sont pas incorporées au modèle. L’allocation de chargeuses LHD et camions qui s’y rattache pourrait modifier les séquences des tâches. Une autre remarque serait l’incapacité du modèle à interrompre les tâches lors des changements de quart et les faire reprendre à la suite du changement de quart. Un dernier point est l’absence de la notion de trafic souterrain associé aux déplacements des équipements mobiles qui engendre une augmentation des temps de complétion des tâches.

Un autre modèle de programmation par contrainte (CP) appliqué aux mines souterraines est présenté au chapitre 6 de [4]. L’outil permet de résoudre le problème de planification à court terme tout en considérant des objectifs de planification à moyen terme. Ce modèle considère les tâches de développement et de production sur un horizon d’un an afin de prendre en

compte les objectifs de la planification à long terme. De même, les auteurs utilisent l'outil ILOG CP développé par IBM [19] pour modéliser le problème. Cet outil de modélisation possède toutes les fonctions utiles aux problèmes de planification en programmation par contraintes. L'unité de base du modèle est la durée d'un quart de travail de 10 heures. Les résultats du modèle de programmation par contraintes surpassent ceux du modèle MIP présenté par les mêmes auteurs en [5] pour les cinq scénarios. Toutefois, cette représentation du problème de planification avec une unité de base de 10 heures n'est pas adaptée à notre cas où l'on cherche à planifier les effets du trafic qui peuvent varier d'une minute à l'autre. Pour des applications plus récentes voir [20] et [21].

2.1.4 Temps réel

L'intérêt de l'optimisation à temps réel est de déterminer l'affectation optimale d'activités à des ressources spécifiques au courant de la journée. L'objectif de ce type de planification est de minimiser le temps de complétion maximal des activités en respectant les liens de précedence entre les activités et la disponibilité de chaque ressource. Pour une revue complète sur le fonctionnement de la planification à temps réel le lecteur peut se référer à [22]. Un des premiers algorithmes de temps réel pour les mines souterraines est mis de l'avant dans [3] pour la gestion des chargeuses LHD. L'article présente les trois éléments clés d'un système de gestion de la flotte minière. Il y a la répartition des tâches aux véhicules (*dispatching*), l'orientation des véhicules (*routing*) et la planification (*scheduling*). La représentation du réseau de transport de la mine sous forme de graphe rend aisée la compréhension du problème au lecteur. Cette représentation en graphe a permis aux auteurs d'utiliser l'algorithme Dijkstra pour trouver les plus courts chemins entre les paires de noeuds à chaque fois qu'un véhicule doit se déplacer. Un autre avantage de ce modèle de plus courts chemins (PCC) est son niveau de précision quant à l'orientation des véhicules dans l'espace et sa capacité à éviter des situations de congestion sur les routes comme les face-à-face (*deadlocks*). Cependant, le prix à payer pour un tel niveau de précision est la complexité associée aux calculs des chemins optimaux à chaque fois qu'un véhicule est disponible. Le réseau présenté ne représente qu'une infime partie de la mine et ne permet la présence de quelques chargeuses LHD. Il serait alors très ardu, voire impossible, de transposer cette approche au réseau complet d'une mine avec toute sa flotte d'équipements mobiles.

Un outil d'optimisation du trafic en temps réel a été créé dans [23] où chaque véhicule de la mine possède un dispositif de pistage qui permet de connaître la position du véhicule à chaque moment de la journée. Une interface avec des feux de circulation indique aux camions de continuer leur chemin ou de s'arrêter pour laisser passer un camion ayant la priorité.

L'avantage de cet outil est alors la gestion de l'utilisation des routes lorsque les équipements mobiles possèdent, *a priori*, des séquences de tâches à effectuer dans la journée. Cependant, aucune optimisation de l'ordonnancement des tâches n'est mise de l'avant. Ainsi, l'outil en temps réel présenté ici ne peut qu'atteindre un optimal local en cherchant à minimiser le trafic provoqué par un ordonnancement qui ne considérerait pas le trafic.

Enfin, l'optimisation en temps réel est utilisée afin de réduire la congestion en guidant les opérateurs sur les actions à prendre sur la route. Ce type de planification permet aussi l'affectation d'équipements mobiles libre aux activités afin de réduire le temps de complétion total pendant la journée. Cependant, cette approche ne permet pas de modifier l'ordonnancement de toutes les tâches de la semaine. L'idée abordée dans ce mémoire est plutôt d'agir en amont lors de la planification à court terme afin d'ordonnancer les activités de la semaine tout en considérant le trafic qui en résulte.

2.2 Ordonnancement de tâches

Un problème d'ordonnancement consiste à séquencer de façon optimale les tâches et à affecter les ressources nécessaires à leur exécution tout en respectant les contraintes opérationnelles. Comme il est présenté dans [24], ce problème d'optimisation a été découvert bien avant 1960. Les travaux de recherche menés depuis plus de soixante-dix ans sur ce problème ont permis de distinguer plusieurs classes de problèmes d'ordonnancement (connus sous les termes anglais *flowshop*, *jobshop*, *openshop*, etc.) et leurs variantes. Le problème de planification à court terme dans les mines souterraines s'apparente à un problème de *flexible jobshop*. Les tâches sont composées de différentes opérations qui doivent être exécutées dans un ordre particulier. Par exemple, la tâche de développement qui consiste à effectuer des avancements dans une galerie est constituée des opérations suivantes : chargement de la roche, boulonnage de la galerie, nettoyage du plancher, forage, remplissage des trous avec un mélange explosif et dynamitage. Chacune de ces tâches doit être affectée à une équipe de travail (machine et ressources humaines) particulière. Pour le problème de planification à court terme dans les mines souterraines, plusieurs équipes de travail non homogènes peuvent exécuter chacune des tâches. Chaque équipe de travail ne peut exécuter qu'une seule tâche à la fois. Il existe en plus des contraintes de précédence entre les opérations de chaque tâche où l'opération $i + 1$ ne peut débuter que si l'opération i est terminée. Dans ce type de problème, l'objectif le plus commun consiste à minimiser le temps de fin pour exécuter toutes les tâches (mieux connu en anglais sous le terme *makespan*). Le problème de *jobshop* a été identifié comme NP-difficile et reste un sujet d'actualité. En effet, il est possible de retrouver le problème de séquençage

dans le domaine de l’informatique où l’on cherche à optimiser l’affectation des tâches de calcul entre plusieurs processeurs disponibles. Dans [25], deux algorithmes (HEFT et CPOP) sont présentés à cet effet. Plus tard, l’algorithme génétique CPGA est développé dans [26] pour le même problème et permet des gains de performance grâce à une meilleure parallélisation des tâches entre les processeurs. Une revue globale des plus récentes techniques en planifications de tâches appliquées à ce domaine a été faite par [27].

2.2.1 Gestion de projets avec contraintes de ressources limitées

Dans le problème de planification à court terme dans les mines souterraines, il existe en plus des contraintes de précédence entre certaines tâches ce qui fait que ce problème s’apparente également au problème de gestion de projet avec contraintes de ressources. Les outils de planification présentés dans ce mémoire font partie de la famille des problèmes de gestion de projets avec contraintes de ressources limitées (RCPSP). Tout comme le RCPSP de base présenté dans [28], les données d’entrée de notre problème, par exemple les temps de tâches, sont déterminées d’avance. Il existe toutefois des différences notables entre nos modèles et un modèle RCPSP de base. Dans le problème de planification à court terme dans les mines souterraines, il est possible d’interrompre l’exécution de certaines opérations et de les reprendre plus tard. On parle alors d’un problème d’ordonnancement avec préemption. Plusieurs équipements mobiles (ressources) de la mine ne peuvent être dédiés qu’à un seul type d’activité, ce qui constitue un cas particulier du RCPSP. Également, il y a l’utilisation d’une fonction multiobjectif dans nos outils de planifications avec contraintes de trafic. Une dernière différence majeure est la prise en compte de l’emplacement spatial des activités qui oblige la notion de déplacement des ressources dans le problème.

2.2.2 Programmation par contraintes

La programmation par contraintes permet de résoudre des problèmes combinatoires linéaires ou non linéaires. Cette approche s’est avérée particulièrement efficace pour résoudre des problèmes de planification et d’ordonnancement. Ce paradigme de programmation a permis de grandement accélérer et faciliter la résolution de plusieurs problèmes de planification autrefois envisagés sous la forme de programmes linéaires : ([29], [30] et [31]) pour la fabrication d’horaire dans le domaine de la santé, ([32], [33] et [34]) pour le domaine de l’informatique, ([35] et [36]) pour le problème de planification des trains, ([37], [38] et [39]) en gestion de projet, [40] pour aider les hôpitaux lors de la pandémie de COVID-19, [41] pour l’industrie automobile, [42] pour les problèmes de regroupement (*clustering*), etc.

Un logiciel de programmation par contraintes dédié à la planification est créé par IBM [19]. D'ailleurs, IBM a consacré beaucoup d'efforts pour développer ce solveur efficace pour les problèmes d'ordonnancement. À ce jour, il s'agit d'un des solveurs les plus efficaces et des plus complets sur le marché pour résoudre divers problèmes d'ordonnancement. L'extension Python du logiciel est utilisée dans [4] et dans le présent mémoire pour de la planification appliquée aux mines souterraines. Une autre application de la programmation par contraintes au domaine des mines souterraines est présentée dans [18].

2.3 Simulation dans les mines

La simulation à événements discrets a permis à plusieurs auteurs de modéliser leur processus sur ordinateur tout en considérant la variabilité et le dynamisme présent dans la réalité. Pour les mines à ciel ouvert, [43] utilise un modèle de simulation pour analyser la performance de certaines heuristiques de *dispatch* sur le système camion-pelle (*truck-shovel system*). [44] insère la notion de trafic dans le modèle de simulation afin de mieux représenter le système camion-pelle dans les mines à ciel ouvert. Grâce à ce modèle, les auteurs ont réussi à déterminer des solutions diminuant la congestion à la surface en optant pour le chemin de plus court sur la base du temps et non de la distance à parcourir. Cette stratégie permet ainsi d'augmenter la productivité de la mine. Une analyse de la performance de plusieurs heuristiques présentes dans la littérature est faite dans [45] à l'aide d'un modèle de simulation à événements discrets. Un autre exemple plus récent est l'utilisation de la simulation et du facteur de correspondance hétérogène (*heterogeneous match factor*) pour analyser des heuristiques de dispatch pour le système camion-pelle [46].

Des applications de la simulation sont retrouvées dans le domaine des mines souterraines aussi. [47] utilise la simulation pour établir des stratégies de dispatch dans l'éventualité qu'une mine souterraine possède une flotte automatisée. Une autre étude utilise l'approche par signal pour dicter le partage des routes [48]. Chaque intersection possède un ensemble d'expressions pour établir quel segment de route peut avoir le feu vert pour passer. À première vue, l'approche exacte des auteurs semble être précise. Cependant, un point faible d'une telle approche est la quantité de temps mise à la création du modèle de simulation. De plus, les mines souterraines sont des environnements en changement constant. Il serait intéressant de voir la complexité de la mise à jour du modèle des auteurs pour représenter la même mine dans le futur. Contrairement à cette approche, nous avons opté pour une approche par itération qui pourrait être un peu moins précise, mais définitivement plus rapide à développer et plus facile à adapter avec l'agrandissement de la mine au fil du temps.

L'article [6] démontre que le trafic dans les rampes est un élément majeur contribuant à la productivité des mines souterraines grâce à un modèle de simulation. D'ailleurs, les auteurs ont été capables d'identifier plusieurs états d'équilibre dans la rampe en analysant les graphiques de position des véhicules en fonction du temps. Certaines limites sont imposées telles que : seuls les déplacements dans la rampe sont considérés et aucune variabilité n'est utilisée pour les vitesses de déplacement et les temps d'opération. De plus, la flotte est seulement composée de 1 à 5 camions et ceux-ci se dirigent tous au même point de chargement. Sans ces limitations, il serait peu probable de retrouver les mêmes motifs voire même un état d'équilibre dans le déplacement des équipements mobiles. [49] mentionne aussi l'importance d'une bonne gestion de la congestion des rampes souterraines. Une version améliorée de la politique de verrouillage (*lock-out policy*), qui est essentiellement l'utilisation de feux de circulation, est proposée par les auteurs afin d'augmenter le flot de camions dans la rampe. D'autre part, l'article [50] propose un modèle de simulation pour comparer l'efficacité d'utiliser des chutes à minerai (*ore passes*) à celle d'utiliser une flotte de camions pour transporter le minerai à la surface selon plusieurs tailles de mine.

Un modèle de simulation fait avec le logiciel SimMine est utilisé pour représenter une mine comportant trois LHD et trois camions [51]. L'objectif est de tester plusieurs scénarios avec différentes quantités de camions. Ils ont observé qu'en augmentant le nombre de camions de trois à six la productivité a grandement augmentée. Après six camions, les gains de productivité sont de moins en moins intéressants, voire nuls. Ceci s'explique du fait que le trafic causé par le partage de la route augmente lorsque le nombre de camions est augmenté. De plus, les chargeuses LHD deviennent saturées et des camions doivent attendre à ne rien faire. Contrairement aux modèles présentés précédemment, [52] utilise un modèle de Monte-Carlo. Il s'agit ici d'un outil de simulation statique dans le temps et stochastique (aléatoire). L'outil est utilisé afin de tester des stratégies de planification à moyen et long terme. Il aurait été impossible d'utiliser un tel modèle pour du court terme où le dynamisme est un facteur primordial. Un simple exemple est la perte de temps et de productivité associés à l'interaction des équipements mobiles au fil du temps.

2.4 Méthodes choisies

Finalement, l'optimisation de la planification dans les mines souterraines comporte plusieurs volets différents en fonction de l'horizon de la planification. Compte tenu de l'efficacité de la programmation par contraintes, cette approche sera utilisée pour modéliser le problème d'optimisation de la planification à court terme dans les mines avec des contraintes de trafic.

Également, la simulation est un outil d'aide à la décision couramment utilisé afin d'analyser les effets engendrés par des modifications sur le processus original. Par exemple, l'effet d'une nouvelle méthode de planification de tâches sur le temps de complétion total du plan. Ainsi, un modèle de simulation sera développé afin de valider les résultats obtenus avec le nouveau modèle qui prend en compte le trafic.

CHAPITRE 3 MÉTHODOLOGIE

Ce chapitre se concentre sur les étapes nécessaires à la création des contraintes de trafic pour l’outil de planification. Comme il a été défini au chapitre précédent, les outils de planification à court terme utilisés jusqu’à ce jour ne considèrent pas les pertes de temps associées au partage de la route sous terre. L’idée mise de l’avant est d’employer un simulateur à événements discrets afin de mesurer les effets du trafic sur la productivité du processus de transport du minerai vers la surface. Une fois le simulateur validé avec les données de la mine, celui-ci peut être considéré comme un environnement de test pour les plans générés par l’outil de planification. L’intuition revient à simuler plusieurs plans de production différents afin d’obtenir des données de trafic qui seront récoltées à une certaine fréquence tout au long des simulations. Plusieurs cycles du processus présenté à la figure 3.1 sont effectués avec des besoins hebdomadaires différents provenant d’une heuristique pour acquérir un ensemble de données diverses. Par la suite, les données seront analysées pour créer des contraintes quant au trafic. Le chapitre comporte ainsi les sections suivantes : le modèle de simulation et l’outil de planification à court terme.

3.1 Le modèle de simulation

La modélisation à événements discrets employée dans le contexte présent évalue l’état du système à chaque occurrence d’un événement [53]. Par exemple, l’arrivée d’un véhicule à une intersection représente un nouvel état du système. Ainsi, lorsqu’arrivé à l’intersection, le véhicule peut poursuivre dans la logique de processus comme vérifier si le tronçon de route suivant est libre pour ensuite s’y engager. Afin de garantir un modèle fidèle à la réalité, plusieurs étapes clés doivent être suivies [54]. Cette section est en lien avec le premier sous-objectif qui est de modéliser le trafic en lien avec les activités de production. Les sous-sections suivantes s’attardent sur l’analyse du processus, les données d’entrée, la construction et la validation du modèle. Les dernières étapes de design, d’analyse des résultats et de recommandations finales de [54] seront présentées au chapitre 4.

3.1.1 Analyse du processus de transport de minerai

Afin d’établir la cartographie du processus de la mine souterraine, il est important d’avoir une bonne vue d’ensemble au point de vue macro. En outre, pour acheminer la roche à la surface, une succession d’étapes de transport et d’opérations est nécessaire. Cette information

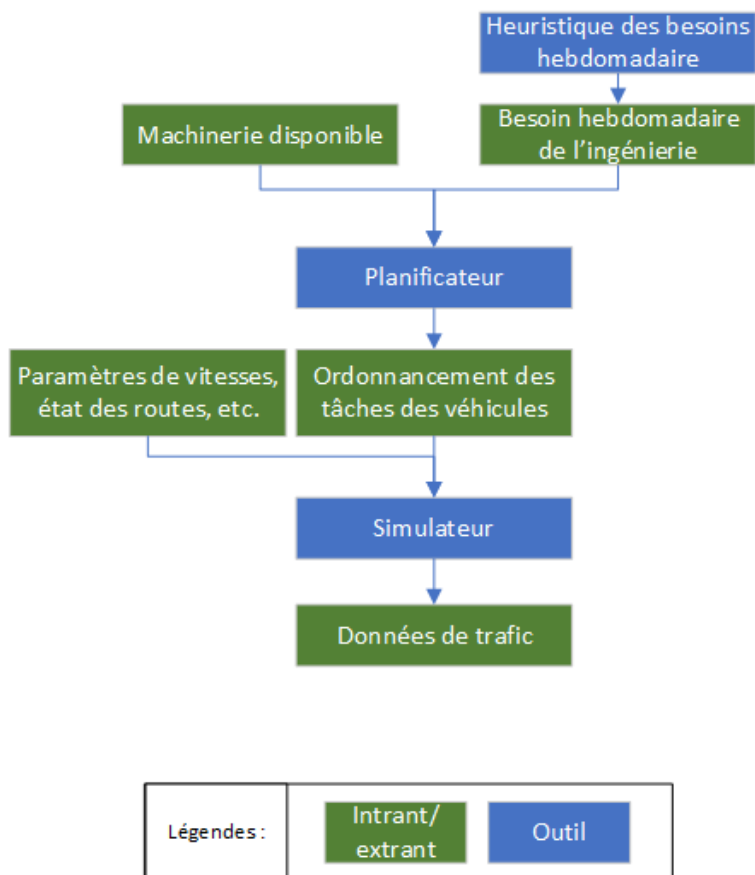


Figure 3.1 Le processus d'acquisition de données de trafic pour la mine étudiée.

à grande échelle est obtenue par l'administration des opérations minières, majoritairement localisée au site minier. Par conséquent, une présence physique à la mine est primordiale afin de capter tous les détails. De plus, les visites souterraines avec les opérations permettent d'assurer la justesse du processus à cartographier. Ainsi, un organigramme logique du transport du minerai est créé dans le simulateur à la fin de cette étape. L'ensemble des logiques du modèle se trouve à l'annexe A. Le modèle de simulation se compose de cinq sections. La section 1 est la création des entités qui représentent les véhicules miniers. Une fois les entités créées au début de la simulation, celles-ci ne retournent plus dans cette section. La section 2 permet l'affectation de tâches aux véhicules tout au long de la simulation. Donc par exemple, lorsqu'un véhicule termine une tâche, son entité est envoyée à cette section pour savoir quelle sera sa prochaine destination et sa prochaine tâche. Ensuite, la section 3 comporte tous les déplacements horizontaux et verticaux des véhicules dans les rampes et les niveaux. Une fois que l'entité entre dans cette section, elle est soit dirigée dans la logique des déplacements dans les rampes ou la logique de déplacement dans les niveaux. Lorsqu'il s'agit d'un dépla-

cement intra-niveau, l'entité reçoit une station cible à atteindre d'une heuristique de plus court chemin jusqu'à ce que la station finale où la tâche se trouve est atteinte. La section 4 est la logique des opérations minières comme le chargement de roche et le déchargement de celle-ci et la section 5 est la logique des changements de quarts de travail qui obligent l'arrêt de toute activité.

3.1.2 Les données d'entrée

D'emblée, trois types de données sont importées pour simuler la mine. Les valeurs sont toutes entreposées dans un fichier Excel principal avec lequel le modèle Arena importera son contenu. En premier lieu, il y a les paramètres de la flotte de véhicule tels que les capacités, les vitesses, les priorités des véhicules pour le partage de la route, etc. Des distributions triangulaires sont utilisées pour la plupart de ces paramètres afin de considérer l'aspect stochastique présent dans le processus d'extraction du minerai. Ces distributions ont été établies à l'aide d'avis d'experts de la mine. La flotte est composée de camions, chargeuses LHD, boulonneuses, foreuses (*jumbo*) et camions à émulsion. En deuxième lieu, il y a les paramètres d'environnement tels que les longueurs des segments de route, les surfaces de roulement des routes et les états initiaux des points de soutirage (*drawpoints*) au démarrage de la simulation. Des distributions triangulaires avec une faible variabilité, basées sur l'avis d'experts, sont utilisées pour les surfaces de roulement des routes. Étant donné que les routes ne sont pas parfaitement planes, les véhicules peuvent être ralentis s'ils entrent en contact avec des trous sur la route. Finalement, il y a les paramètres de séquences de tâches de travail qui déterminent les positions, les étapes et les temps des tâches que chaque véhicule devra effectuer. Étant donné le grand nombre de paramètres, les tableaux suivants présentent seulement les paramètres jugés les plus importants.

Les paramètres de la flotte

Ces paramètres cherchent à reproduire les caractéristiques physiques de la flotte de véhicules. L'information quant aux valeurs de ces paramètres provient de fichiers CSV des départements de la mine et de l'ingénierie. Également, de multiples entrevues ont été faites avec les employés du département de la mine pour récolter les données manquantes. Cela dit, les paramètres de la flotte présentés au Tableau 3.1 sont des expressions distribuées à titre d'attributs aux entités, soit les véhicules miniers, dans Arena.

Tableau 3.1 Les paramètres attribués aux types de véhicules de la flotte.

Paramètre	Type de véhicule	Description
Capacité en tonnes de minerais	Camion et LHD	Tonnage transportable par chaque type de véhicule.
Temps de remplissage camion	Camion et LHD	Matrice des temps de remplissage entre différentes tailles de camions et de LHD en minutes.
Priorité sur la route	Tous	Lorsque le paramètre vaut 1 pour un certain type de véhicule, tous les véhicules de ce type ont la priorité sur les segments. L'inverse demeure lorsque le paramètre vaut 0.
Temps de remplissage d'essence et d'urée	Tous	Temps en minutes nécessaire pour chaque type de véhicule afin de remplir le véhicule d'essence et d'urée.
Vitesse de déplacement	Tous	Vitesse de déplacement en km/h.
Coefficient de ralentissement en montée	Tous	Coefficient entre 0 et 1 qui ralentit la vitesse de déplacement lorsque le véhicule monte une rampe.

Les paramètres d'environnement

Contrairement aux paramètres de flotte, les paramètres d'environnement sont utilisés afin de répliquer la structure et l'état de la mine. Afin de bien visualiser ces paramètres, il faut imaginer la mine comme un immense graphe 3D. Les nœuds de ce graphe sont appelés des stations au fil de cet ouvrage, car les entités se déplacent dans le graphe pour aboutir à des nœuds de *drawpoint* (DP) ou d'épentes inférieures (*footwall*) (FW) qui sont leurs stations de travail. Toutefois, même si les nœuds des rampes sont nommés stations, le modèle ne permet pas l'allocation de tâches de travail dans lesdites stations. Les arcs sont appelés des segments. Ensuite, il est expliqué plus loin dans l'ouvrage que les paramètres d'environnement sont couplés aux variables de transport et aux logiques de transport afin de répliquer fidèlement la mine.

La plupart des paramètres d'environnement présentés au Tableau 3.2 proviennent d'entrevues avec des employés. Aussi, les états initiaux des stations proviennent de l'outil de planification lorsqu'un plan d'ordonnancement est généré. Ce paramètre ne peut être touché manuellement à risque d'entraver le fonctionnement de la simulation.

De surcroît, les longueurs des segments des niveaux ne sont pas présentées dans le tableau puisqu'elles ont été rentrées directement dans Arena. Il est à signaler que contrairement aux segments des rampes, la plupart des longueurs des segments des niveaux sont similaires.

Tableau 3.2 Les paramètres d’environnement provenant du fichier Excel principal.

Paramètre	Description
État initial	Valeur entre 0 et 100 permettant d’établir l’état initial d’une station. Chaque valeur représente le type de tâche devant être réalisé à la station. Par exemple, une valeur de 0 signifie que du chargement de roche pour le premier dynamitage doit être fait. Une valeur de 1 symbolise le boulonnage de la face de travail suite au premier dynamitage. La valeur 7 représente du chargement de roche pour le second dynamitage et ainsi de suite.
Surfaces de roulement des niveaux	Un seul coefficient par niveau qui varie entre 0 et 1 afin de diminuer la vitesse de déplacement des véhicules selon l’état général des segments.
Surfaces de roulement des rampes	Un seul coefficient par segment de rampe qui varie entre 0 et 1 afin de diminuer la vitesse de déplacement des véhicules selon l’état du segment.
Longueurs des segments de rampe	Longueur en mètre entre deux stations consécutives d’une rampe.

D’ailleurs, afin d’alléger le modèle, les longueurs entre les stations FW sont fixées à 19 m et les longueurs entre les stations de FW et DP à 7 m. Ces deux valeurs ont été choisies étant donné la fréquence à laquelle elles survenaient lors de l’analyse de l’architecture de la mine. Seules les longueurs des segments de coupe transversale (*crosscut*) et d’entrée de niveau peuvent différer entre les niveaux.

Les paramètres de séquençement

Enfin, chaque véhicule possède une liste de tâches à faire dans la journée. C’est ici que les quatre paramètres de séquençement générés par l’outil de planification entrent en jeu. Il y a la position du niveau, la position de la station sur le niveau, la durée de la tâche en minutes et l’étape à laquelle la tâche doit être réalisée. De plus, les paramètres de séquençement contiennent aussi l’information quant à la position initiale des véhicules au début de la simulation. Chaque véhicule possède un attribut d’étape de séquence (*A_Seq_Step*) qui agit à titre d’index pour savoir à quelle tâche l’entité est rendue. À l’aide de l’exemple affiché à la figure 3.2, il est possible de remarquer à la première ligne que la boulonneuse #3 est localisée au niveau 2 à la station 165 et que la longueur et l’étape de la tâche sont de zéro. Ceci indique alors qu’il s’agit bel et bien de la position initiale du véhicule au début de la simulation. Par la suite, à la deuxième ligne se trouve la première tâche qui elle se situe au

niveau 7 à la station 132, donc la boulonneuse se déplacera de sa position courante pour se rendre à la station de travail suivante. Lorsque l'index de l'entité arrive à la dernière ligne de zéros cela signifie que le véhicule a terminé toutes ses tâches. Il est à mentionner que les données présentées n'utilisent pas la terminologie de la mine afin d'en assurer son anonymat.

Il est à mentionner que dans la réalité il arrive souvent que des délais de production soient produits par des événements imprévus. Par exemple, un bris mécanique d'un camion dans la rampe 1 qui arrête tout véhicule voulant remonter à la surface par cette rampe. Ces événements ne sont pas simulés pour limiter la complexité du modèle. Ainsi, lorsque l'outil de planification génère les séquences de travail pour les véhicules, celles-ci sont définitives tout au long de la simulation.

Boulonneuse #3			
Level	Station	Size	Task
2	165	0	0
7	132	290	2
3	137	290	8
10	167	290	8
0	0	0	0

Figure 3.2 Exemple des paramètres de séquençement des tâches d'un véhicule.

3.1.3 La construction du simulateur

Le fonctionnement du modèle est fondée sur le concept de la simulation par agents. Selon [55], deux types de logiques sont utilisées pour ce type de problème. En premier lieu, il y a les logiques de décision où les entités vont effectuer certaines actions en fonction de critères. Par exemple, s'il ne reste plus de minerai dans le point de soutirage, le LHD se dirigera vers un autre point de soutirage. En second lieu viennent les logiques de contrôle qui fonctionnent avec des entités fictives afin d'exercer un contrôle sur les entités actives. Notamment, il y a l'entité des quarts de travail qui fige les véhicules à leur position respective à la fin d'un quart dans la simulation pour répliquer l'effet des arrêts de production entre les quarts. De plus, un tableau de bord présenté en annexe à la figure A.22 est mis sur place afin de vérifier le déplacement des camions et l'état de plusieurs indicateurs de performance lors des simulations. Cette vérification se fait tout au long de la construction du modèle afin de corriger les erreurs de logique possibles. Par exemple, le tableau de bord permet observer si le camion A se déplace vers la station de travail qui lui a été attribué. Cela dit, les parties suivantes traitent des limites et des cinq types de sections présentes dans le modèle.

Les limites du modèle

Comme il est signalé dans [6], les données de sortie de tout simulateur sont une approximation de la réalité. De fait, le modèle possède des limites afin d'arriver à modéliser les parties cruciales du processus d'extraction de minerai dans les temps convenus. Les limitations suivantes ont été utilisées :

1. seules les activités de développement et de production assurées par les chargeuses, camions, boulonneuses, *jumbo* et camions à émulsion sont simulées ;
2. aucune maintenance n'est nécessaire pour la flotte de véhicules ;
3. un véhicule reste sur place jusqu'à ce que sa prochaine tâche devienne accessible ;
4. le remaniement du stéril (*remucking*) pour accélérer la phase de soutirage n'est pas permis ;
5. le remplissage d'essence et d'urée est fait en même temps et dépend du nombre de tâches complétées et non du kilométrage ;
6. les temps de tâches ne possèdent pas de variabilité sauf pour les temps de chargement entre les divers types de chargeuses et de camions ;
7. lors des déplacements des véhicules, les accélérations et décélérations ne sont pas considérées ;
8. les voies d'évitement des rampes et les stations *footwall* et *drawpoint* peuvent contenir plus qu'un véhicule si nécessaire ;
9. lorsqu'il y a un laissez-passer sur un niveau, le déplacement du véhicule qui bloque le chemin n'est pas simulé. Cependant, un délai d'une minute est ajouté au véhicule pour prendre en compte ce déplacement qui arriverait en réalité ;
10. tous les camions possèdent la même capacité en roche étant donné que l'outil de planification ne peut considérer qu'un seul type de camion sans modifier les temps de tâche.

L'initialisation et les logiques de contrôle

Cette section décrit la création de toutes les entités utilisées dans la simulation. D'une part, il y a la création des entités représentant les véhicules à la section 1.0 du modèle Arena auxquelles plusieurs valeurs d'attributs sont fixées. Le diagramme de flux de cette section est présenté à la figure A.1 Ces entités passent ensuite à la section 1.1 (A.2) du simulateur pour comptabiliser le nombre total de tâches dans leurs séquences. Elles iront ensuite à la section 2.0 (A.4) du simulateur. D'autre part, il y a les entités de contrôle telles que les changements

de quart de travail et l'enregistrement des données de trafic. D'ailleurs, dans la section 1.2 (A.3) de l'outil se trouve la création des quarts de travail qui changent l'état de la variable V_Shift_Change à chaque 12 heures. Lorsque cette variable est active, toutes les entités du système quittent leurs modules pour se rendre à la section 5.0 (A.21) du logiciel, soit les files d'attente lors des changements de quart, afin d'arrêter toutes les activités. Enfin, les entités d'enregistrement de données sont créées à chaque 30 secondes afin de capter les valeurs de variables comme le nombre de chargeuses actives sur un niveau au moment t . Ces valeurs de variables sont alors exportées dans un fichier CSV. De fait, c'est ce fichier qui est exploité pour construire la contrainte de trafic pour l'outil de planification présenté à la section 3.2 du présent ouvrage.

L'indexation des tâches

Le but de cette partie du modèle est d'affecter les véhicules à leur emplacement de travail. En utilisant l'index A_Seq_Step et la séquence de tâches de travail de l'entité, une vérification est faite pour voir si la tâche à effectuer par l'entité peut être réalisée ou si l'entité doit attendre que l'opération à effectuer à la station de travail possède le même index que la tâche qu'il doit effectuer. Par exemple, si la boulonneuse #6 est disponible pour effectuer la tâche 8 à la station 340 du niveau 11 et que la variable de phase ($V_Station_Phase(340, 11)$) vaut 8, l'entité se dirigera vers cet emplacement pour effectuer la tâche. Par la suite, les attributs de position de l'entité sont mis à jour pour assurer ce déplacement de l'emplacement actuel vers le prochain. Ces attributs sont présentés au tableau 3.3 avec une courte description expliquant quelles seront les valeurs prises par ceux-ci. Dernièrement, si l'entité a terminé toutes ses tâches, c'est à cette section qu'elle sera sortie du modèle. L'ensemble des diagrammes de flux de cette section est présenté aux figures A.4 à A.8.

Les logiques de transport

Deux méthodes de déplacement des entités entre les stations de la mine ont été analysées et discréditées afin de trouver la méthode qui sera employée dans ce mémoire.

Méthode 1

D'emblée, il y a l'option des transporteurs guidés offerte par le logiciel Arena. Le chemin optimal à suivre entre deux stations est généré automatiquement pendant la simulation (voir [54]). Cependant, une fois que le transporteur de l'entité est en mouvement, celui-ci ignore les autres véhicules sur sa route et entraîne rapidement des erreurs système de face-à-face (*deadlock*). De telles erreurs arrêtent la simulation. Il est alors impératif que la méthode

Tableau 3.3 Les principaux attributs mis à jour lors de l'indexation des tâches.

Attribut	Description
A_Previous_Level	Prend la valeur de l'attribut <i>A_Position_Level</i> avant que celui-ci prenne la valeur du nouvel emplacement.
A_Previous_Station	Prend la valeur de l'attribut <i>A_Position_Station</i> avant que celui-ci prenne la valeur du nouvel emplacement.
A_Position_Level	Prend la valeur de niveau du nouvel emplacement.
A_Position_Station	Prend la valeur de station du nouvel emplacement.
A_Position_Size	Prend la valeur de temps de tâche du nouvel emplacement.
A_Position_Task	Prend la valeur d'étape de tâche du nouvel emplacement.

permette aux véhicules de céder le passage lorsqu'il en est nécessaire.

Méthode 2

Ensuite, la deuxième méthode discréditée était de générer d'avance tous les plus courts chemins possibles entre les stations avec l'algorithme de Dijkstra. Une vérification de variables de transport est alors faite à chaque nœud pendant le trajet pour savoir si le prochain segment de route est libre d'accès. Les problèmes de *deadlock* observés à la méthode 1 sont résolus puisqu'ici les variables de transport permettent des logiques de laissez-passer. Toutefois, cette méthode implique l'importation de grandes quantités de chemins possible. Effectivement, puisqu'il peut y avoir jusqu'à 600 emplacements de travail dans un niveau qui possède trois *footwalls* et qu'il existe $n(n-1)$ chemins orientés optimaux possibles, la phase d'importation par Arena prend une trentaine de minutes. Ainsi, ce délai d'importation nuit grandement à la création, au débogage, à la validation et à l'exploitation du modèle puisqu'il faut toujours attendre une demi-heure avant que celui-ci ne s'active.

Méthode finale

L'idée finale a été de garder le principe de vérification de variables de transport de la méthode 2 et de créer une heuristique dans un module VBA du modèle qui donne le plus court chemin à entreprendre sur chaque niveau. En bref, l'heuristique utilise les intrants qui sont les attributs de la position actuelle du véhicule et de la position de la station finale à atteindre sur le niveau. Grâce à ces intrants, il est possible de donner une station cible (*A_Target_Node*) à atteindre lorsqu'un déplacement vertical doit être fait. Toutefois, il existe une exception à la règle. En effet, si la station finale se trouve sur un *footwall* au lieu d'un *drawpoint* et que l'entité

est située sur ce *footwall*, la cible sera la station finale même s'il n'y a pas de déplacement vertical. Ici les termes de déplacements horizontaux et verticaux sont utilisés pour visualiser le déplacement du véhicule sur le niveau en 2D, comme montré dans les figures suivantes. À l'aide d'une logique itérative, l'entité Arena continue de passer dans le module VBA à chaque fois qu'elle a atteint une station cible qui n'est pas la station finale. Trois exemples sont abordés pour bien comprendre la logique d'attribution de ces stations cibles. Sans oublier que la phase d'importation par Arena prend maintenant moins de deux minutes au lieu de 30 minutes avec la méthode 2. L'ensemble des diagrammes de flux représentant cette section du modèle de simulation est présenté aux figures A.9 à A.16.

Trois exemples de calcul du chemin optimal

Lorsqu'il y a qu'un seul *footwall*, les véhicules ne peuvent avoir qu'un seul chemin possible entre deux stations sur le niveau comme il est montré à la figure 3.3. Donc, la station cible serait alors B puisqu'il n'y a aucun autre déplacement vertical à effectuer avant d'atteindre la station finale B.

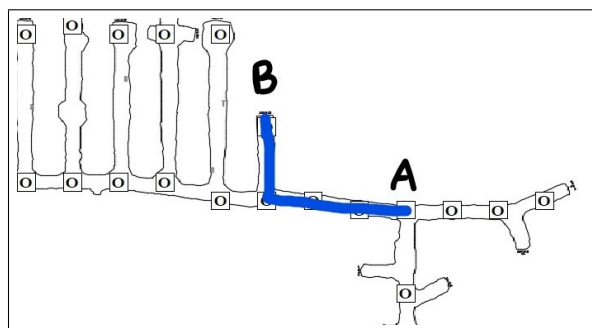


Figure 3.3 Exemple du chemin possible entre A et B sur un niveau à un *footwall*.

Par la suite, il y a les niveaux à deux *footwalls*. Comme il a été présenté à la sous-section 3.1.2 aux paramètres d'environnement, des *crosscuts* sont utilisés pour la jonction entre les *footwalls*. D'ailleurs, deux *crosscuts* de distance identique sont présents entre deux *footwalls* dans la mine en général. Ainsi, les véhicules peuvent avoir deux chemins possibles entre deux stations qui ne se trouvent pas sur le même *footwall*. De surcroît, il est possible d'observer ici que pour passer de A vers B il sera nécessaire d'appeler deux fois le module VBA puisqu'il y a deux segments verticaux à traverser. À la première entrée du module, la station cible donnée sera de prendre un *crosscut* pour passer du FW1 au FW2. Une fois le *crosscut* traversé, la deuxième station cible sera la station B. Cela dit, le chemin optimal sera celui qui aura la distance horizontale la plus petite puisque les distances verticales sont les mêmes sur les *crosscuts*. Le chemin optimal du cas de la figure 3.4 est le tracé en bleu dont la première station cible sera la station du *crosscut* de gauche sur le FW2. Une fois le *crosscut* traversé,

la station cible est la station B.

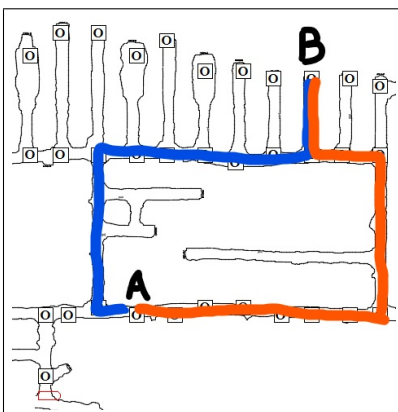


Figure 3.4 Exemple de deux chemins possibles entre A et B sur un niveau à deux *footwalls*.

Enfin, les niveaux à trois *footwalls* possèdent quatre crosscuts qui sont alignés verticalement. L'heuristique va toujours analyser deux chemins possibles. Les tracés sur la figure 3.5 dénotent les deux chemins utilisés pour le calcul du chemin optimal pour passer de la station A vers la B. Notamment, soit le véhicule prendra les deux *crosscuts* de gauche dans le tracé orange ou les deux de droite dans le tracé bleu. Dans la réalité, le tracé orange serait le passage de A vers B en utilisant le FW2 et ensuite le FW1. Cependant, cette distinction est mise à l'écart puisque le chemin a une distance totale identique au tracé orange. Enfin, tout comme pour les niveaux à deux *footwalls*, le chemin optimal est celui avec la distance horizontale la plus petite, soit le tracé bleu ici. La première cible est la station du *crosscut* de droite sur le FW2, la deuxième cible sera la station du *crosscut* de droite sur le FW1 et la dernière cible sera la station B sur le FW1.

Variables de transport

Maintenant, il est nécessaire d'ajouter des règles de vérifications de l'état des routes lors des déplacements vers les stations cibles et finales afin de ne pas avoir de collisions entre des véhicules qui traversent en sens inverse. Cette logique est assurée par des variables de transport qui sont associées aux segments et dans deux types de directions, soit Nord-Ouest et Sud-Est. Par exemple, si un véhicule veut passer de la station A vers la station B qui est au sud de A, une vérification est faite à la variable Nord-Ouest du segment qui relie A et B. Si la voie est libre, soit une valeur de 0 à la variable Nord-Ouest, la valeur de la variable Sud-Est est incrémentée de 1 et le véhicule se déplace ensuite. Une fois le déplacement terminé, la variable est décrémentée de 1.

Si la voie n'est pas libre et qu'il s'agit d'un segment de niveau, le véhicule attend jusqu'à ce

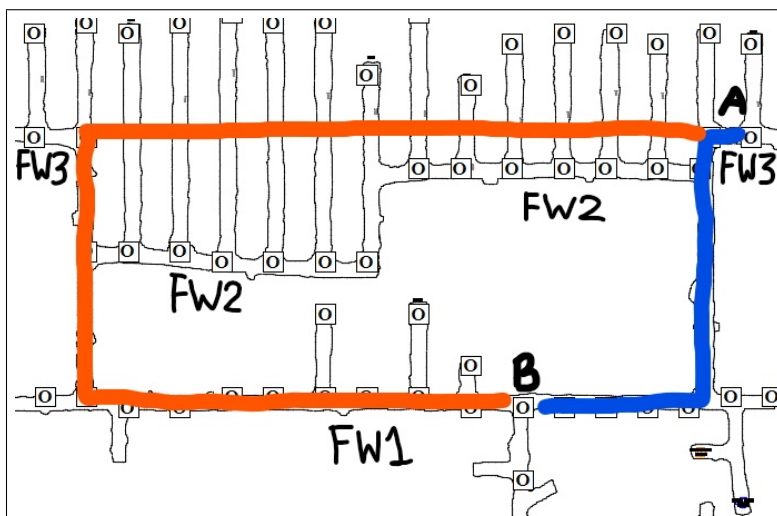


Figure 3.5 Exemple de deux chemins considérés entre A et B par l’heuristique sur un niveau à trois *footwalls*.

que la variable soit à 0. Il est supposé que le véhicule puisse se déplacer quelque peu sur un autre segment lié à cette station pour ne pas bloquer le chemin du véhicule arrivant en sens inverse. Ainsi, ce petit mouvement n’est pas inclus dans la simulation, car il est effectué en temps caché avec l’attente de libération de la voie.

Si la voie n’est pas libre et qu’il s’agit d’un segment de rampe, le véhicule doit se déplacer dans une voie de dépassement (*remuck point*) pour dégager la voie. Cette logique est incluse dans la simulation puisque contrairement aux niveaux, il y a moins d’intersections et de segments dans les rampes. De plus, il pourrait être intéressant de comprendre à quelles portions des rampes il y a le plus de temps perdu pour possiblement trouver des solutions quant à la modélisation du trafic dans l’outil de planification.

Variables de noeuds

Afin de vérifier le bon fonctionnement du modèle, il est nécessaire de bien visualiser le mouvement des véhicules dans le simulateur. Les véhicules sont animés lors des déplacements dans les rampes, mais pas dans ceux des niveaux. Des variables pour chaque noeud des niveaux sont alors créées pour voir le déplacement des entités vers leurs stations de travail. Un exemple avec le niveau 425 est montré à la figure 3.6. Lorsqu’un camion arrive à un noeud i , la variable de noeud est alors incrémenté de 1 et lors du départ celle-ci est décrémentée de 1.

Déplacements intra niveau

Cette section du modèle est composée du module VBA contenant l’heuristique qui permet

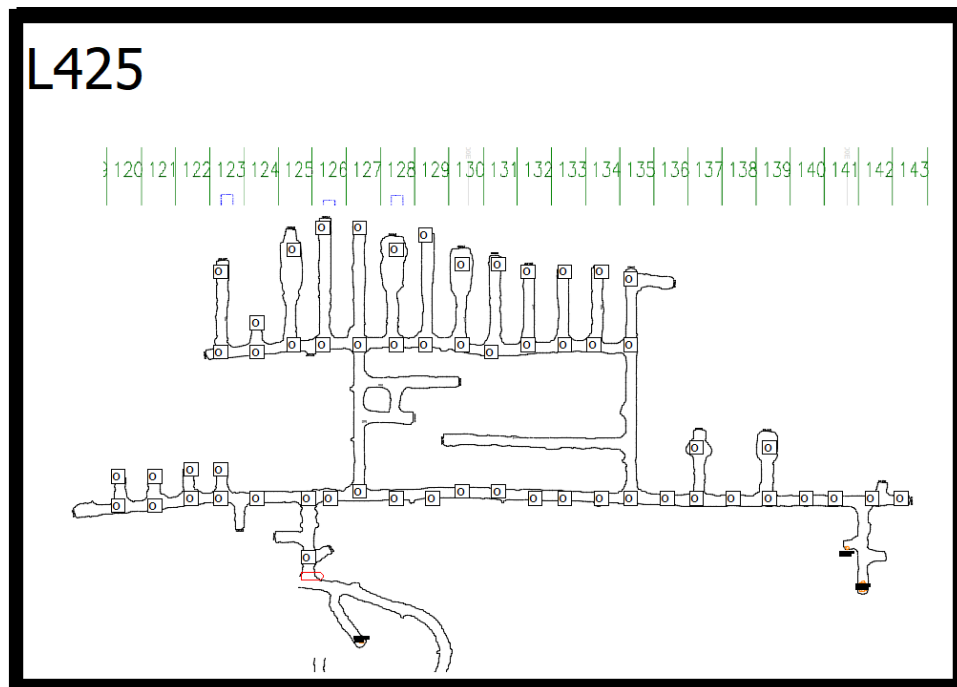


Figure 3.6 Exemple d'un niveau faisant partie du tableau de bord. Les encadrés avec des zéros sont les valeurs des variables de noeuds.

d'établir le chemin optimal à prendre en spécifiant quelles seront les stations cibles à atteindre pour se rendre à destination. Lorsqu'une station cible est donnée à l'entité, celle-ci passe à travers une logique itérative qui évalue et modifie les variables de transport du niveau pour se déplacer vers la cible sans collisions. D'ailleurs, tous les délais provoqués par l'évitement des face-à-face sont recueillis pendant la simulation. De plus, cette section du simulateur comporte les logiques de déplacements horizontaux et verticaux. Ici les déplacements horizontaux se font dans les *footwalls* et les déplacements verticaux dans les *crosscuts* et les *drawpoints*.

Déplacements inter niveau

Une fois le véhicule positionné dans une des rampes, celui-ci entre dans la section des logiques de transport associée à la rampe correspondante. Dans cette section, aucune heuristique n'est nécessaire puisqu'il existe qu'un seul chemin pour passer d'un niveau vers un autre. Effectivement, les actions possibles sont de monter, descendre ou laisser passer. En fonction du niveau de la destination ($A_Position_Level$) et de sa position dans la rampe ($A_Position_RP1$ ou $A_Position_RP2$), le véhicule sera amené à descendre ou monter la rampe. Tout comme pour les déplacements intra niveau, l'entité passe à travers une logique itérative de rampe qui évalue et modifie les variables de transport de la rampe pour se déplacer vers le niveau. De plus, si une autre entité dans le sens inverse a déjà réservé le segment pour passer, l'action

posée sera de se diriger dans un point de soutirage ou l'entrée d'un niveau pour libérer la voie. Encore une fois, les délais provoqués par l'évitement sont recueillis par le modèle.

Priorité de passage

Les camions pleins ont la priorité sur tous les autres véhicules de production. Un attribut de priorité (*A_Priority_Road*) est alors donné aux entités en début de simulation en fonction du type de véhicule dont il s'agit. En plus, cet attribut est modifié en cours de simulation pour les camions qui se remplissent et se vident de roche. Lors des déplacements intra niveau horizontaux et des déplacements inter niveau, les véhicules ayant l'attribut de priorité égal à 1 vont réserver deux segments de routes à l'avance au lieu d'un seul. Les entités ayant l'attribut égal à 0 réservent qu'un seul segment à l'avance. La réservation d'un segment est l'incréméntation de la variable de transport qui y est associé. Ainsi, lorsqu'un camion plein se déplace, celui-ci réserve plusieurs segments à l'avance et les véhicules en sens inverse vont le laisser passer étant donné que les segments à prendre sont déjà réservés à l'avance par le camion plein.

Le processus d'extraction

Dans cette partie du modèle, les véhicules qui sont arrivés à leurs stations de travail commencent à effectuer les tâches qui leur ont été attribuées à la section d'indexation des tâches. Deux types de logiques sont utilisées afin de simuler les opérations de chargement de roche et les opérations n'étant pas du chargement de roche. D'une part, pour les opérations de remplissage, la chargeuse va passer au travers de la logique et remplir un camion. Lorsque le camion est rempli, celui-ci quitte pour se rendre à la surface au portail 1 ou 2 et le temps de tâche de la chargeuse est décréémté d'une certaine valeur. Après, l'entité de la chargeuse va réitérer sur la logique pour remplir un autre camion et ainsi de suite jusqu'à ce que le temps de sa tâche soit écoulé. D'ailleurs, si aucun camion n'est présent, la chargeuse attend et vice versa. D'autre part, les activités qui ne nécessitent pas une coordination avec des camions sont modélisées plus simplement. Une fois arrivé sur place, le véhicule exécute sa tâche pendant les heures d'opérations et passera à la prochaine tâche sans avoir besoin d'attendre un camion. De plus, ce dernier type de logique modélise des activités qui s'étalent sur plusieurs heures selon la tâche. Il devient alors nécessaire de segmenter les tâches et itérer plusieurs fois sur la logique pour empêcher les opérations de continuer pendant des changements de quart de travail. Les diagrammes de flux du processus d'extraction sont présentés aux figures A.17 à A.20.

Les files d'attente lors des changements de quart

Comme il a été mentionné à la fin de la section du processus d'extraction, il est important de s'assurer qu'aucune action ne se produise pendant les changements de quart. De nombreuses vérifications de la valeur de la variable *V_Shift_Changes* sont faites au travers des sections du modèle. Lorsque la variable est activée, les entités sont placées dans des files d'attente positionnées dans cette section pour simuler l'inactivité des opérations. Une fois la variable désactivée, les entités sont affectées à leur position respective dans le modèle.

3.1.4 La validation

Une fois le modèle créé, il est nécessaire de valider celui-ci en le comparant avec des résultats réels provenant de la mine. Dans le cas présenté, la validation a été faite à l'aide de données de production hebdomadaires du département de la mine. Plus précisément, quatre semaines d'affectations ont été récupérées avec les quantités de roches extraites, le nombre de dynamitages et le temps de cycle camion obtenus en réalité. Ces affectations ont ensuite été simulées afin de comparer ces indicateurs de performance provenant du réel et ceux provenant du simulé. Pour chaque semaine simulée, dix répliques ont été faites afin que les résultats comportent la variabilité présente dans plusieurs parties du processus d'extraction. Certains ajustements ont été apportés à des paramètres du modèle après avoir discuté avec des experts du département de la mine. Aussi, certains écarts entre le réel et les résultats de simulation sont expliqués du fait qu'il est impossible que le modèle puisse capter toute la variabilité possible des événements se produisant sous terre. Sans oublier que le modèle possède des limites comme il a été présenté à la sous-section 3.1.3. Par exemple, dans la réalité, des délais quant à l'arrivée des services d'eau et d'électricité à une station peuvent causer un retard aux opérations de cette station. Dans la simulation, ces services ne sont pas simulés et il est alors impossible que les opérations soient retardées pour cette raison.

3.1.5 Les données de sortie

Le modèle possède deux types d'extrants. Tout d'abord, il y a le fichier CSV qui se met à jour tout au long de la simulation avec des valeurs de variables à une fréquence de 30 secondes. Il y a les variables de transport et de voies d'évitement des rampes, le nombre de stations en phase de soutirage par niveau, la variable de changement de quart, la quantité de roches extraite, les temps d'attente sur chaque rampe et le numéro de la réplique. Par la suite, ce fichier sera analysé afin d'obtenir les temps de trafic moyens en fonction du nombre de chargeuses actives. De même, comme dernier type d'extrant, il y a le rapport Arena à la fin

de la simulation qui dévoile les résultats finaux de la simulation tels que le temps de cycle des camions et le nombre de tonnes extraites. Il s'agit ici des indicateurs de performance les plus importants pour l'administration de la mine. La validation du modèle s'appuie sur les résultats de ce type d'extrait afin de comparer la performance du modèle à celle de la mine réelle. Ainsi, avec ces deux types d'extraits, il devient possible de valider le simulateur et aussi d'obtenir un vaste éventail de données de trafic récoltées tout au long des simulations.

3.2 L'outil de planification à court terme

La présente section décrit le contenu de l'outil de planification créé afin de produire des séquençages de tâches optimisés pour la flotte minière. Afin d'établir un tel outil, il est important de connaître les cibles de production à atteindre. D'ailleurs, le département de la mine reçoit les cibles de production et de développement hebdomadaires du service de l'ingénierie. En fonction de ces besoins et de la machinerie disponible, le département des opérations planifie le séquençage des tâches de la flotte selon l'avis de ses experts. Cependant, ces tâches de séquençage consomment beaucoup de temps et il devient impératif d'utiliser des méthodes d'optimisation telle que la programmation par contraintes pour alléger ce travail, mais aussi pour considérer toutes les options possibles (voir [18] et [5]). L'intérêt est alors de créer un outil de planification utilisant la programmation par contraintes afin de trouver des solutions optimales au problème de séquençage de tâches. Cependant, les outils présentés dans la littérature ne considèrent pas les délais générés par le trafic souterrain dans la phase d'optimisation. Ainsi, un modèle de planification à court terme de base sans trafic a été créé. Les planifications générées par ce modèle seront utilisées comme données d'entrée pour l'outil de simulation. Les données de sortie du modèle de simulation contiennent alors l'information quant au trafic qui s'est produit lors des simulations. Cet ensemble de données est finalement analysé pour récolter les moyennes de temps perdu à attendre dans le trafic en fonction du nombre de chargeuses en opération. Ces moyennes sont alors utilisées afin de former des contraintes de trafic dans le programme par contraintes. Ce nouveau modèle permet alors à l'outil de planification d'optimiser tout en considérant les délais de trafic générés par le niveau d'activité de la mine. Les résultats du nouveau modèle sont finalement comparés à ceux de l'ancien au chapitre 4. Les sous-sections qui suivent présentent les composantes clés à l'outil soit les données d'entrée, le modèle de programmation par contraintes, les données de sortie et l'ajout des contraintes de trafic.

Les limites de l'outil

Afin d'arriver à produire des planifications qui soient réalistes, résolubles et utilisables par le simulateur, il a été nécessaire d'apporter certaines limites et simplifications.

1. seules les activités de développement et de production assurées par les chargeuses, camions, boulonneuses, *jumbo* et camions à émulsion sont planifiées ;
2. les activités de chargement de roche ignorent les mouvements de va-et-vient des camions avec la surface, qui sont normalement requis pour exécuter ce genre d'activité ;
3. la matrice des distances entre les stations est créée à l'aide des stations d'entrée des niveaux, la distance moyenne entre deux *footwalls* (60 m), la distance moyenne entre deux *drawpoints* (19 m) et la distance moyenne entre deux niveaux (200 m) ;
4. tous les véhicules ont une efficacité de 100% sauf pour les chargeuses de 8 verges qui ont une efficacité de 75%. Cette distinction permet d'assurer l'attribution des tâches de chargement de roche aux chargeuses de plus grande capacité. De plus, il n'y a pas de distinction entre l'efficacité d'un camion de 30 tonnes et celle d'un camion de 50 tonnes, car les temps de tâche de chargement comprennent une flotte non homogène de camions ;
5. chaque véhicule possède une vitesse de déplacement constante de 4.2 km/h et nécessite un total de 10 minutes pour préparer et nettoyer une station de travail ;
6. lorsqu'une tâche doit continuer suite à un changement de quart, le même véhicule est affecté à la tâche et ce sera la première tâche qu'il effectuera dans le prochain quart ;
7. la flotte reste identique à travers tous les scénarios.

3.2.1 Les données d'entrée

Les données de la flotte, des tâches et une matrice des distances entre les stations sont fournies sous la forme de fichiers Excel. En premier lieu, les données de la flotte comportent le nom, le type, la position initiale et l'efficacité de chaque véhicule. Par exemple, si la chargeuse 3 a une efficacité de 0.7, la tâche A qui dure normalement 5 minutes va plutôt être de $5/0.7$, ce qui donne 7.14 minutes. Le fichier de la flotte reste identique, peu importe les scénarios puisqu'il s'agit d'une des limitations de l'étude. Pour le reste de la présente section, le terme *véhicule* sera remplacé par *équipement* afin de reprendre les termes utilisés dans la littérature.

En deuxième lieu, le fichier des tâches est créé à partir du fichier des besoins hebdomadaires de l'ingénierie choisi. Chaque tâche est identifiée par un nom, un type d'équipement, une station,

une durée, un prédécesseur, une priorité et un type qui distingue les tâches de développement à celles de production. Le nom est composé en fonction de l'action à effectuer, la station, la valeur de dynamitage (*blast*) et le numéro du fragment de la tâche. De fait, il est nécessaire de fragmenter toutes les tâches, car celles qui commencent en fin de quart doivent être arrêtées afin de simuler l'arrêt de la production lors des changements de quart. Ces tâches interrompues sont continuées au début du quart suivant. De même, la taille des fragments est définie par l'hyperparamètre T_SIZE . Par exemple, si $T_SIZE = 2$, une tâche de durée 10 sera divisée en 5 fragments. Le premier fragment sera le prédécesseur du second fragment et ainsi de suite jusqu'au dernier fragment. Par exemple, la tâche *Bolting_DP1-350-159_0_2* représente le fragment #2 de l'action de boulonnage à la station DP1-350-159 au *blast* 0. Son prédécesseur est la tâche *Bolting_DP1-350-159_0_1*. Pour le reste du chapitre, le terme *tâche* signifiera un *fragment de tâche* afin d'alléger le texte.

En dernier lieu, les données de distances entre les stations sont générées automatiquement en fonction des stations présentes dans le fichier des besoins de l'ingénierie pour la semaine. Comme il a été mentionné dans les limitations de l'outil, il est possible de déterminer la distance entre deux stations avec les paramètres de distance entre deux FW ($dist_ftw = 60$ m), deux DP ($dist_dp = 19$ m) et deux niveaux ($dist_lvl = 8$ m). Ce coefficient de distance verticale de 8 m provient du fait qu'il y a environ 200 m de chemin à parcourir dans la rampe pour passer d'un niveau subjacent à un autre. Ainsi, entre le niveau 400 et 425, la distance est de $25 \times 8 = 200$ m. Par exemple, pour calculer la distance entre DP1-400-150 et DP2-425-151, la distance entre chaque station et sa station d'entrée de niveau respective est calculée. Pour DP1-400-150, la distance d_1 avec sa station d'entrée de niveau DP1-400-125 est de :

$$d_1 = dist_dp \times |150 - 125| + dist_ftw \times |1 - 1| = 475\text{ m}$$

Ensuite, la distance d_2 entre la station DP2-425-151 et sa station d'entrée DP1-425-125 est de :

$$d_2 = dist_dp \times |151 - 125| + dist_ftw \times |2 - 1| = 554\text{ m}$$

La distance totale entre ces deux stations utilise ainsi d_1 , d_2 et la distance entre les deux niveaux :

$$d_1 + d_2 + dist_lvl \times |400 - 425| = 1229\text{ m}$$

3.2.2 Le modèle de programmation par contraintes

Le problème de planification est résolu à l'aide du CP ILOG optimizer développé par IBM [19]. Ce logiciel est spécialement conçu pour résoudre des problèmes de planification à l'aide de la

programmation par contraintes. L'API docplex sur Python a été utilisé pour créer l'outil de planification présenté ici. Ensuite, l'unité de base du modèle est en dizaine de minutes afin de simplifier les valeurs de temps au plus petit entier possible. Ceci réduira la complexité du problème à résoudre puisqu'il y aura moins de fragments de tâche à planifier. Par exemple, une tâche de 30 minutes est représenté par 3 fragments de tâche lorsque l'unité de base est la dizaine de minute tandis que si l'unité de base était en minute, la tâche aurait 30 fragments de tâche. Il est à ajouter que le modèle de base sans contraintes de trafic est nommé CP-M0 dans le chapitre 4.

Cinq indices sont employés dans le modèle : t pour la tâche, o pour l'équipement, l pour l'emplacement, i pour la priorité et e pour le type d'équipement. Étant donné qu'une tâche de chargement de roche nécessite plus d'un type de véhicule (camion et chargeuse) en même temps, l'indice e est ajouté afin de créer une variable d'affectation pour chaque type d'équipement. Finalement, les parties suivantes sont nécessaires pour résoudre le problème d'optimisation : les ensembles, les paramètres, les variables, la fonction objectif, les contraintes et le solveur.

Les ensembles

Cinq groupes d'ensembles, présentés au tableau 3.4, sont utilisés dans le modèle : $\mathcal{T}$, $\mathcal{E}$, $\mathcal{O}$, $\mathcal{L}$ et $\mathcal{V}$.

Tableau 3.4 Descriptions des ensembles.

Ensemble	Description
$\mathcal{T}$	Tâches à effectuer durant l'horizon de planification provenant du fichier de l'ingénierie.
$\mathcal{E}_t$	Types d'équipements nécessaires pour effectuer la tâche t .
$\mathcal{E}_t^P$	Types d'équipements nécessaires au prédécesseur de la tâche t .
$\mathcal{O}$	Équipements provenant du fichier de la flotte.
$\mathcal{O}_e$	Équipements de type e .
$\mathcal{L}$	Emplacements dans la mine où des tâches doivent être exécutées.
$\mathcal{V}^{Te}$	Ensemble des variables d'intervalle des tâches (la notion de variable d'intervalle sera définie plus loin).
$\mathcal{V}^{Teo}$	Ensemble des variables d'intervalle d'affectation des tâches aux équipements.
$\mathcal{V}_i^{Te}$	Ensemble des variables d'intervalle des tâches de priorité i .

Les paramètres

Les paramètres utilisés par le programme de programmation par contraintes sont regroupés en quatre catégories. Les tâches, les équipements, les emplacements et les hyperparamètres. Le descriptif de chacun des paramètres est présenté au tableau 3.5.

Les variables

Trois groupes de variables sont utilisés pour le modèle de base sans contraintes de trafic. En premier lieu, il y a les variables d'intervalle (*interval variable*) qui possèdent chacune une durée, un temps de début et un temps de fin, ce qui est primordial pour représenter les tâches à affecter aux équipements (véhicules). Ce type de variable peut être optionnel, ce qui est utile afin de ne pas retrouver certaines variables d'intervalle dans la solution finale. Par exemple, si la tâche A n'est pas affectée à l'équipement B, la variable d'intervalle représentant cette affectation sera absente. Il est à mentionner que le terme *variable d'intervalle* sera remplacé par *variable* pour le reste du chapitre afin d'alléger le texte. En deuxième lieu, il y a les variables de séquence (*sequence variable*) qui permettent de séquencer des variables d'intervalle en fonction de contraintes. Par exemple, la chargeuse #3 possède une variable de séquence qui empêche le chevauchement des tâches qui lui sont affectées. En dernier lieu, des variables de fonction d'état sont nécessaires pour faire le suivi des affectations des tâches aux équipements. Ce type de variable est présenté en détail à la sous-section des contraintes au niveau des contraintes de distance de transition. Le tableau 3.6 présente toutes les variables utilisées par le modèle.

Une variable v_{te} oblige la présence de la tâche t dans la solution, nécessitant le type d'équipement e , dans le modèle. Cependant, il existe plusieurs équipements d'un même type e qui pourraient être affectés à la tâche t , ce qui nécessite la présence de variables optionnelles v_{teo} . Par exemple, avec une flotte de deux boulonneuses, la tâche de boulonnage A (v_{Ab}) pourrait être desservie par la boulonneuse #1 ou la boulonneuse #2. Ainsi il existera deux variables v_{teo} en lien avec cette tâche, soit v_{Ab1} et v_{Ab2} . Une seule variable v_{teo} peut être active pour une tâche v_{te} sauf pour les tâches nécessitant des camions. Des contraintes alternatives (*alternative constraints*) sont présentées à la sous-section des contraintes afin de créer ce lien entre les variables v_{te} et v_{teo} .

Les variables de séquence $eseq_o$ et $lseq_l$ sont utilisées avec des contraintes de non chevauchement (*non-overlapping constraints*) afin d'éviter le chevauchement des tâches attribuées à l'équipement o et l'emplacement l respectivement. Les variables de séquence $eseq_o$ seront aussi utilisées pour les contraintes de précedence des fragments de tâches.

Tableau 3.5 Descriptions des paramètres.

Paramètre	Description
ETP_t	Type d'équipement nécessaire pour exécuter la tâche t . Ce paramètre prend deux valeurs différentes pour les activités de chargement de roche qui nécessitent des camions et des chargeuses LHD.
LOC_t	Emplacement de la tâche t .
$TIME_t$	Durée de la tâche t .
PRE_t	Tâche prédécesseur de la tâche t .
$CMPL_t$	Complétion de la tâche t . Ce paramètre est à zéro si la tâche n'a pas déjà été commencée avant la planification.
C_TIME_t	Temps restant pour accomplir la tâche t . Ce paramètre équivaut à $TIME_t$ si la tâche n'a pas commencée avant la planification.
$PRIO_t$	Priorité de la tâche t entre 1 et 6. Plus la valeur est élevée, moins la tâche est prioritaire.
AT_t	Type d'activité de la tâche t . Ce paramètre permet de distinguer les tâches de développement des tâches de production.
TP_o	Type de l'équipement o .
L_o	Emplacement initial de l'équipement o .
RTM_o	Taux d'efficacité de l'équipement o . La valeur peut varier de 0 à 1.
TM_o	Matrice de temps de transition de l'équipement o entre les emplacements.
$DIST_l$	Ensemble des distances entre l'emplacement l et tous les autres emplacements $\in \mathcal{L}$.
IDX_l	Indice unique associé à l'emplacement l pour le distinguer des autres emplacements $\in \mathcal{L}$.
QTD	Nombre de camions nécessaire lors d'une tâche de chargement de roche pour le développement. Ce nombre de camions est fixe pour toutes les tâches de chargement en développement.
QTP	Nombre de camions nécessaire lors d'une tâche de chargement de roche pour la production. Ce nombre de camions est fixe pour toutes les tâches de chargement en production.
SRD	Délai d'inactivité en début de chaque quart.
SL	Durée d'un quart de travail.
T_SIZE	Taille des fragments des tâches. Cette taille de fragment est identique pour toutes les tâches.

Tableau 3.6 Descriptions des variables.

Variable	Description
v_{te}	Variable d'intervalle de la tâche t nécessitant le type d'équipement e .
v_{teo}	Variable d'intervalle optionnelle d'affectation de la tâche t , nécessitant le type d'équipement e , à l'équipement o .
$strt_o$	Variable d'intervalle de départ de l'équipement o . Cette variable de durée 0 force l'équipement à commencer à son emplacement initial au début de sa séquence de tâches $eseq_o$.
f_o	Variable de fonction d'état de l'équipement o .
$eseq_o$	Variable de séquence qui comporte toutes les tâches v_{teo} associées à l'équipement o .
$lseq_l$	Variable de séquence qui comporte toutes les tâches v_{te} associées à l'emplacement l .

La fonction objectif

L'objectif du modèle est de réduire la durée totale de la planification tout en affectant les tâches en ordre de priorité. Plus précisément, la fonction objectif (`minimizeStaticLex`) cherche à minimiser en ordre lexicographique le temps de fin (`endOf`) maximum parmi tous les temps de fin des tâches d'une même priorité. Ainsi, chacune des six classes de priorité possèdera un temps de fin maximal dans une solution. Une nouvelle solution réalisable est admissible si elle améliore le temps de fin d'au moins une classe sans détériorer les classes de plus grande priorité. Par exemple, si la solution optimale actuelle est de [100, 200, 300, 400, 500, 600] et qu'une nouvelle solution trouvée est de [100, 300, 100, 400, 500, 600], celle-ci ne sera pas admissible, puisque cette nouvelle solution améliore le temps de fin maximal de la classe de priorité 3, mais détériore le temps de fin maximal de la classe de priorité 2, qui est plus prioritaire.

$$\text{minimizeStaticLex}\left(\max(\text{endOf}(\mathcal{V}_1^{Te}), \dots, \max(\text{endOf}(\mathcal{V}_6^{Te}))\right) \quad (3.1)$$

Les contraintes alternatives

Une contrainte alternative `alternative( $i, J, k$ )` est utilisée afin d'obliger la présence de k tâches parmi l'ensemble J de tâches lorsqu'une tâche i est présente. Les k tâches choisies doivent commencer et terminer au même moment que la tâche i . D'ailleurs, la contrainte 3.2 permet alors l'affectation d'un seul équipement (v_{teo}) à la tâche (v_{te}), qui n'est pas de type camion. Toutefois, les tâches avec l'équipement de type camion nécessitent plusieurs camions à la fois, donc les contraintes 3.3 et 3.4 sont utilisées afin de forcer l'affectation de plusieurs v_{teo} différents à la même tâche v_{te} .

$$\text{alternative}(v_{te}, \mathcal{V}^{Teo}, 1) \quad \forall t \in \mathcal{T}, e \in \mathcal{E}_t \setminus \{Truck\} \quad (3.2)$$

$$\text{alternative}(v_{te}, \mathcal{V}^{Teo}, QTD) \quad \forall t \in \mathcal{T} \mid \{AT_t = Dev\}, e = Truck \quad (3.3)$$

$$\text{alternative}(v_{te}, \mathcal{V}^{Teo}, QTP) \quad \forall t \in \mathcal{T} \mid \{AT_t = Prod\}, e = Truck \quad (3.4)$$

Les contraintes de non-chevauchement

Les contraintes 3.5 et 3.6 empêchent le chevauchement des tâches affectées aux équipements et emplacements respectivement. Un équipement ne peut effectuer qu'une seule tâche à la fois

et un délai de transition doit être considéré entre deux emplacements différents. La contrainte 3.5 prend alors en deuxième paramètre la matrice de transition de l'équipement. De plus, une seule tâche peut être effectuée à la fois à chaque emplacement.

$$\text{noOverlap}(eseq_o, TM_o) \quad \forall o \in \mathcal{O} \quad (3.5)$$

$$\text{noOverlap}(lseq_l) \quad \forall l \in \mathcal{L} \quad (3.6)$$

Les contraintes de synchronisation

Certaines tâches nécessitent plusieurs types d'équipements au même moment. La contrainte 3.7 oblige ainsi toutes ces tâches complémentaires à commencer et terminer au même moment.

$$\text{synchronize}(v_{te}, \mathcal{V}^{Te}) \quad \forall t \in \mathcal{T}, e \in \mathcal{E}_t \mid \{|\mathcal{E}_t| > 1\} \quad (3.7)$$

Les contraintes de précédence

Chaque tâche qui possède un prédécesseur doit attendre que ce prédécesseur soit terminé avant de pouvoir débiter. La contrainte 3.8 assure ainsi que le début de la tâche soit après la fin du prédécesseur.

$$\text{endBeforeStart}(v_{t'e'}, v_{te}) \quad \forall t \in \mathcal{T}, t' = PRE_t, e \in \mathcal{E}_t, e' \in \mathcal{E}_t^P \quad (3.8)$$

Par la suite, en revenant au principe des fragments de tâches, il faut que chaque fragment d'une tâche affectée à un équipement soit présent dès qu'un seul des fragments est présent. Il est alors possible d'utiliser la fonction **ifThen** avec la fonction **presenceOf** pour créer la contrainte 3.9. Lorsque le fragment prédécesseur $v_{t'e'o}$ est présent, le fragment v_{teo} sera aussi présent. Tous les fragments d'une même tâche sont ainsi présents et affectés au même équipement. Il serait irréaliste d'affecter plusieurs fragments d'une même tâche à plusieurs équipements différents si la tâche nécessite qu'un seul équipement.

$$\begin{aligned} &\text{ifThen}(\text{presenceOf}(v_{t'e'o}), \text{presenceOf}(v_{teo})) \\ &\forall o \in \mathcal{O}, t \in \mathcal{T} \mid \{ETP_t = TP_o\}, t' = PRE_t, e \in \mathcal{E}_t, e' = e \end{aligned} \quad (3.9)$$

Également, chaque fragment d'une tâche doit être en ordre de précédence et être affecté au même équipement. Ainsi, les fragments d'une même tâche sont regroupés dans la séquence de l'équipement. La contrainte 3.10 utilise la fonction `previous(seq, i, j)` qui prend en paramètre une séquence de variables d'intervalle seq , la variable j et la variable i qui doit être placée juste avant j dans la séquence. Cette contrainte permet aussi d'assurer la continuité des fragments d'une même tâche malgré les changements de quart. Par exemple, soit une tâche A possédant 4 fragments qui débute juste avant un changement de quart. Si, les 2 premiers fragments ont été effectués avant le changement de quart, l'équipement débutera au prochain quart avec les 2 derniers fragments.

$$\begin{aligned} & \text{previous}(eseq_o, v_{t'e'o}, v_{teo}) \\ & \forall o \in \mathcal{O}, t \in \mathcal{T} \mid \{ETP_t = TP_o\}, t' = PRE_t, e \in \mathcal{E}_t, e' = e \end{aligned} \quad (3.10)$$

Les contraintes de changement de quart

Les explosions (*blasts*) sont planifiées en fin de quart de travail après avoir évacué tout le personnel. Une période d'une dizaine de minutes est allouée à cet effet entre deux quarts de travail. Également, à la fin de chaque quart de travail se trouve un délai d'inactivité (*SRD*) en lien avec le changement d'équipe et l'évacuation des gaz d'explosifs. Par la suite, il faut contraindre les tâches d'explosion à être présentes que dans ces périodes de changement de quart et aussi contraindre l'absence de production selon le délai *SRD*. L'idée est alors d'utiliser les fonctions par étape (*step functions*) 3.11 et 3.12 pour définir les périodes de changements de quart. Ce genre de fonction est l'équivalent d'un calendrier auquel des périodes de changements de quart sont fixées. Maintenant, la fonction 3.11 pour les explosions prend la valeur de 1 lors des changements de quart et 0 sinon. La fonction 3.12 pour les opérations prend la valeur de 0 lors des changements de quart et 1 sinon.

$$\begin{aligned} \text{stepFunction } BC = \text{stepwise} \{ & 0 \rightarrow SL; 1 \rightarrow SL + 1; \dots; \\ & 0 \rightarrow n(SL) + n - 1; 1 \rightarrow n(SL) + n \} \end{aligned} \quad (3.11)$$

$$\begin{aligned} \text{stepFunction } OC = \text{stepwise} \{ & 1 \rightarrow SL; 0 \rightarrow SL + 1 + SRD; \dots; \\ & 1 \rightarrow n(SL) + n - 1; 0 \rightarrow n(SL) + n + SRD \} \end{aligned} \quad (3.12)$$

Par la suite, il est possible d'établir des contraintes d'interdiction d'étendue `forbidExtent(i, F)` à l'aide des fonctions par étapes 3.11 et 3.12. Ce type de contrainte interdit la présence d'une

variable i dans les sections de valeur 0 de la fonction F . Les tâches d'explosion et de production sont alors contraintes par 3.13 et 3.14 respectivement.

$$\text{forbidExtent}(v_{te}, BC) \quad \forall t \in \mathcal{T} \mid \{Blast \in t\}, e = None \quad (3.13)$$

$$\text{forbidExtent}(v_{te}, OC) \quad \forall t \in \mathcal{T} \mid \{Blast \notin t\}, e \in \mathcal{E}_t \quad (3.14)$$

Les contraintes de distance de transition

Des variables de fonction d'état (*state function variable*) f_o sont nécessaires afin d'observer l'affectation des tâches aux équipements au fil du temps. Grâce à ces variables, il est possible d'obtenir un historique des valeurs qu'elles prennent. Ces variables sont alors très utiles afin d'émettre le séquençage des tâches de chaque équipement en données de sortie par l'outil de planification. Les contraintes qui s'y rattachent permettent alors de mieux caractériser la solution. Dans le travail présenté ici, la valeur de chaque état sera l'index de la position de la tâche. La contrainte **alwaysEqual**(f, i, val) permet de fixer la valeur de la variable de fonction d'état f à val sur l'intervalle de temps i . Au début de la planification, soit l'intervalle de temps $(-1, 0)$, chaque équipement commence à sa position initiale. La variable f_o est alors contrainte à prendre la valeur d'index de la position initiale sur l'intervalle $(-1, 0)$ à l'aide de la contrainte 3.15. Ensuite, il est nécessaire d'ajouter une contrainte pour chaque tâche affectée à un équipement afin que la variable f_o garde en mémoire ladite affectation. Enfin, la contrainte 3.16 est établie à cet effet. La variable f_o prendra l'index de l'emplacement de la tâche sur l'intervalle de temps correspondant à celui de v_{teo} . Il est à mentionner que si une tâche n'est pas affectée à un certain équipement, la variable v_{teo} de cette affectation est alors absente et la variable f_o ne sera pas modifiée.

$$\text{alwaysEqual}(f_o, (-1, 0), IDX_l) \quad \forall o \in \mathcal{O}, l = L_o \quad (3.15)$$

$$\text{alwaysEqual}(f_o, v_{teo}, IDX_l) \quad \forall o \in \mathcal{O}, \forall t \in \mathcal{T} \mid \{TP_o \in ETP_t\}, l = LOC_t \quad (3.16)$$

Le solveur

Le solveur utilise deux paramètres qui sont le temps de recherche (*time limit*) et l'optimalité relative tolérée (*relative optimality tolerance*). Le temps de recherche est le temps alloué au solveur afin de trouver la meilleure solution, alors que le paramètre d'optimalité relative tolérée est utilisé pour comparer les solutions entrent-elles à partir de leur valeur de fonction

objectif. Lorsque le solveur s'arrête sur une solution, cela exprime alors qu'il n'y a aucune autre solution trouvée qui améliore l'objectif d'au moins la valeur de l'objectif multiplié par ce paramètre. Le CP Ilog utilise la propagation de contraintes dans la phase de recherche de solutions. Cette stratégie est un élément clé dans les outils de programmation par contraintes afin de réduire le domaine de nombreuses variables de décisions. Conjointement à la propagation de contraintes, le CP Ilog utilise des stratégies de recherche constructives afin d'explorer l'arbre de recherche et trouver une solution réalisable au problème.

3.2.3 Les données de sortie

Lorsque le solveur trouve une solution optimale ou atteint la limite de temps avec au moins une solution, les variables de fonction d'état de la solution sont utilisées pour reconstituer les affectations des tâches. En premier lieu, grâce à ces variables, un fichier Excel de donnée de sortie est créé. Celui-ci possède trois sections qui sont le sommaire des tâches, le sommaire des ressources et le graphique des ressources. Notamment, le sommaire des tâches contient les tâches avec leur temps de début, leur temps de fin, leur durée et la ressource (équipement) affectée. La deuxième section, soit le sommaire des ressources, comporte les équipements avec leur temps d'activité, leur temps de déplacement, leur temps d'inactivité et leur séquence de travail. Par exemple, une partie de la séquence de la boulonneuse # 3 est $(DP1-150-162, 1)-(4)-(DP2-400-154, 30)$. La première parenthèse représente l'emplacement initial de l'équipement. Après un délai de 4 dizaines de minutes, la boulonneuse commence sa tâche située à DP2-400-154 d'une durée de 30 dizaines de minutes.

La dernière section du fichier est un graphique d'utilisation des ressources en fonction du temps. Un exemple est montré à la figure 3.7. D'ailleurs, les intervalles de temps sans équipement actif représentent les changements de quart. Ce type de graphique est seulement utilisé afin d'observer l'utilisation des équipements en fonction du temps. Chaque couleur représente un véhicule. Par exemple, on remarque qu'au temps 430 seul le véhicule 37 est actif.

En deuxième lieu, la solution optimale est transmise en donnée d'entrée à la partie séquences de tâches du fichier Excel principal présenté à la section 3.1.2. Il sera alors possible de simuler la solution optimale du programme par contraintes et d'analyser l'écart entre les résultats planifiés et les résultats simulés pour un même scénario.

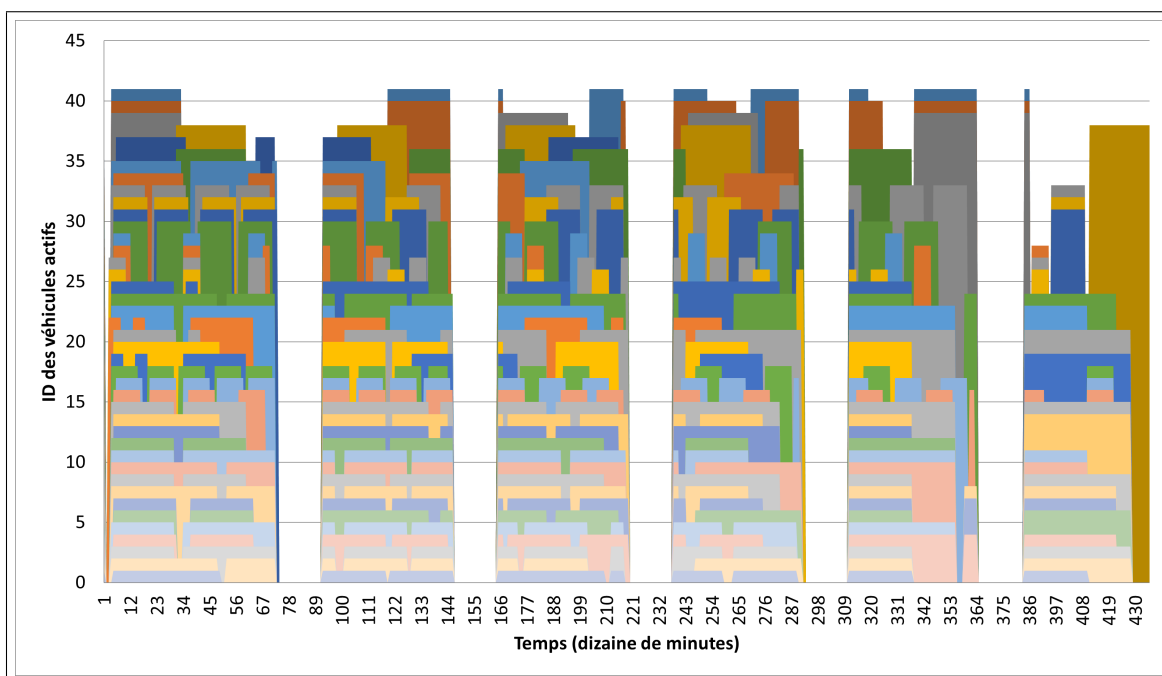


Figure 3.7 Exemple d'un graphique d'utilisation des équipements au travers du temps où chaque couleur représente un équipement.

3.2.4 L'ajout des contraintes de trafic

À l'aide du modèle de simulation, le trafic généré par les activités de production de la mine peut être modélisé. En lien avec le second sous-objectif de ce mémoire qui est de déterminer des contraintes fidélisant la congestion des routes, il est alors primordial d'analyser le trafic qui se produit lors de scénarios de production différents. Ces scénarios seront présentés en détail à la section 4. De fait, les résultats des simulations dévoilent que le trafic provient presque totalement des interactions entre les camions qui se déplacent dans les rampes. Une très faible portion du trafic se trouve dans les interactions sur les niveaux. Le nombre de chantiers ouverts en chargement de roche, soit pour du développement ou de la production, a une incidence directe sur le trafic qui se produit dans les rampes. Plus il y a de chantiers ouverts en même temps, plus le trafic est élevé. Ici, le terme interaction signifie l'action de laissez-passer un véhicule qui a la priorité de passage.

À première vue, une bonne idée serait de planifier les interactions des camions dans la rampe afin de diminuer le trafic qui s'y produit. Cependant, il est presque impossible de prévoir à quels moments se produisent ces interactions. Effectivement, beaucoup de variabilité demeure dans la vitesse des véhicules, le comportement humain, l'état des routes, le nombre de camions en marche, les problèmes mécaniques des véhicules et plusieurs autres facteurs qui influencent

la position des camions.

L'approche prise dans ce mémoire est plutôt de comptabiliser le trafic moyen à chaque minute en fonction du nombre de chargeuses LHD en opération pour une tâche de chargement de roche. La notion de trafic ici considère l'ensemble des temps perdus dans toute la mine pour tous les véhicules qui attendent afin de céder le passage. Par exemple, si en tout temps il y a toujours 3 camions en attente peu importe le nombre de chargeuses LHD en fonction pour des tâches de chargement de roche, le trafic moyen sera de 3 minutes pour chaque minute de production. En effet, chaque tâche de chargement de roche en développement ou en production nécessite plusieurs camions afin de remonter la roche à la surface. Il s'agit ici de la source expliquant le déplacement des camions. Plus il y a de chargeuses en opération, plus il y aura de camions en déplacement dans la mine et plus il y aura du trafic. Ainsi, une valeur de trafic moyen par minute est établie pour chaque nombre de chargeuses actives grâce aux résultats des simulations. Par exemple, lorsqu'il n'y a qu'une seule chargeuse active, une valeur de trafic moyen de 0.1 minute (ou 6 secondes) exprime alors que chaque minute où une seule chargeuse est en opération pour une tâche de chargement, 6 secondes de trafic sont attendues. L'avantage de cet estimateur de trafic est qu'il évite le besoin de planifier les interactions entre les camions. De plus, la simplicité de cette approche rend aisé l'ajout du trafic dans l'outil de programmation par contraintes qui ignore les mouvements de va-et-vient des camions avec la surface.

Une nouvelle variable entière cs_z qui compte le nombre de chargeuses effectuant une tâche de chargement est établie à chaque unité de temps $z \in \mathcal{Z}$, soit à chaque dizaine de minutes pour l'outil de planification. Puis, un nouvel ensemble $\mathcal{D}$ comptabilise chaque valeur de trafic moyen en fonction du nombre de chargeuses actives. À l'aide de ces nouvelles variables et de cet ensemble de temps de trafic, plusieurs contraintes de trafic peuvent être créées. Les planifications deviennent alors plus réalistes et potentiellement plus optimales puisque le trafic est dorénavant considéré par l'outil. Trois modèles de planification avec des contraintes de trafic différentes ont été créés. Le premier (CP-M1) utilise la somme de trafic global dans sa fonction objectif. Le deuxième (CP-M2) comptabilise le trafic qui se produit pendant chaque tâche de chargement en développement et ajoute un délai après ladite tâche afin de repousser le début de la prochaine tâche du cycle. Le dernier modèle (CP-M3) est similaire à CP-M2, mais inclut aussi les délais de trafic après les tâches de chargement de roche en production.

La contrainte de compte du nombre de chargeuses

Peu importe le modèle avec trafic, les variables cs_z doivent être contraintes à prendre la valeur du nombre de chargeuses actives au moment z . Donc, il faut premièrement établir un ensemble $OLAP_z$ de toutes les tâches de chargement qui ont lieu au moment z à l'aide de la contrainte $\text{overlapLength}(a, b)$ (3.17). Cette contrainte permet de connaître la longueur d'intersection entre deux variables d'intervalle a et b si elles s'intersectent. Pour ce travail, cette valeur d'intersection ne peut être que de 0 ou 1 étant donné que la contrainte est utilisée entre une tâche de chargement et une unité de temps z qui est d'une dizaine de minutes. Ensuite, la quantité de 1 présent dans l'ensemble $OLAP_z$ est comptée à l'aide de $\text{count}(l, 1)$ où l est un ensemble d'entiers (3.18).

$$OLAP_z = \left\{ \text{overlapLength}(v_{te}, z) : v_{te} \in \mathcal{V}^{Te} \wedge \text{Mucking} \in t \wedge e = \text{Scooptram} \right\} \quad \forall z \in \mathcal{Z} \quad (3.17)$$

$$cs_z = \text{count}(OLAP_z, 1) \quad \forall z \in \mathcal{Z} \quad (3.18)$$

Le modèle CP-M1

L'intuition derrière ce modèle est d'ajouter la somme des délais de trafic au cours de toutes les tâches de chargement à la fonction objectif qui minimise en ordre lexicographique. Cette nouvelle fonction objectif comporte deux éléments différents, soient les six classes de tâches et une somme de délai de trafic. Cette somme est établie en utilisant la fonction $\text{element}(l, i)$ qui donne une valeur entière (élément) parmi l'ensemble l en fonction de l'indice i . Alors, ceci permet de déterminer le délai de trafic moyen parmi l'ensemble $\mathcal{D}$ associé au nombre de chargeuses actives cs_z au moment z . De plus, il est possible de composer sept fonctions objectifs différentes selon la position de la somme de trafic dans l'ordre lexicographique. Par exemple, la fonction 3.19 avec la somme de trafic en priorité 1 cherchera d'abord à minimiser le plus possible le trafic avant de minimiser les temps de complétion des six classes de tâches. Dans ce cas-ci, un résultat attendu serait une augmentation des temps de complétion puisqu'en espaçant les tâches, le nombre de chargeuses actives au même moment est minimisé et le trafic est alors minimal.

$$\text{minimizeStaticLex}\left(\sum_{cs_z} \text{element}(\mathcal{D}, cs_z), \max(\text{endOf}(\mathcal{V}_1^{Te})), \dots, \max(\text{endOf}(\mathcal{V}_6^{Te}))\right) \quad (3.19)$$

$$\text{minimizeStaticLex}\left(\max(\text{endOf}(\mathcal{V}_1^{Te})), \dots, \max(\text{endOf}(\mathcal{V}_6^{Te})), \sum_{cs_z} \text{element}(\mathcal{D}, cs_z)\right) \quad (3.20)$$

L'équation 3.20 présente la fonction objectif lorsque la somme de trafic est en dernière position. Cette somme provient de l'addition de tous les délais de trafic associés aux nombres de chargeuses actives à chaque itération de la planification. Par exemple, pour une planification qui a un horizon de 2 dizaines de minutes avec 0 chargeuses actives pour la première itération de temps et 2 chargeuses actives pour la dernière itération, le trafic associé à 0 chargeuses est additionné au trafic associé à 2 chargeuses. Les cinq autres équations de fonction objectif avec une somme de trafic ne sont pas présentées pour alléger le texte. En bref, sept versions différentes du modèle CP-M1 ont été créées et se dénotent de la manière suivante : CP-M1P1, ..., CP-M1P7 en fonction de la position de la somme de trafic dans la fonction objectif.

Le modèle CP-M2

Contrairement au modèle CP-M1, l'idée ici est plutôt d'accumuler le trafic se produisant pendant chaque tâche de chargement de roche en développement afin d'ajouter ce délai de trafic à la fin de ladite tâche. En effet, pour les opérations en temps réel, les temps de tâches de chargement sont influencés par le trafic et les tâches qui suivent commencent plus tard que s'il n'y avait pas de trafic. L'objectif de ce modèle est alors de reproduire ce phénomène lors de la planification. Tout d'abord, une nouvelle variable entière d'ensemble $t_sum_{v_{t'e'}}$ (`integerVarList`) recueille le compte de chargeuses actives lors d'une tâche de chargement de roche en développement $v_{t'e'}$ à chaque unité de temps. Ainsi, la valeur de cette variable à l'unité de temps z est représentée par $t_sum_{v_{t'e'}z}$. La contrainte 3.21 oblige la variable à prendre la valeur du nombre de chargeuses actives cs_z si la variable $v_{t'e'}$ intersecte avec la période z . De plus, il est important de noter ici que les délais de trafic ne sont pas pris en compte pour les tâches de chargement en production. Étant donné que les tâches de chargement de roche en production ont comme successeur une tâche de dynamitage (*blast*), l'expression $Blast \notin t$ empêche la prise en compte du trafic pour ces tâches.

$$\begin{aligned}
& \text{ifThen}(\text{overlapLength}(v_{t'e'}, z) > 0, t_sum_{v_{t'e'}z} = cs_z) \\
& \forall t \in \mathcal{T} \mid \{Mucking \notin t \wedge Blast \notin t\}, t' = PRE_t \wedge Mucking \in t', e \in \mathcal{E}_t, e' \in \mathcal{E}_t^P, z \in \mathcal{Z}
\end{aligned} \tag{3.21}$$

Les délais de trafic associés aux valeurs des variables parmi l'ensemble $t_sum_{v_{t'e'}}$ sont ensuite additionnés entre eux à l'aide de 3.22 et arrondis à l'entier supérieur avec $\text{ceil}(i)$. Puis, cette somme de trafic, qui s'est produit lors de la tâche $v_{t'e'}$, est utilisée après la tâche pour retarder le départ de son successeur v_{te} dans la contrainte de précédence de ces deux tâches (voir 3.23).

$$\begin{aligned}
& \text{traffic_sum} = \text{ceil}\left(\sum_z \text{element}(\mathcal{D}, t_sum_{v_{t'e'}z})\right) \\
& \forall t \in \mathcal{T} \mid \{Mucking \notin t \wedge Blast \notin t\}, t' = PRE_t \wedge Mucking \in t', e \in \mathcal{E}_t, e' \in \mathcal{E}_t^P
\end{aligned} \tag{3.22}$$

$$\begin{aligned}
& \text{endBeforeStart}(v_{t'e'}, v_{te}, \text{traffic_sum}) \\
& \forall t \in \mathcal{T} \mid \{Mucking \notin t \wedge Blast \notin t\}, t' = PRE_t \wedge Mucking \in t', e \in \mathcal{E}_t, e' \in \mathcal{E}_t^P
\end{aligned} \tag{3.23}$$

Le modèle CP-M3

Ce modèle se distingue du modèle CP-M2 avec l'ajout des délais de trafic aux tâches de chargements en production. Pour les chantiers de production, seules les tâches de chargement de roche et de dynamitage sont planifiées par l'outil (voir 3.2). La contrainte 3.26 diffère de celle du modèle CP-M2 (3.23) par le fait qu'elle permet d'ajouter des délais aux tâches de chargement en production. Effectivement, seules les tâches de chargement en production ont une tâche successeur *Blast* lorsque comparées aux tâches de chargement en développement. En retirant la restriction concernant les tâches *Blast* de l'ensemble $\mathcal{T}$, les délais de trafic sont alors possibles pour les tâches de chargement en production. D'ailleurs, cette modification double le nombre de contraintes et augmente d'un tiers le nombre de variables pour chaque scénario. L'outil ne trouve pas de solution dans un horizon de 10 minutes de résolution. Ainsi, pour ce modèle, le paramètre de temps de résolution de modèle a été augmenté à 30 minutes afin d'obtenir des solutions réalisables. Le modèle CP-M0 aura aussi un temps de résolution augmenté à 30 minutes lorsqu'il viendra temps de comparer l'optimalité du modèle CP-M3 avec trafic vs le modèle CP-M0 sans trafic pour ne pas biaiser la comparaison.

$$\begin{aligned} & \text{ifThen}(\text{overlapLength}(v_{t'e'}, z) > 0, t_sum_{v_{t'e'}z} = cs_z) \\ & \forall t \in \mathcal{T} \mid \{Mucking \notin t\}, t' = PRE_t \wedge Mucking \in t', e \in \mathcal{E}_t, e' \in \mathcal{E}_t^P, z \in \mathcal{Z} \end{aligned} \quad (3.24)$$

$$\begin{aligned} & traffic_sum = \text{ceil}\left(\sum_z \text{element}(\mathcal{D}, t_sum_{v_{t'e'}z})\right) \\ & \forall t \in \mathcal{T} \mid \{Mucking \notin t\}, t' = PRE_t \wedge Mucking \in t', e \in \mathcal{E}_t, e' \in \mathcal{E}_t^P \end{aligned} \quad (3.25)$$

$$\begin{aligned} & \text{endBeforeStart}(v_{t'e'}, v_{te}, traffic_sum) \\ & \forall t \in \mathcal{T} \mid \{Mucking \notin t\}, t' = PRE_t \wedge Mucking \in t', e \in \mathcal{E}_t, e' \in \mathcal{E}_t^P \end{aligned} \quad (3.26)$$

Dans ce chapitre, les modèles de planification et les moyens mis en place pour valider ceux-ci ont été présentés. Au chapitre 4, nous testerons leur efficacité à produire des planifications réalistes et optimales.

CHAPITRE 4 RÉSULTATS EXPÉRIMENTAUX ET DISCUSSION GÉNÉRALE

Tous les ensembles de données concernant la flotte d'équipements mobiles, l'architecture de la mine et les tâches hebdomadaires proviennent d'une mine d'or canadienne. Un générateur de tâches hebdomadaires a pu être construit afin de générer 10 scénarios aléatoires qui permettent de tester l'efficacité des modèles. D'ailleurs, une dizaine de plans de travail hebdomadaires de la mine réelle ont été utilisés afin de paramétrer le générateur, ce qui nous permet d'obtenir des scénarios réalistes. Comme il a été présenté au chapitre 3, les tâches de chaque scénario sont utilisées en intrant pour le modèle de planification à court terme qui affecte les tâches aux équipements mobiles dans un ordre optimal. L'ensemble de la mine avec ses 14 niveaux et jusqu'à 610 emplacements de travail par niveau peuvent être utilisés dans nos modèles. Ce nombre d'emplacements est plus élevé que le nombre d'emplacements disponibles en réalité afin que nos outils prennent en compte l'agrandissement des niveaux sans avoir à faire d'importantes mises à jour. De plus, la flotte d'équipements mobiles est composée de 15 camions de 50 tonnes, 4 chargeuses LHD de 8 verges, 7 chargeuses LHD de 11 verges, 3 camions à émulsion, 4 jumbos et 8 boulonneuses.

Les planifications ont été obtenues grâce à notre modèle de programmation par contraintes construit avec l'extension Python DOcplex du ILOG CPLEX Optimization Studio V12.10.0 d'IBM. La phase d'optimisation utilise huit *threads* pour du calcul en parallèle sur un ordinateur Intel(R) Core(TM) i7-9700 CPU @ 3.00GHz 3.00 GHz avec 16 GB de RAM. Le temps de recherche de l'outil est fixé à 600 secondes et l'optimalité relative tolérée à 5% pour les modèles CP-M0 (10 minutes de résolution), CP-M1 et CP-M2. Le modèle de base sans trafic CP-M0 et CP-M3 ont un temps de recherche réglé à 1800 secondes. Au total, 10 scénarios ont été construits pour tester les trois modèles d'inclusion des contraintes de trafic qui seront comparés au modèle CP-M0 sans trafic. Un générateur de scénarios paramétré à l'aide de plans hebdomadaires de la mine a été conçu afin d'obtenir des ensembles de tâches hebdomadaires réalistes et aléatoires. Le tableau 4.1 dévoile le nombre de variables et de contraintes pour chaque problème d'optimisation résolu. Les solutions des modèles de programmation par contraintes pour chaque scénario seront fournies en entrée au modèle de simulation fait avec une license académique d'Arena Simulation installée sur un laptop Intel(R) Core(TM) i5-8265U CPU @ 1.60GHz 1.80 GHz avec 8 GB de RAM. Chaque scénario sera simulé pour un total de 10 réplifications afin de prendre en compte la variabilité du modèle. Ce nombre de réplifications est considéré suffisant afin de prendre en compte l'aspect stochastique associés

aux distributions utilisées. Environ trois minutes sont nécessaires afin de simuler 10 répliques pour un scénario. Il n'aurait pas été possible d'augmenter de beaucoup plus le nombre de répliques étant donné que les données de sortie seraient beaucoup trop volumineuses. Le modèle de simulation comporte plusieurs distributions triangulaires pour plusieurs facteurs comme la vitesse des véhicules, le temps d'évacuation des gaz après le dynamitage, etc. Une moyenne sur les résultats des 10 répliques est utilisée pour chacun des scénarios. Ce chapitre présentera d'abord les écarts entre le temps de complétion planifié et celui obtenu par la simulation pour chaque modèle CP et ensuite plusieurs comparaisons sont faites entre les modèles CP sans trafic et les modèles CP avec trafic afin de valider l'ajout de contraintes de trafic au modèle de planification (dernier sous-objectif).

4.1 Planifié vs simulé

Dans cette section, les temps de complétion des planifications provenant des modèles CP sont comparés avec les temps de complétion provenant de la simulation de ces mêmes planifications. Maintenant, plus l'écart entre le planifié et le simulé est faible, plus le modèle CP permet d'obtenir des planifications qui se rapprochent de la réalité. L'intérêt d'utiliser un modèle CP plus réaliste est de concevoir des plans de production qui ne causeront pas de surprises aux opérations en temps réel. Par exemple, une planification avec un temps de complétion de 800 dizaines de minutes provenant d'un outil de planification qui possède une grande différence entre le planifié et le simulé pourrait prendre 1000 dizaines de minutes à être complété en réalité. En ajoutant la notion de trafic à la phase d'optimisation de la planification, nous nous attendons à ce que cet écart diminue puisque l'outil optimisera tout en considérant les délais associés à la congestion. Les écarts entre les planifications et les simulations de chaque modèle CP sont présentés aux sous-sections 4.1.1 à 4.1.4. Ensuite, les écarts moyens (moyenne sur l'ensemble des scénarios) de chaque modèle sont comparés entre eux à la sous-section 4.1.5 afin de déterminer quel modèle est le plus réaliste à utiliser pour la mine souterraine.

4.1.1 Écarts entre la planification et la simulation pour le modèle CP-M0

Dans tous les scénarios, sauf pour le 5, le modèle CP-M0 avec 10 minutes de résolution sous-estime le temps de complétion (en bleu) nécessaire afin de terminer toutes les tâches à exécuter comme le dévoile la figure 4.1. Ces écarts peuvent être justifiés par l'absence de notion du trafic dans le modèle de planification et la variabilité de plusieurs facteurs, comme l'état des routes et la vitesse des équipements mobiles, qui sont seulement pris en compte dans l'outil de simulation.

Tableau 4.1 Le nombre de variables et contraintes pour chaque scénario en fonction du modèle.

	Scénario	1	2	3	4	5	6	7	8	9	10
CP-M0	Contraintes	71273	70225	75216	58658	64011	76070	64899	59593	68918	62572
	Variables	25482	25122	26963	21129	23008	27149	23282	21410	24633	22461
CP-M1	Contraintes	72474	71426	76417	59859	65212	77271	66100	60794	70119	63773
	Variables	26683	26323	28164	22330	24209	28350	24483	22611	25834	23662
CP-M2	Contraintes	418073	406225	476016	362258	432411	412070	411699	352393	372518	420172
	Variables	65082	63522	72563	55929	65008	65549	62882	55010	59433	63261
CP-M3	Contraintes	886073	856225	908016	686258	756411	952070	789699	712393	858518	762172
	Variables	96282	93522	101363	77529	86608	101549	88082	79010	91833	86061

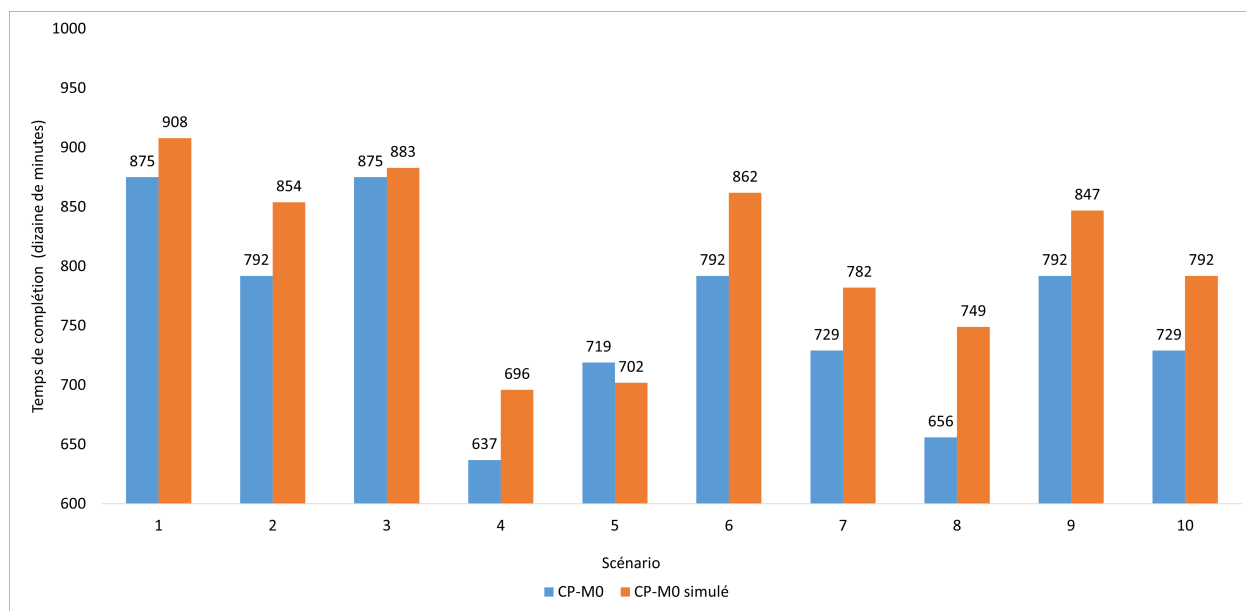


Figure 4.1 Comparaison des temps de complétion des planifications du CP-M0 (10 minutes de résolution) vs les temps de complétion moyens simulés de ces mêmes planifications.

Des résultats similaires sont observés pour le modèle CP-M0 avec 30 minutes de résolution à la figure 4.2. Il est intéressant de remarquer que pour le scénario 7, avec une résolution de 30 minutes, l'écart entre le planifié et le simulé est d'au moins 133 dizaines de minutes. Ainsi, si la mine utilisait ce modèle, les opérations en temps réel auraient été beaucoup plus lentes que prévu.

Il est à mentionner que les résultats diffèrent entre le modèle CP-M0 avec une résolution de 10 minutes et celui avec une résolution de 30 minutes, car les planifications obtenues sont différentes. D'ailleurs, en observant le scénario 7, le modèle qui résout en 10 minutes trouve un temps de complétion de 729 dizaines de minutes et le modèle qui résout en 30 minutes trouve une valeur de 875 dizaines de minutes. À première vue, on s'attendrait à ce que le modèle qui a le plus de temps pour résoudre trouve un temps de complétion plus faible, mais la programmation par contraintes utilise plusieurs heuristiques qui peuvent ne pas trouver la même solution lorsque des hyperparamètres, comme le temps de recherche, sont modifiés. Par contre, pour les neuf autres scénarios, un temps de recherche plus élevé a permis d'obtenir des temps de complétion identiques, voire meilleurs.

4.1.2 Écarts entre la planification et la simulation pour le modèle CP-M1

Comme il a été mentionné à la sous-section 3.2.4, ce modèle de planification avec trafic utilise une somme de trafic dans la fonction objectif. On cherche ainsi à minimiser cette somme. Un

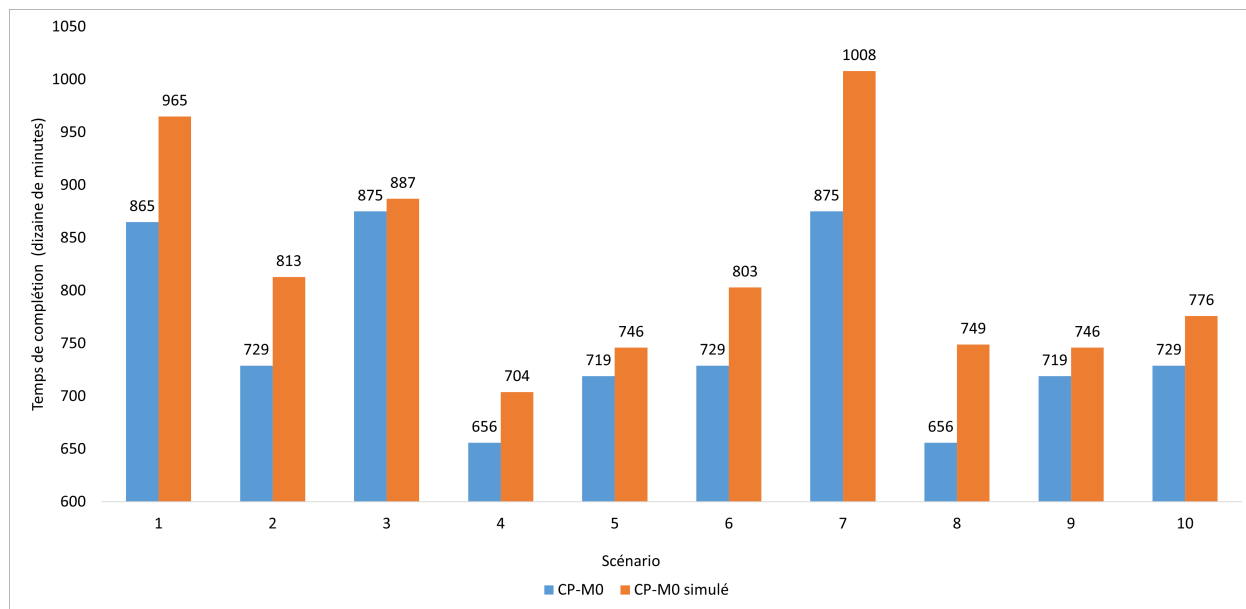


Figure 4.2 Comparaison des temps de complétion des planifications du CP-M0 (30 minutes de résolution) vs les temps de complétion moyens simulés de ces mêmes planifications.

comportement attendu est l'augmentation des temps de complétion des scénarios, car plus les tâches sont éloignées, plus le trafic est diminué. Les différentes versions du modèle sont notées CP-M1Pk où k représente la position de la somme de trafic dans l'ordre lexicographique. La figure 4.3 dévoile qu'effectivement, lorsque la somme de trafic est placée en première position (barres en orange), toutes les tâches de priorité 1 à 6 sont dispersées pour éviter le plus possible le trafic et les écarts entre le planifié et le simulé sont au maximum. Un écart négatif signifie que le temps de complétion du modèle de planification est plus élevé que le temps de complétion moyen simulé. En changeant la position de la somme de trafic dans la fonction objectif, de moins en moins de tâches peuvent être dispersées afin de minimiser le trafic. Par exemple, lorsque la somme de trafic est en position 4 (CP-M1P4), le modèle de planification cherchera à minimiser les temps de complétion des tâches de priorité 1, 2 et 3 avant de minimiser la somme de trafic. Seules les tâches de priorité 4 à 6 peuvent être dispersées pour réduire la somme de trafic. Moins que les tâches sont dispersées, moins que l'écart entre le planifié et le simulé est grand. Cet écart peut se justifier par le fait que le modèle CP-M1 surestime l'impact du trafic sur la planification en ayant la possibilité de disperser les tâches afin d'améliorer la fonction objectif. Il est d'ailleurs possible de remarquer une forme d'escalier dans la figure 4.3 pour la plupart des scénarios, par exemple le scénario 10.

De grands écarts, comme -800 dizaines de minutes, entre la planification et la simulation peuvent être expliqués du fait que le modèle de simulation utilise les séquences de tâches

pour chaque équipement mobile provenant du modèle CP sans prendre en compte les temps de début des tâches. Par exemple, dans le modèle CP-M1P1, la tâche i peut seulement commencer à la 300^{ième} dizaine de minutes par l'équipement A tandis que dans le modèle de simulation, si l'équipement A est rendu à la tâche i dans sa séquence et que la station de travail est rendue à la tâche i dans son cycle, la tâche sera débutée, peu importe le moment. Il est alors très difficile d'utiliser ce type de résultat afin de juger de la qualité du modèle CP-M1. Une analyse plus approfondie de l'efficacité de ce modèle sera montrée à la section 4.2. Cependant, pour le modèle CP-M1P7, les temps de complétion de toutes les tâches ne peuvent être modifiés et cela produit ainsi des écarts positifs pour tous les scénarios. De plus, même si ce modèle ne peut pas disperser les tâches, lorsqu'il y a plusieurs solutions équivalentes, il permet de choisir la solution avec le moins de trafic. Plus tard dans cette section, les écarts des sept versions du modèle CP-M1 seront comparés avec les écarts des autres modèles.

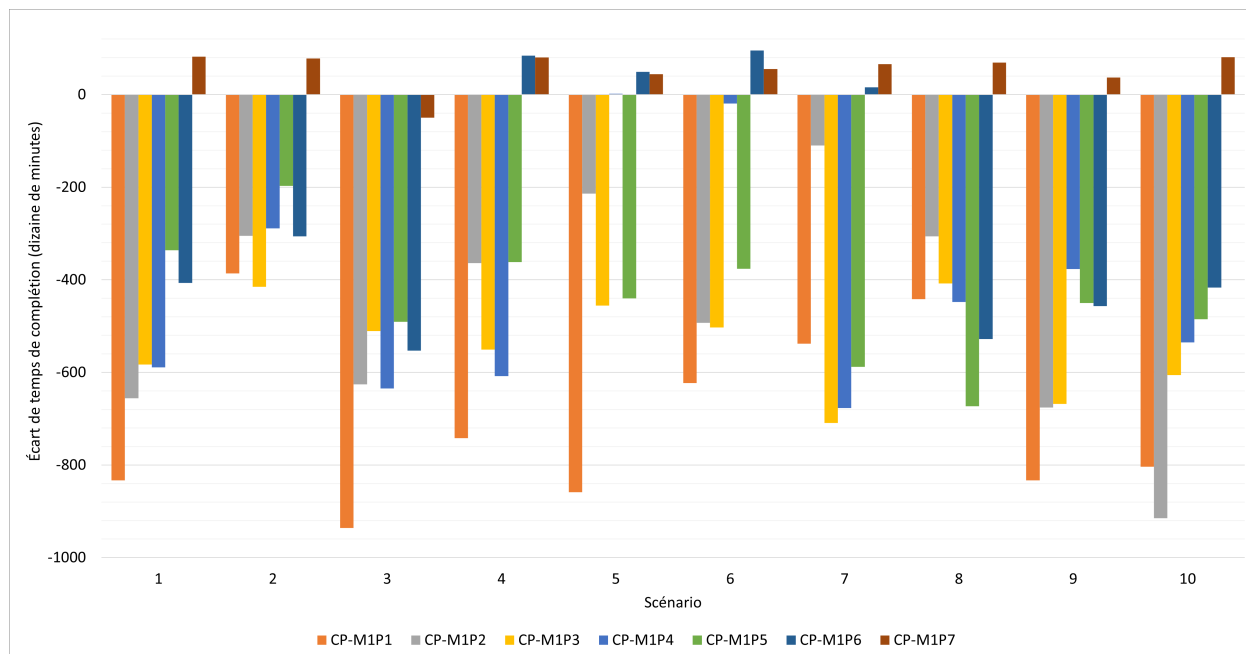


Figure 4.3 Comparaison des écarts entre les temps de complétion simulés avec les temps de complétion des planifications pour tous les modèles CP-M1.

4.1.3 Écarts entre la planification et la simulation pour le modèle CP-M2

Tout comme les résultats du modèle CP-M0, la marge entre le planifié (barre bleue) et le simulé (barre orange) semble être d'environ un quart de travail (72 dizaines de minutes) pour la majorité des scénarios (voir la figure 4.4). Cependant, pour les scénarios 3 à 5, l'outil de

planification surestime le temps de complétion nécessaire pour toutes les tâches d'environ 30 dizaines de minutes. Une explication à cette observation serait que l'outil pourrait surestimer le trafic qui se produit réellement. Pour l'outil de planification, les délais de trafic entre les tâches de développement proviennent d'une moyenne de trafic en fonction du nombre de chargeuses actives à chaque unité de temps. Par exemple, pendant trois unités de temps t_1, t_2 et t_3 il y a deux chargeuses actives. Sachant que le délai moyen de trafic est de 0.1 dizaine de minutes à chaque unité de temps lorsque deux chargeuses sont actives, le délai de trafic total planifié s'avère alors de $0.1 + 0.1 + 0.1 = 0.3$ dizaines de minutes. Tandis que dans la simulation, il se peut que le délai de trafic total soit de 0.1 dizaine de minutes. Aussi, le fait que les tâches n'ont pas un temps de début fixe dans la simulation peut avoir eu un effet sur ces résultats. Par exemple, il se peut que la tâche i soit planifiée de débuter à la 100^{ième} dizaine de minutes. Dans la simulation, la tâche pourrait débuter à la 98^{ième} dizaine de minute si la tâche prédécesseur à i est terminée plus tôt que prévu et si l'emplacement de la tâche i est rendu à la bonne étape de son cycle d'activités.

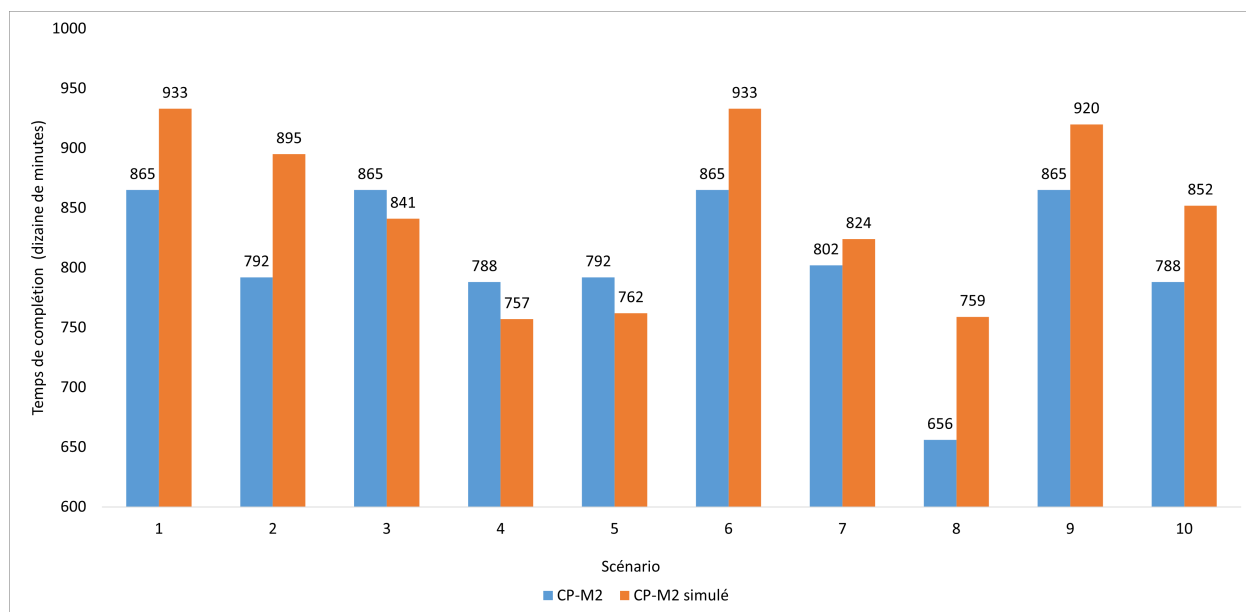


Figure 4.4 Comparaison des temps de complétion des planifications du CP-M2 vs les temps de complétion moyens simulés de ces mêmes planifications.

4.1.4 Écarts entre la planification et la simulation pour le modèle CP-M3

Pour ce modèle qui intègre les délais de congestion associés aux activités de développement et de production, les marges entre le planifié et le simulé semblent plus faibles que dans tous les autres modèles au premier regard comme le dévoile la figure 4.5. Ces résultats sont

comparés à ceux des autres modèles dans la sous-section suivante afin de déterminer si le modèle CP-M3 est bel et bien le plus efficace à produire des planifications plus réalistes.

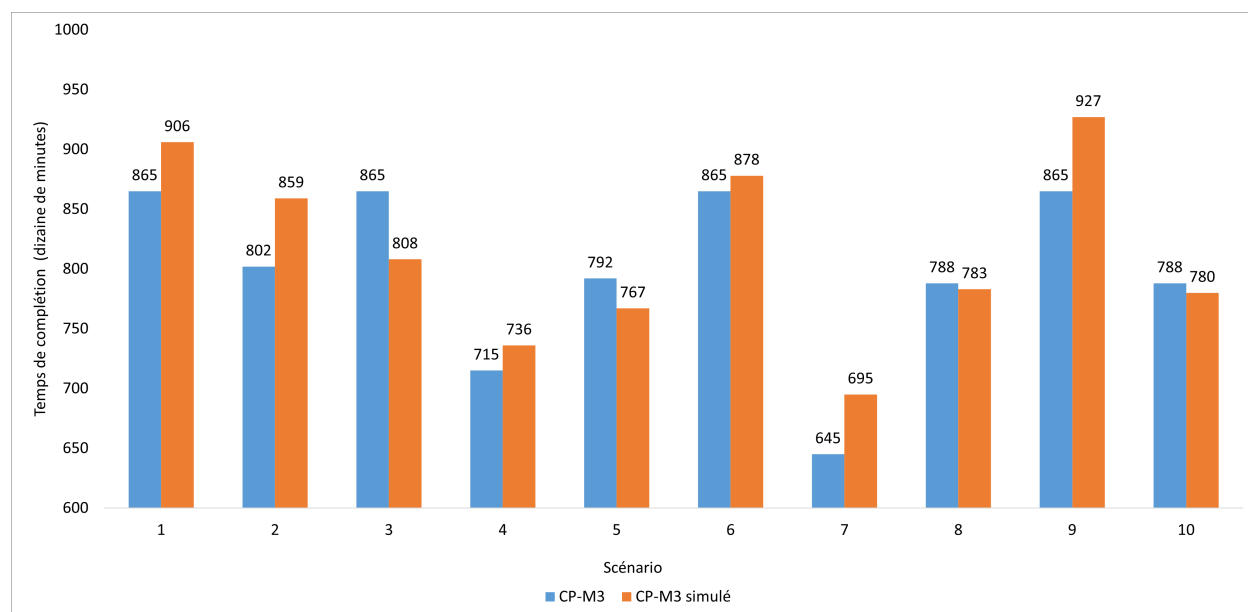


Figure 4.5 Comparaison des temps de complétion des planifications du CP-M3 vs les temps de complétion moyens simulés de ces mêmes planifications.

4.1.5 Comparaison des écarts entre la planification et la simulation de tous les modèles

En comparant chaque modèle en fonction de leur écart de temps de complétion moyen (moyenne sur les 10 scénarios) entre le planifié et le simulé, il est possible d'observer sur la figure 4.6 que le modèle CP-M3 possède l'écart le plus faible, soit 14.9 dizaines de minutes. Ceci signifie que lorsque le département des opérations utilisera cet outil, qui intègre la notion de trafic, on peut s'attendre à ce que le temps de complétion réel soit plus élevé d'au moins 15 dizaines de minutes sur un ensemble de tâches hebdomadaire. On utilise le terme au moins, car l'outil de simulation est une représentation de la réalité qui a ses limitations. Le modèle CP-M3 s'avère alors meilleur que l'outil de base sans trafic CP-M0 (30 minutes de résolution) qui possède un écart moyen 4 fois plus élevé. Ici, on compare le modèle CP-M3 avec le CP-M0 (30 minutes de résolution), car comme il a été mentionné à la section 3.2.4 le temps de recherche utilisé pour le modèle CP-M3 est de 30 minutes afin d'obtenir des solutions réalisables. Également, il est à noter qu'un écart moyen positif signifie que le temps de complétion moyen simulé (moyenne sur les 10 répliquions) est plus élevé que le temps de complétion planifié. Un écart moyen négatif signifie l'inverse.

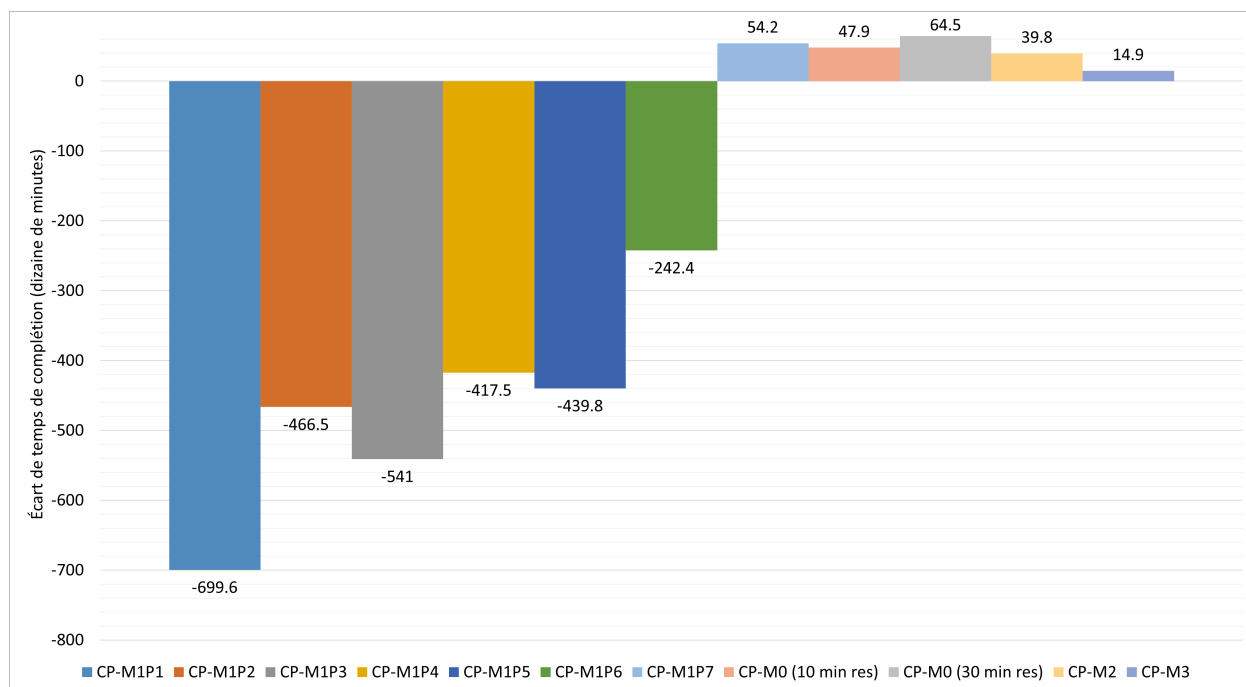


Figure 4.6 Comparaison des écarts moyens (moyenne sur les 10 scénarios) entre les temps de complétion simulés et les temps de complétion planifiés pour tous les modèles.

De plus, le modèle CP-M2, qui considère le trafic que pour les tâches de développement, est le deuxième meilleur modèle en termes d'écart moyen entre le planifié et le simulé. Cependant, lorsque le modèle CP-M2 est comparé au modèle de base sans trafic CP-M0 (10 minutes de résolution), la différence est seulement que d'environ 8 dizaines de minutes. Les modèles CP-M1P1 à CP-M1P6 possèdent de grands écarts négatifs donc on ne peut prédire à quoi ressembleront les temps de complétion réels. Par exemple, avec le modèle CP-M1P1, le temps de complétion planifié peut être de 1800 dizaines de minutes et lorsque simulé il n'en prend que 1000 dizaines de minutes. Si ce plan était exécuté en réalité, on ne pourrait pas savoir d'avance le temps total de complétion réel sans l'outil de simulation. Ceci s'avère alors très risqué à utiliser pour planifier les opérations d'une mine réelle qui a des objectifs à atteindre à court, moyen et long terme. Le modèle CP-M1P7 possède un écart positif, mais plus élevé que le modèle de base sans trafic CP-M0 (10 minutes de résolution). Donc la méthode de prise en compte du trafic dans ce modèle, sous la forme d'une somme de trafic dans la fonction objectif, est moins pertinente que celles provenant du modèle CP-M2 et CP-M3, qui utilisent des contraintes de délais de trafic entre les tâches, afin de prédire le temps de complétion réel.

Les écarts moyens absolus entre la planification et la simulation de tous les modèles sont

présentés à la figure 4.7. Afin d’obtenir cette figure, l’écart absolu entre le planifié et le simulé pour chaque scénario est déterminé pour un modèle. La moyenne de ces écarts est ensuite calculée. Cette démarche est répétée pour tous les modèles de planification. En analysant ces moyennes absolues, le modèle CP-M3 reste le plus performant afin d’obtenir les planifications les plus réalistes. Le modèle CP-M0 (30 minutes de résolution) a un écart moyen absolu deux fois plus élevé que celui provenant du modèle CP-M3. En suite, les modèles CP-M1P7 et CP-M2 possèdent des écarts absolus plus élevés, lorsque comparés au modèle CP-M0 (10 minutes de résolution). Ces deux modèles produisent alors des planifications moins réalistes que le modèle de base sans trafic.

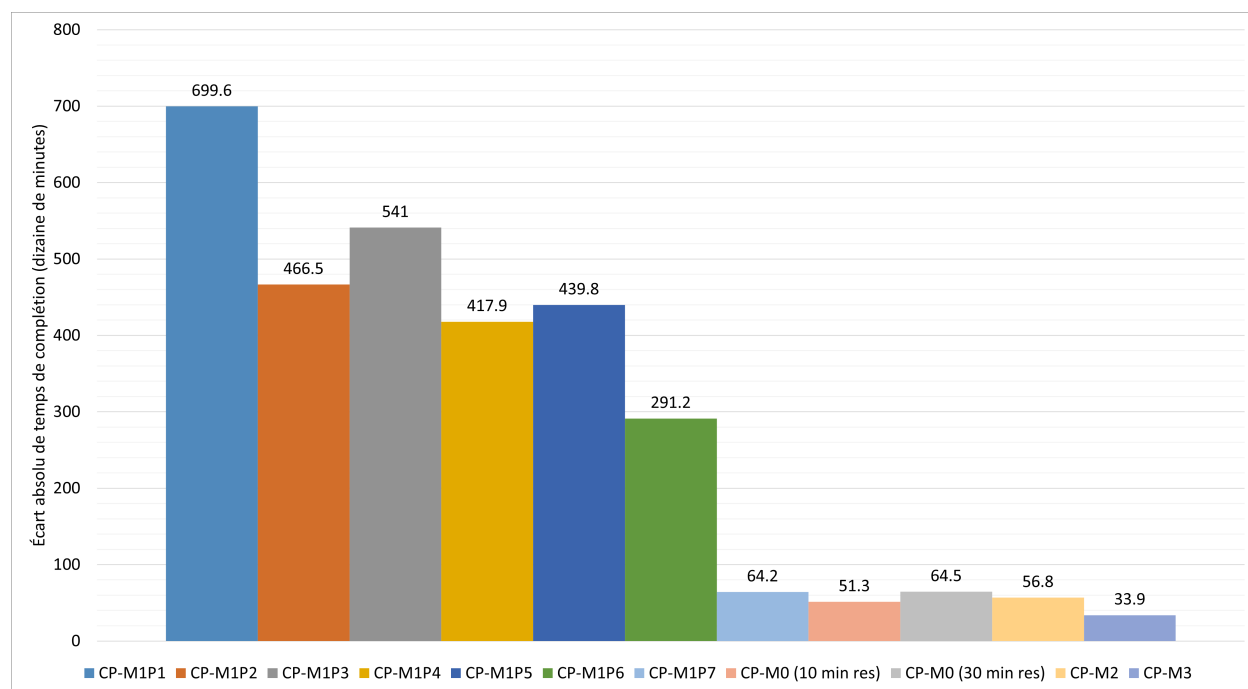


Figure 4.7 Comparaison des écarts moyens absolus (moyenne sur les 10 scénarios) entre les temps de complétion simulés et les temps de complétion planifiés pour tous les modèles.

Enfin, après avoir analysé les écarts entre le planifié et le simulé, il est clair que le modèle CP-M3 avec trafic permet d’obtenir des planifications qui sont plus réalistes. En utilisant cet outil, le planificateur peut dire au département des opérations le temps de complétion (planifié) qu’il faudra pour exécuter les tâches de la semaine. Lorsque ces tâches seront finies, le temps de complétion réel sera très près de la valeur donnée par le planificateur. Sachant que les planifications sont plus réalistes avec la notion de trafic, il reste à savoir si l’ajout du trafic dans les outils de planification permet d’obtenir de meilleures solutions en réduisant le temps de complétion simulé. La prochaine section présente une analyse à cet effet.

4.2 Optimalité des outils sans trafic vs avec trafic

Les temps de complétion simulés provenant des modèles avec contraintes de trafic sont comparés avec ceux provenant des modèles sans contraintes de trafic. L'intérêt de cette comparaison est de déterminer quel modèle permet d'obtenir les meilleures planifications (temps de complétion minimal) pour nos scénarios lorsqu'elles sont exécutées dans le modèle de simulation. En utilisant le meilleur modèle, le planificateur peut s'attendre à ce que les tâches hebdomadaires soient terminées plus rapidement que s'il utilisait les autres modèles. Avec de meilleures planifications, la mine peut s'attendre à extraire de plus grands volumes de roche sur une même période de temps, ce qui représente des gains économiques non négligeables pour l'entreprise. Dans cette section, chaque modèle avec trafic sera comparé avec un modèle sans trafic possédant le même temps de recherche en hyperparamètre. Après, une comparaison globale des modèles sera faite afin de bien visualiser quel modèle permet de minimiser les temps de complétion simulés des scénarios.

4.2.1 CP-M0 vs CP-M1

Dans l'ensemble des scénarios, le modèle CP-M0 (10 minutes de résolution) possède les temps de complétion moyens simulés parmi les plus faibles en observant la figure 4.8. Par contre, dans 6 scénarios sur 10, une version du modèle CP-M1 possède le temps de complétion minimal. Il ne semble pas y avoir une version du modèle CP-M1 qui prédomine dans ces 6 scénarios. Par exemple, pour le scénario 1, le modèle CP-M1P2 est le meilleur tandis que dans les autres scénarios, ce modèle possède des temps de complétion très élevés. Une approche du planificateur serait de planifier les tâches hebdomadaires dans le modèle CP-M0 et les 7 versions du modèle CP-M1 pour ensuite simuler chacun de ces plans afin de déterminer quelle planification sera la meilleure. Cependant, ces démarches sont coûteuses en temps, puisqu'il faudrait $8 * 10 \text{ min} = 80 \text{ min}$ au minimum pour obtenir les planifications des modèles. Sans oublier le temps consacré à la simulation de toutes ces planifications.

En analysant les durées de trafic moyen simulés (moyenne sur les 10 répliques), il est possible d'observer que le modèle CP-M1P1 obtient les durées de trafic les plus faibles dans la majorité des scénarios (voir la figure 4.9). Aussi, dans plusieurs scénarios, comme le 6 et le 8, les durées de trafic des modèles CP-M1 augmentent plus que la somme de trafic diminuée en priorité dans la fonction objectif du modèle. Ainsi, cela nous indique que le modèle CP-M1 permet bel et bien une diminution du trafic grâce à l'ajout du trafic dans la fonction objectif d'un modèle de planification. Cependant, cet ajout ne permet pas de produire des solutions plus optimales. En dispersant les tâches, le trafic diminue, mais les temps de complétion

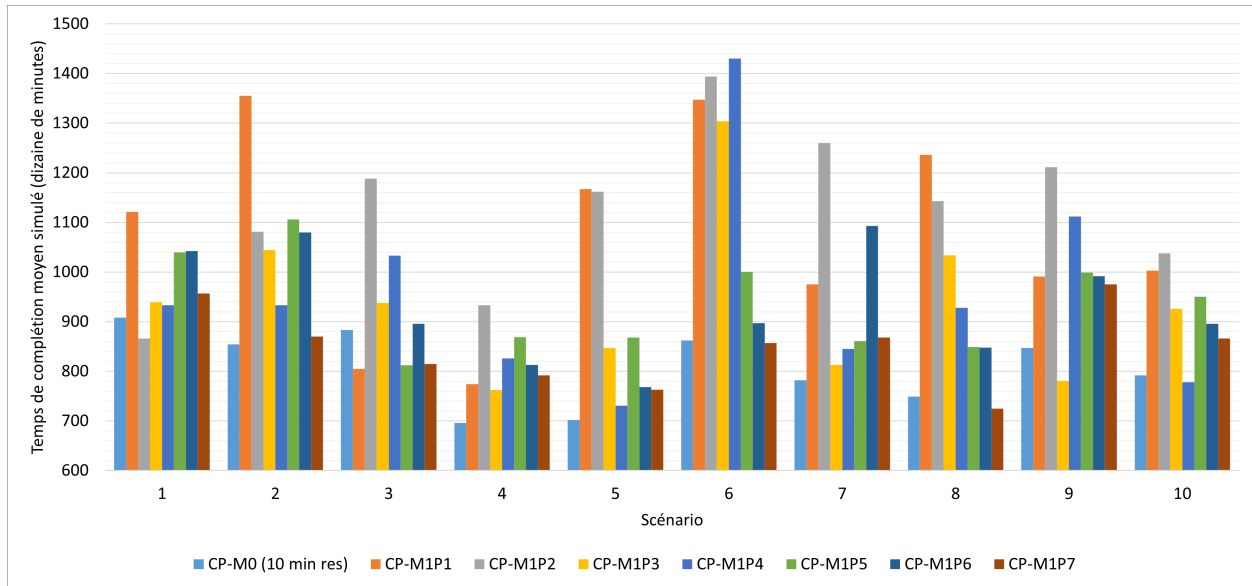


Figure 4.8 Comparaison des temps de complétion moyens simulés provenant du CP-M0 (10 minutes de résolution) avec ceux des modèles CP-M1.

augmentent, ce qui n'est pas profitable pour la mine. D'ailleurs, en observant le scénario 9, le modèle (CP-M1P3) avec la durée de trafic la plus élevée possède le temps de complétion le plus faible. Logiquement, en rapprochant les tâches entrent-elles, le trafic augmente et la productivité aussi.

Étant donné les écarts élevés entre le planifié et le simulé discutés à la sous-section 4.1.2 et l'absence d'un modèle CP-M1 prédominant pour l'ensemble des scénarios, cette méthode d'ajout du trafic dans la planification n'est pas recommandée. Des approches utilisant des contraintes de délais de trafic entre les tâches, comme les modèles CP-M2 et CP-M3, sont moins drastiques sur la minimisation du trafic. Les résultats de ces méthodes sont abordés dans les prochaines sous-sections.

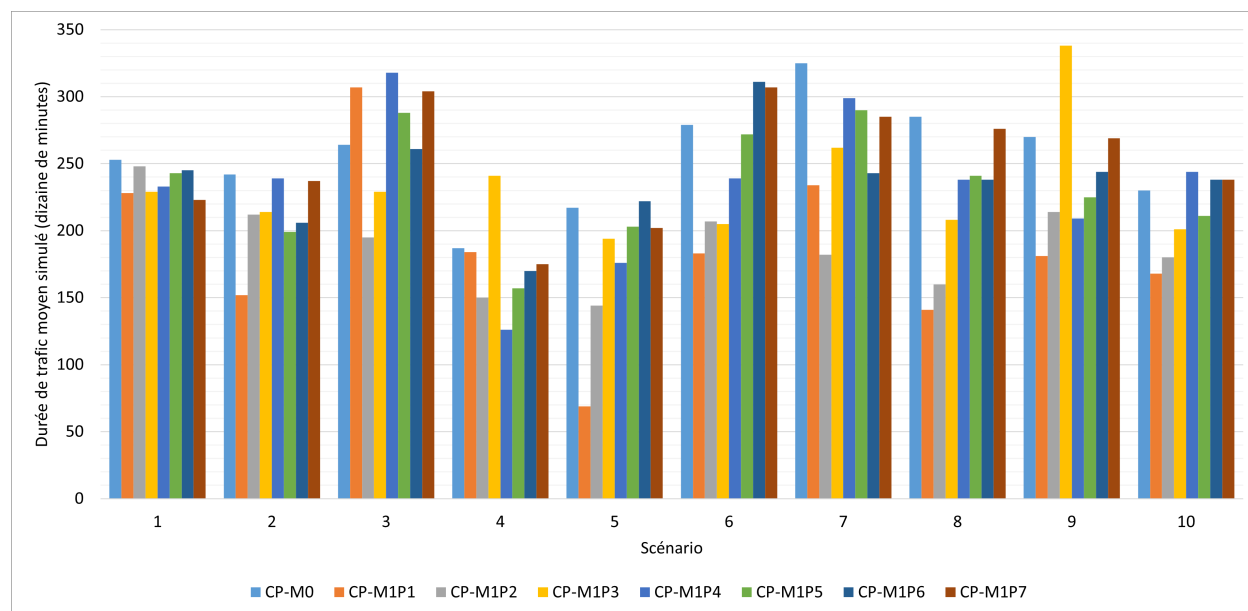


Figure 4.9 Comparaison des durées de trafic moyens simulés provenant du CP-M0 (10 minutes de résolution) avec ceux des modèles CP-M1.

4.2.2 CP-M0 vs CP-M2

La figure 4.10 dévoile que pour tous les scénarios, sauf le 3, le modèle CP-M2 donne des planifications qui ont de plus grands temps de complétion simulés lorsque comparé au modèle CP-M0 (10 minutes de résolution). À la lumière de ces résultats, le modèle CP-M2 permettrait ainsi des planifications plus réalistes (voir la sous-section 4.1.5), mais moins performantes en termes de temps de complétion lorsque comparé au modèle CP-M0.

La durée du trafic simulé est plus faible avec le modèle CP-M2 dans tous les scénarios, sauf le 3 (voir la figure 4.11). La marge moyenne de trafic entre les barres bleues et oranges est d'environ 20 dizaines de minutes. Il est à noter que la relation entre les délais de trafic et le temps de complétion des scénarios pour le modèle CP-M2 est identique à celle provenant des résultats du modèle CP-M1. Effectivement, dans le scénario 3, le modèle CP-M2 génère plus de trafic, mais permet d'atteindre un temps de complétion de 841 dizaines de minutes au lieu de 883 dizaines de minutes (modèle CP-M0).

Enfin, seules les tâches de chargement en développement possèdent des délais de trafic dans le modèle CP-M2, ce qui n'est qu'une partie du trafic global. De grands volumes de roches se situent dans les chantiers de production. Une tâche de chargement en production s'étale sur plusieurs heures voire plusieurs jours. L'ajout d'un délai de trafic qui est l'accumulation du trafic lors de la tâche de production permettrait une meilleure représentation de la réalité

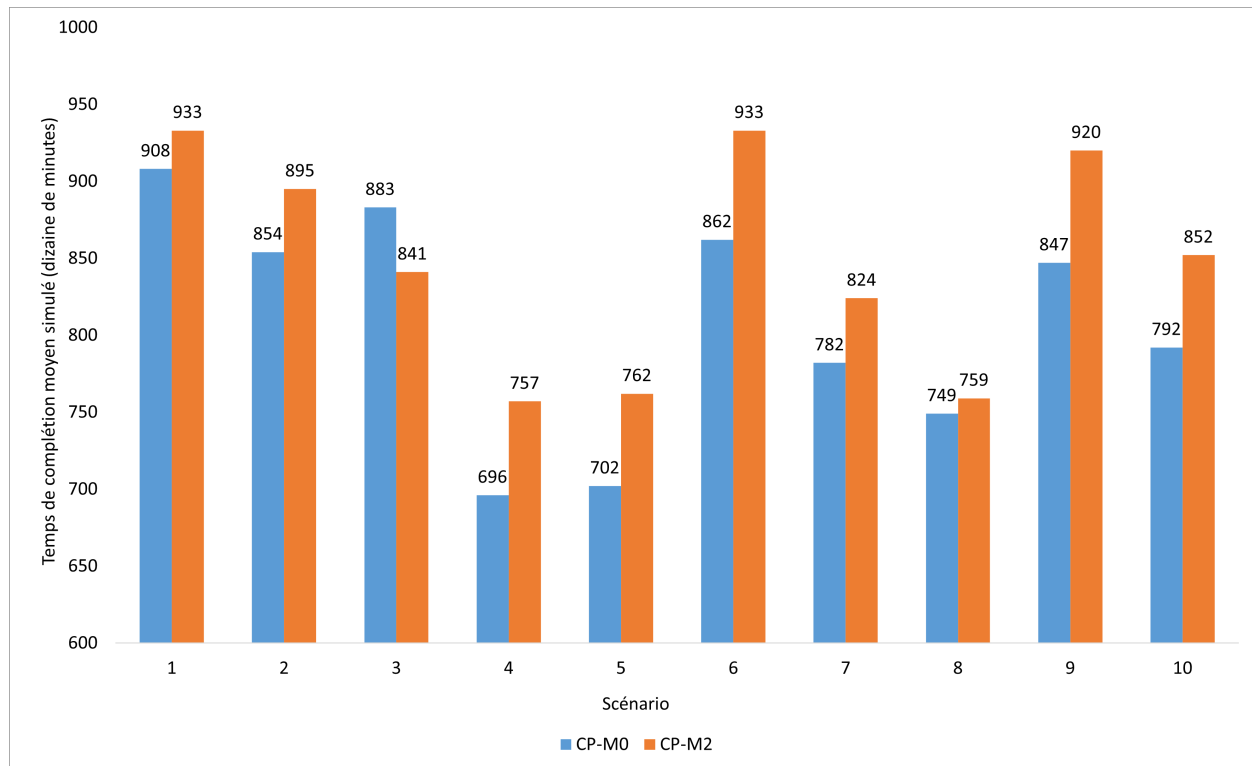


Figure 4.10 Comparaison des temps de complétion moyens simulés provenant du CP-M0 (10 minutes de résolution) avec ceux du CP-M2.

dans l'outil de planification. Effectivement, ces délais qui ralentissent la fin des tâches de production dans la réalité sont non-négligeables. Ainsi, en ajoutant ces délais à la fin des tâches de production, le solveur pourrait effectuer un meilleur ordonnancement de tâches. Le modèle CP-M3 a été conçu à cet effet.

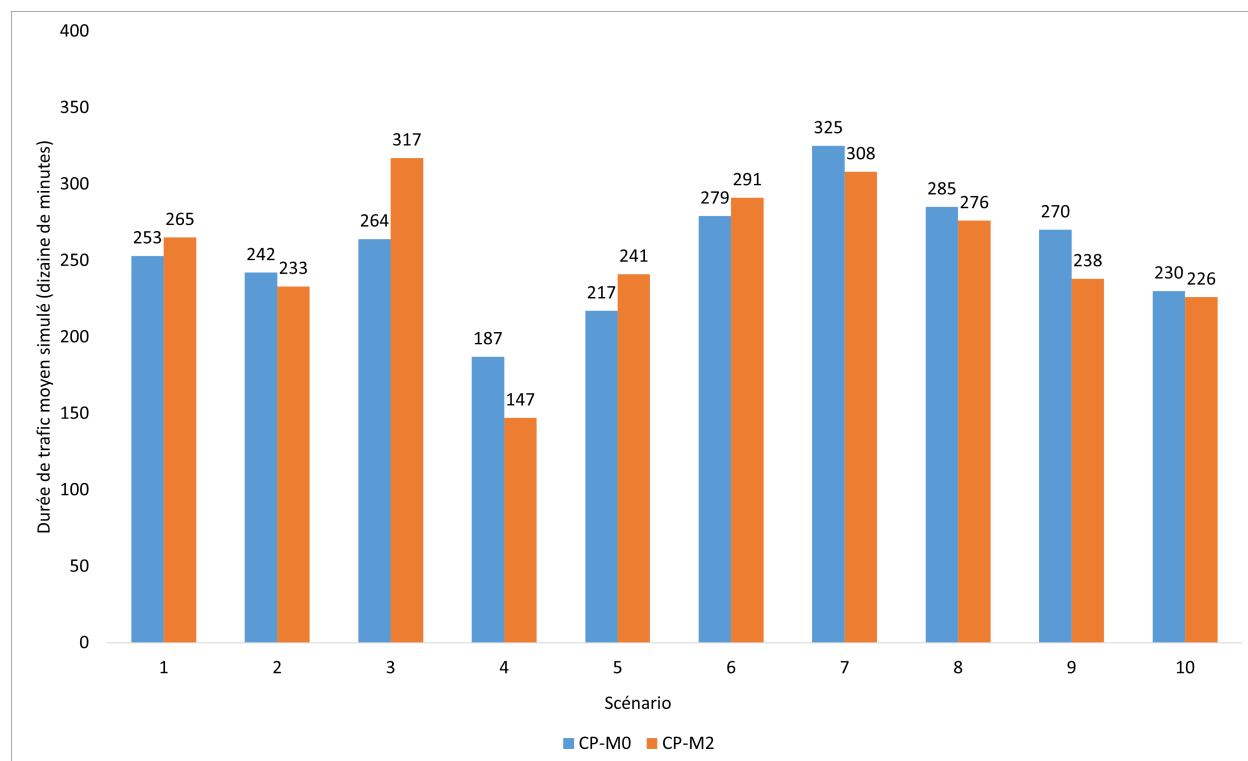


Figure 4.11 Comparaison des durées de trafic simulés provenant du CP-M0 (10 minutes de résolution) avec ceux du CP-M2.

4.2.3 CP-M0 vs CP-M3

Pour les scénarios 1 et 3, le modèle CP-M3 obtient des temps de complétion simulés plus petits que ceux du modèle CP-M0 (30 minutes de résolution), soient 906 et 808 dizaines de minutes comparativement à 965 et 887 dizaines de minutes. De plus, le scénario 7 est aussi terminé plus rapidement grâce au modèle CP-M3. L'écart entre le modèle avec trafic et sans trafic pour ce scénario est d'environ 300 dizaines de minutes, ce qui est une très grande amélioration de la qualité de la planification. Cependant, le modèle CP-M0 obtient un temps de complétion aussi très faible pour le scénario 9 avec un écart d'environ 180 dizaines de minutes. Ensuite, pour le scénario 10, les deux modèles obtiennent des temps de complétion presque identiques. L'ajout des délais de trafic aux tâches de production a ainsi permis d'améliorer la qualité des planifications de plus de scénarios, lorsque comparé au cas du modèle CP-M2 présenté à la sous-section 4.2.2.

En ce qui concerne le trafic généré lors des simulations, les planifications du modèle CP-M3 n'obtiennent pas toujours des durées de trafic plus faibles, lorsque comparées aux planifications du modèle CP-M0 comme le dévoile la figure 4.13. Par exemple, pour le scénario 1 où

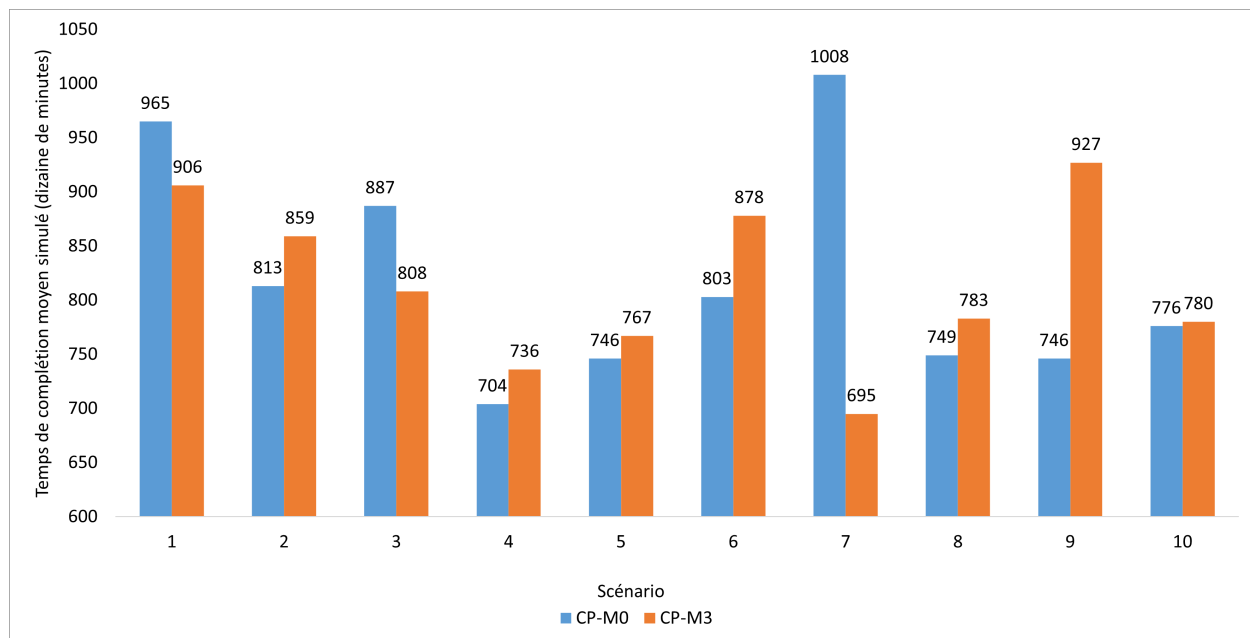


Figure 4.12 Comparaison des temps de complétion moyens simulés provenant du CP-M0 (30 minutes de résolution) avec ceux du CP-M3.

le modèle CP-M3 possède le meilleur de temps de complétion, la durée de trafic est de 235 dizaines minutes, ce qui est plus faible que le trafic provoqué par la planification du modèle CP-M0 (256 dizaines de minutes). À l'inverse, pour le scénario 7, la durée de trafic avec la planification du modèle CP-M3 est de 354 dizaines de minutes contre 239 dizaines de minutes pour la planification du modèle CP-M0. Il est possible alors d'affirmer qu'en ajoutant des contraintes de délais de trafic à la planification, le solveur cherche à diminuer ces délais, mais n'en fait pas une priorité, contrairement au modèle CP-M1. Un avantage de ces nouvelles contraintes est que la durée des tâches, avec les délais, se rapproche de la réalité et le solveur peut mieux ordonnancer les tâches. Par exemple, un modèle de planification sans contraintes de trafic va affecter une ressource k à une tâche i qui débute à la 200^{ième} dizaine de minutes. Cependant, en sachant que le trafic a ralenti la fin de la tâche prédécesseur j , la tâche i peut seulement débiter à la 208^{ième} dizaine de minutes. L'outil de planification avec contraintes de trafic peut alors affecter une autre tâche l à la ressource k en attendant que la tâche i devienne disponible.

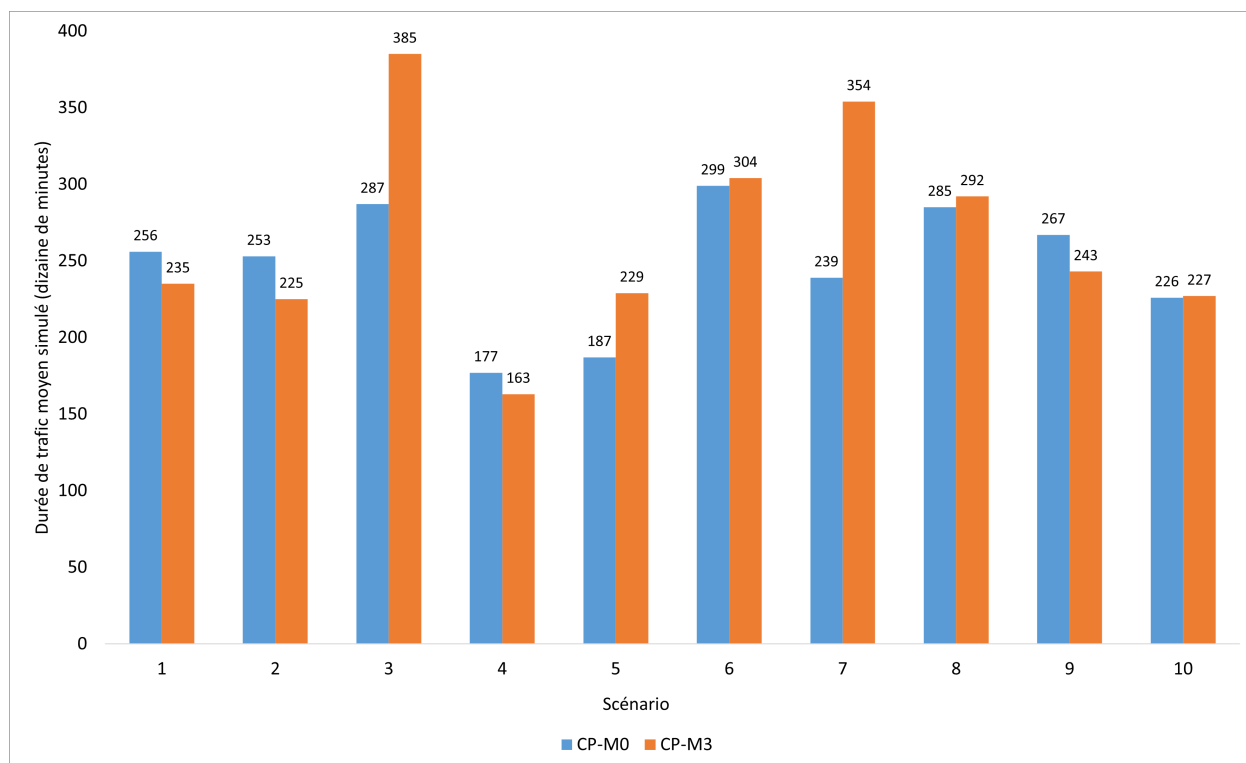


Figure 4.13 Comparaison des durées de trafic simulés provenant du CP-M0 (30 minutes de résolution) avec ceux du CP-M3.

4.2.4 Comparaison des temps de complétion simulés de tous les modèles

Afin de déterminer quel modèle permet d'obtenir les temps de complétion les plus faibles pour l'ensemble des scénarios, les résultats de chaque modèle sont présentés à la figure 4.14. Il s'agit ici des écarts de temps de complétion moyens simulés (moyenne sur les 10 scénarios) entre chaque modèle avec trafic et le modèle sans trafic associé (même temps de recherche). Par exemple, pour le modèle CP-M1P1, le temps de complétion est plus élevé de 269.9 dizaines de minutes en moyenne sur l'ensemble des scénarios, lorsque comparé au modèle CP-M0 (10 minutes de résolution). Le modèle CP-M3 permet de diminuer le temps de complétion de 5.8 dizaines de minutes en moyenne sur l'ensemble des résultats. Aussi, le lecteur peut se référer à la figure B.1 en annexe afin de voir les écarts pour chaque scénario.

En se fiant seulement à la moyenne, l'ajout de contraintes de trafic à la planification permet ainsi de diminuer le temps de complétion des tâches hebdomadaires. Lorsque transposé à un cas d'application réel, le planificateur peut s'attendre à réduire en moyenne 6 dizaines de minutes (1 heure) par semaine le temps de complétion de toutes les tâches pendant 10 semaines (10 scénarios hebdomadaires). Comme il a été mentionné à la sous-section 4.2.3, le

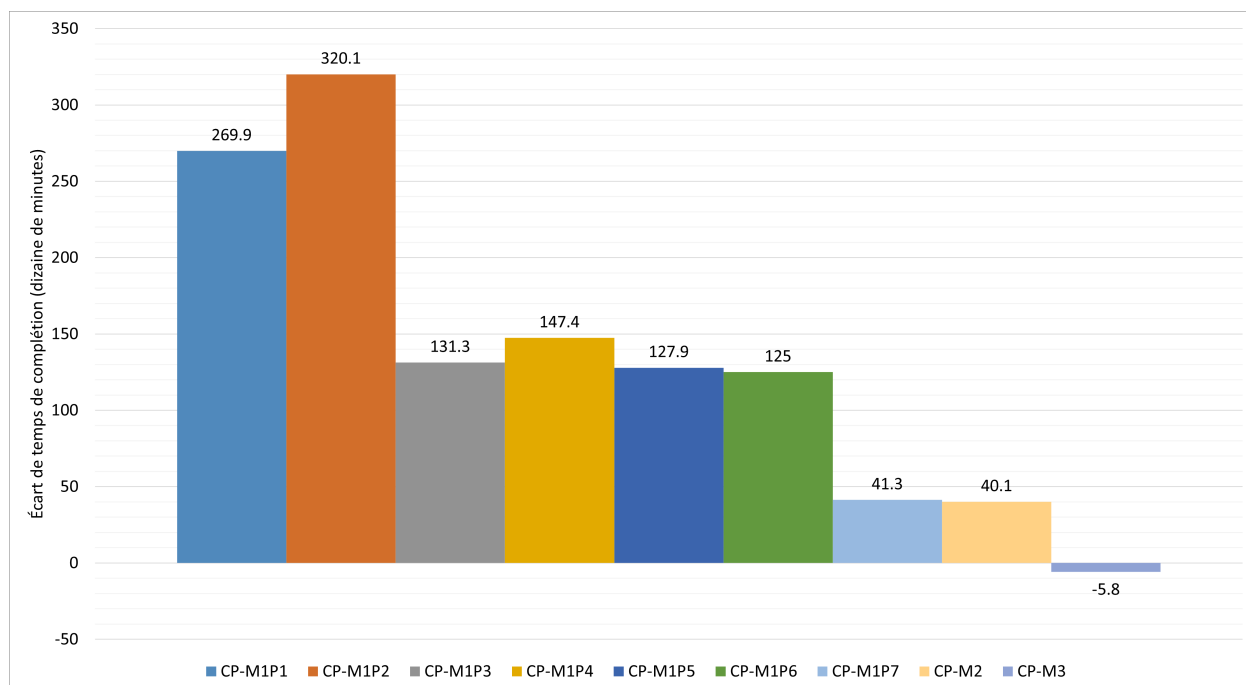


Figure 4.14 Comparaison des écarts moyens entre les temps de complétion simulés des modèles de trafic (CP-M1, CP-M2 et CP-M3) avec ceux des modèles sans trafic.

modèle CP-M3 obtient le meilleur temps de complétion pour 30% des scénarios et un temps de complétion équivalent au modèle CP-M0 pour 10% des scénarios. Les gains de temps obtenus dans ces 3 scénarios surpassent les pertes de temps provenant des 6 autres scénarios. La somme des écarts pour les 10 scénarios est de -58 dizaines de minutes ou environ 10 heures. Ce gain d'environ un quart de travail sur 10 semaines reste cependant relativement faible.

Au final, le plus grand avantage d'utiliser le modèle CP-M3 est qu'il permet d'obtenir des planifications qui se rapprochent plus de la réalité lorsque comparé aux autres modèles comme il a été dévoilé dans la sous-section 4.1.5. De plus, en moyenne, des gains sur les temps de complétion sont obtenus grâce à ce modèle, mais ceux-ci restent peu élevés. Pour ces raisons, nous recommandons l'ajout de contraintes de trafic aux outils de planification à court terme dans les mines souterraines.

CHAPITRE 5 CONCLUSION ET RECOMMANDATIONS

En synthèse, l'objectif de ce mémoire était d'inclure des contraintes de trafic dans les outils de planification à court terme des mines souterraines. D'ailleurs, la revue de la littérature présentée au chapitre 2 a permis de démontrer qu'aucun autre modèle de planification à court terme n'a tenté de considérer le trafic encouru par l'ordonnancement des activités. Afin d'atteindre l'objectif fixé, trois sous-objectifs ont été établis. Le premier sous-objectif était de modéliser le trafic généré par les activités de production. En développant un modèle de simulation à événements discrets, il a été possible de recréer la mine sous un format numérique à l'aide de certaines limitations. Le modèle de simulation permet alors de tester les diverses planifications obtenues par nos outils de planification. Plus précisément, il a été possible de recueillir les temps de complétion et les durées de trafic associés à chaque planification exécutée dans l'environnement numérique. Le deuxième sous-objectif était de déterminer les contraintes de trafic à ajouter à l'outil de planification. Trois modèles ont été créés à cet effet. Le modèle CP-M1 utilise une somme de trafic dans sa fonction objectif afin de minimiser le trafic. Une version du modèle est créée pour chaque position prise par la somme de trafic dans la fonction objectif. Après, le modèle CP-M2 utilise plutôt des délais de trafic entre les tâches de développement pour répliquer les retards des activités causés par la congestion. Enfin, le modèle CP-M3 est une réplique du modèle CP-M2, mais considère aussi les tâches de production. L'hyperparamètre de temps de recherche du solveur a été haussé de 10 minutes à 30 minutes afin d'obtenir des solutions réalisables avec ce modèle. Pour le dernier sous-objectif, 10 scénarios ont été générés et une planification par chaque modèle de planification a été obtenue pour chacun de ces scénarios. Les résultats démontrent que l'approche des contraintes de délais de trafic entre les tâches rend les planifications plus réalistes et contribue à réduire de beaucoup le temps de complétion de certains scénarios.

Afin d'arriver aux conclusions tirées dans ce document, plusieurs limitations ont été posées quant aux modèles de simulation et de planification. L'étude se restreint aux tâches opérationnelles affectées aux équipements mobiles tels que les chargeuses LHD, les boulonneuses, les camions à émulsion, les camions et les jumbos. Tout autre équipement mobile comme des tracteurs et des camions de transport d'employés (*man carriers*) ont été exclus. De plus, seules les tâches de chargement de roche sont incluses quant aux chantiers de production de la mine. L'installation des services, comme l'eau et la ventilation, le forage et autres sont exclus de cette étude. L'effet de l'ajout de ces exclusions sur les résultats reste alors inconnu. En ce qui concerne la flotte d'équipements mobiles, seul un type de camion (50 tonnes) a été considéré pour la simulation et la planification afin de simplifier la modélisation.

Dans le modèle de simulation, le nombre de camions pouvant accéder à une voie de dépassement est illimité. Cette limitation permet de simplifier encore une fois la modélisation. Effectivement, il aurait été laborieux de modéliser tous les cas possibles de congestion dans les rampes. Par exemple, si la prochaine voie de dépassement est occupée, le camion qui n'a pas la priorité doit faire marche arrière à une certaine vitesse pendant une certaine distance. Sans oublier qu'il se peut que des camions en arrière de celui-ci doivent aussi reculer.

Les mouvements de va-et-vient entre les emplacements de chargement de roche et la zone de déchargement à la surface ne sont pas pris en compte dans l'outil de planification. Les camions assignés à un emplacement restent à cet endroit pour la durée totale de l'activité de chargement. Il est difficile de planifier d'avance le nombre exact de voyages entre la surface et l'emplacement de travail. De plus, il aurait été nécessaire de planifier chaque rampe à utiliser pour se rendre à un des deux emplacements pour le déchargement à la surface. Cette limitation permet alors de diminuer la complexité de la modélisation à cet égard en attribuant une durée de tâche globale qui comporte le chargement et déchargement des camions affectés à la tâche de chargement de roche.

Pour continuer l'élan de cette recherche, plusieurs avancées peuvent être adressées quant aux limitations des modèles de simulation et de planification présentés dans ce mémoire afin d'optimiser davantage la planification à court terme dans les mines souterraines. La considération du trafic dans la planification est un facteur non négligeable à l'optimisation de la planification dans les mines. Les approches d'ajout de trafic à la planification présentées à la sous-section 3.2.4 sont fondées sur le principe que le nombre de chargeuses LHD en action est un indicateur clé de trafic. Des moyennes de délais de trafic par unité de temps en fonction du nombre de chargeuses actives (en tâche de chargement) ont été utilisées. Les auteurs concèdent qu'il existe forcément d'autres façons possibles de représenter le trafic dans les outils de planification. D'autres méthodes, comme la création de tâches de trafic qui consomment le temps de disponibilité de certains équipements mobiles, permettraient aussi une meilleure représentativité de la réalité dans les outils de planification à court terme. Aussi, en ayant plus de données provenant d'un système de suivi de la flotte d'équipements mobiles de la mine, il serait possible de dresser un meilleur portrait du trafic à utiliser dans les outils de planification. Ces données de trafic seraient plus précises que celles provenant d'un outil de simulation comportant des limitations.

RÉFÉRENCES

- [1] Queen's University Mine Design Wiki, "Sub-level open stoping," 2011. [En ligne]. Disponible : http://minewiki.engineering.queensu.ca/mediawiki/index.php/Sub-level_open_stoping
- [2] R. F. Subtil, D. M. Silva et J. C. Alves, "A practical approach to truck dispatch for open pit mines," dans *35Th APCOM symposium*, 2011, p. 24–30.
- [3] M. Gamache, R. Grimard et P. Cohen, "A shortest-path algorithm for solving the fleet management problem in underground mines," *European journal of operational research*, vol. 166, n^o. 2, p. 497–506, 2005.
- [4] L.-P. Campeau, "Optimisation de la planification à court et moyen terme dans les mines souterraines," Thèse de doctorat, Polytechnique Montréal, 2019.
- [5] L.-P. Campeau et M. Gamache, "Short-term planning optimization model for underground mines," *Computers & Operations Research*, vol. 115, p. 104642, 2020.
- [6] D. Haviland et J. Marshall, "Fundamental behaviours of production traffic in underground mine haulage ramps," *International Journal of Mining Science and Technology*, vol. 25, n^o. 1, p. 7–14, 2015.
- [7] R. Epstein *et al.*, "Optimizing long term planning for underground copper mines," dans *Proc. Copper*, 2003, p. 265–279.
- [8] A. M. Newman et M. Kuchta, "Using aggregation to optimize long-term production planning at an underground mine," *European Journal of Operational Research*, vol. 176, n^o. 2, p. 1205–1218, 2007.
- [9] S. Carpentier, M. Gamache et R. Dimitrakopoulos, "Underground long-term mine production scheduling with integrated geological risk management," *Mining Technology*, vol. 125, n^o. 2, p. 93–102, 2016.
- [10] R. Martinelli, J. Collard et M. Gamache, "Strategic planning of an underground mine with variable cut-off grades," *Optimization and Engineering*, p. 1–47, 2019.
- [11] S. Terblanche et A. Bley, "An improved formulation of the underground mine scheduling optimisation problem when considering selective mining," *ORiON*, vol. 31, n^o. 1, p. 1–16, 2015.
- [12] M. Nehring *et al.*, "Integrated short-and medium-term underground mine production scheduling," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 112, n^o. 5, p. 365–378, 2012.

- [13] D. O'Sullivan et A. Newman, "Optimization-based heuristics for underground mine scheduling," *European Journal of Operational Research*, vol. 241, n^o. 1, p. 248–259, 2015.
- [14] P. Nancel-Penardab *et al.*, "A heuristic approach for scheduling activities with or-precedence at an underground mine," *International Journal of Mining, Reclamation and Environment, DOI*, vol. 10, n^o. 17480930.2020, p. 1734152, 2020.
- [15] M. Nehring, E. Topal et P. Knights, "Dynamic short term production scheduling and machine allocation in underground mining using mathematical programming," *Mining Technology*, vol. 119, n^o. 4, p. 212–220, 2010.
- [16] M. Schulze *et al.*, "Machine scheduling in underground mining : an application in the potash industry," *OR spectrum*, vol. 38, n^o. 2, p. 365–403, 2016.
- [17] M. Åstrand, M. Johansson et J. Greberg, "Underground mine scheduling modelled as a flow shop : a review of relevant work and future challenges," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 118, n^o. 12, p. 1265–1276, 2018.
- [18] M. Åstrand, M. Johansson et A. Zanarini, "Fleet scheduling in underground mines using constraint programming," dans *International conference on the integration of constraint programming, artificial intelligence, and operations research*. Springer, 2018, p. 605–613.
- [19] P. Laborie *et al.*, "Ibm ilog cp optimizer for scheduling," *Constraints*, vol. 23, n^o. 2, p. 210–250, 2018.
- [20] P. Nesbitt *et al.*, "Underground mine scheduling under uncertainty," *European Journal of Operational Research*, vol. 294, n^o. 1, p. 340–352, 2021.
- [21] A. Brickey *et al.*, "Barrick's turquoise ridge gold mine optimizes underground production scheduling operations," *INFORMS Journal on Applied Analytics*, vol. 51, n^o. 2, p. 106–118, 2021.
- [22] Z. Song, M. Rinne et A. van Wageningen, "A review of real-time optimization in underground mining production," *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 113, n^o. 12, p. 889–897, 2013.
- [23] N. Varandas *et al.*, "Underground mine traffic management and optimization : an intelligent application based on a real-time localization system," dans *Sixth International Symposium High Performance Mining (AIMS 2014)*, Aachen, Germany, 2014.
- [24] J. Błażewicz, W. Domschke et E. Pesch, "The job shop scheduling problem : Conventional and new solution techniques," *European journal of operational research*, vol. 93, n^o. 1, p. 1–33, 1996.
- [25] H. Topcuoglu, S. Hariri et M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE transactions on parallel and distributed systems*, vol. 13, n^o. 3, p. 260–274, 2002.

- [26] F. A. Omara et M. M. Arafa, “Genetic algorithms for task scheduling problem,” dans *Foundations of Computational Intelligence Volume 3*. Springer, 2009, p. 479–507.
- [27] A. Arunarani, D. Manjula et V. Sugumaran, “Task scheduling techniques in cloud computing : A literature survey,” *Future Generation Computer Systems*, vol. 91, p. 407–415, 2019.
- [28] S. Hartmann et D. Briskorn, “A survey of variants and extensions of the resource-constrained project scheduling problem,” *European Journal of operational research*, vol. 207, n^o. 1, p. 1–14, 2010.
- [29] G. Weil *et al.*, “Constraint programming for nurse scheduling,” *IEEE Engineering in medicine and biology magazine*, vol. 14, n^o. 4, p. 417–422, 1995.
- [30] S. Bourdais, P. Galinier et G. Pesant, “Hibiscus : A constraint programming application to staff scheduling in health care,” dans *International Conference on Principles and Practice of Constraint Programming*. Springer, 2003, p. 153–167.
- [31] Z. Zhao et X. Li, “Scheduling elective surgeries with sequence-dependent setup times to multiple operating rooms using constraint programming,” *Operations Research for health care*, vol. 3, n^o. 3, p. 160–167, 2014.
- [32] H. N. Van, F. D. Tran et J.-M. Menaud, “Sla-aware virtual resource management for cloud infrastructures,” dans *2009 Ninth IEEE International Conference on Computer and Information Technology*, vol. 1. IEEE, 2009, p. 357–362.
- [33] L. Zhang, Y. Zhuang et W. Zhu, “Constraint programming based virtual cloud resources allocation model,” *International Journal of Hybrid Information Technology*, vol. 6, n^o. 6, p. 333–344, 2013.
- [34] A. Malapert, J.-C. Régim et M. Rezgoui, “Embarrassingly parallel search in constraint programming,” *Journal of Artificial Intelligence Research*, vol. 57, p. 421–464, 2016.
- [35] J. Rodriguez, “A constraint programming model for real-time train scheduling at junctions,” *Transportation Research Part B : Methodological*, vol. 41, n^o. 2, p. 231–245, 2007.
- [36] G.-H. Chen, J.-C. Jong et A. F.-W. Han, “Applying constraint programming and integer programming to solve the crew scheduling problem for railroad systems : Model formulation and a case study,” Rapport technique, 2021.
- [37] I. Harjunkoski et I. E. Grossmann, “Decomposition techniques for multistage scheduling problems using mixed-integer and constraint programming methods,” *Computers & Chemical Engineering*, vol. 26, n^o. 11, p. 1533–1552, 2002.
- [38] J. Bidot *et al.*, “Using constraint programming and simulation for execution monitoring and progressive scheduling,” *IFAC Proceedings Volumes*, vol. 39, n^o. 3, p. 615–620, 2006.

- [39] T. Berthold *et al.*, “A constraint integer programming approach for resource-constrained project scheduling,” dans *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*. Springer, 2010, p. 313–317.
- [40] W. Ugarte, “Constraint programming for the pandemic in peru,” dans *Applied Technologies : Second International Conference, ICAT 2020, Quito, Ecuador, December 2–4, 2020, Proceedings*, vol. 1388. Springer Nature, 2021, p. 299.
- [41] F. Winter et N. Musliu, “Constraint-based scheduling for paint shops in the automotive supply industry,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 12, n^o. 2, p. 1–25, 2021.
- [42] A. Zuenko *et al.*, “An approach to solution of constrained clustering problems using the constraint programming paradigm and the multiset theory,” dans *Journal of Physics : Conference Series*, vol. 1801, n^o. 1. IOP Publishing, 2021, p. 012041.
- [43] N. Çetin, “Open pit truck/shovel haulage system simulation,” 2004.
- [44] A. Jaoua, D. Riopel et M. Gamache, “A simulation framework for real-time fleet management in internal transport systems,” *Simulation Modelling Practice and Theory*, vol. 21, n^o. 1, p. 78–90, 2012.
- [45] P. Chaowasakoo *et al.*, “Digitalization of mine operations : Scenarios to benefit in real-time truck dispatching,” *International Journal of Mining Science and Technology*, vol. 27, n^o. 2, p. 229–236, 2017.
- [46] —, “Improving fleet management in mines : The benefit of heterogeneous match factor,” *European journal of operational research*, vol. 261, n^o. 3, p. 1052–1065, 2017.
- [47] C.-I. . C.-C. F. Saayman, P.*, “Optimization of an autonomous vehicle dispatch system in an underground mine,” *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 106, n^o. 2, p. 77–86, 2006.
- [48] M. M. Fioroni *et al.*, “Logistic evaluation of an underground mine using simulation,” *Rem : Revista Escola de Minas*, vol. 67, n^o. 4, p. 447–454, 2014.
- [49] M. Pasternak et J. A. Marshall, “On the design and selection of vehicle coordination policies for underground mine production ramps,” *International Journal of Mining Science and Technology*, vol. 26, n^o. 4, p. 623–627, 2016.
- [50] J. Greberg *et al.*, “Alternative process flow for underground mining operations : analysis of conceptual transport methods using discrete event simulation,” *Minerals*, vol. 6, n^o. 3, p. 65, 2016.

- [51] A. Salama et J. Greberg, “Optimization of truck-loader haulage system in an underground mine : A simulation approach using simmine,” dans *International Conference & Exhibition on Mass Mining : 10/06/2012-14/06/2012*, 2012.
- [52] G. Nwaila *et al.*, “Algorithmic optimization of an underground witwatersrand-type gold mine plan,” *Natural Resources Research*, vol. 30, n°. 2, p. 1175–1197, 2021.
- [53] P.-J. Erard et P. Déguénon, *Simulation par événements discrets*. PPUR presses polytechniques, 1996.
- [54] T. Altiok et B. Melamed, *Simulation modeling and analysis with Arena*. Elsevier, 2010.
- [55] N. Zupick, “Solving agent based problems with arena,” <https://www.arenasimulation.com/blog/post/solving-agent-based-problems-with-arena>, February 2016.

ANNEXE A MODÈLE DE SIMULATION ARENA

Les captures d'écran suivantes dévoilent les constructeurs logiques Arena utilisés pour la création du simulateur de la mine.

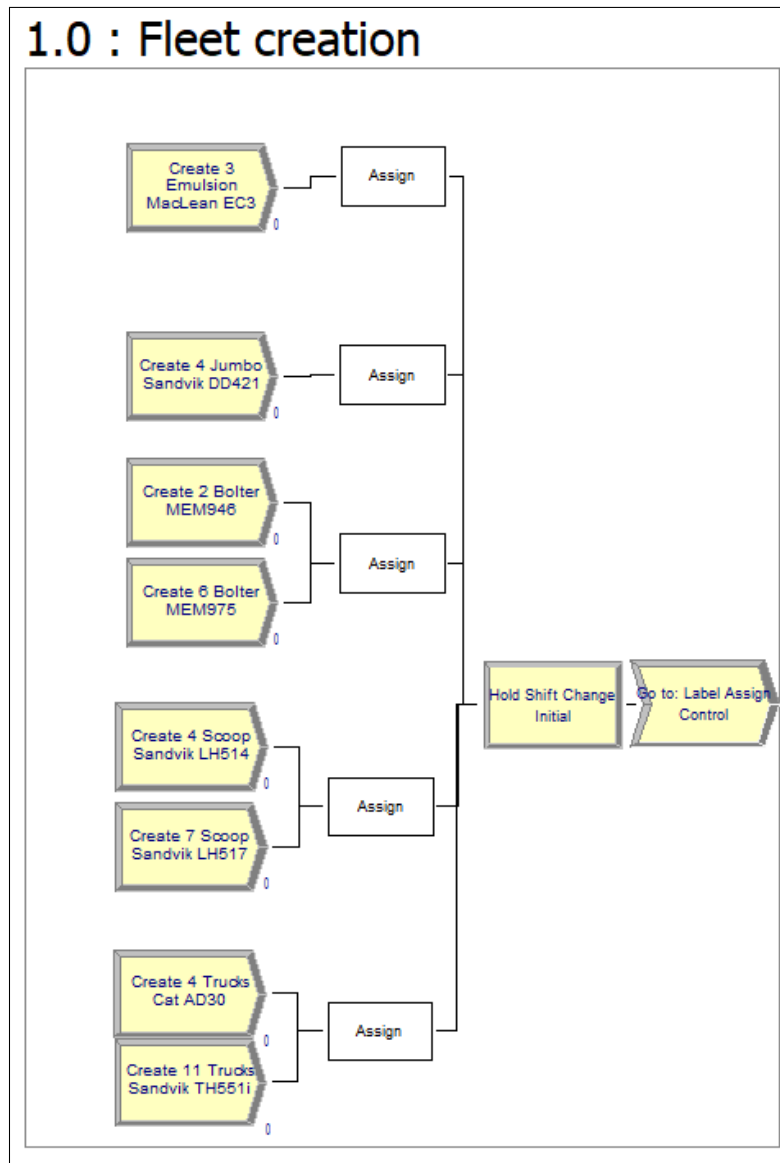


Figure A.1 Section 1.0 : Création des entités de la flotte.

1.1 : Assigning control

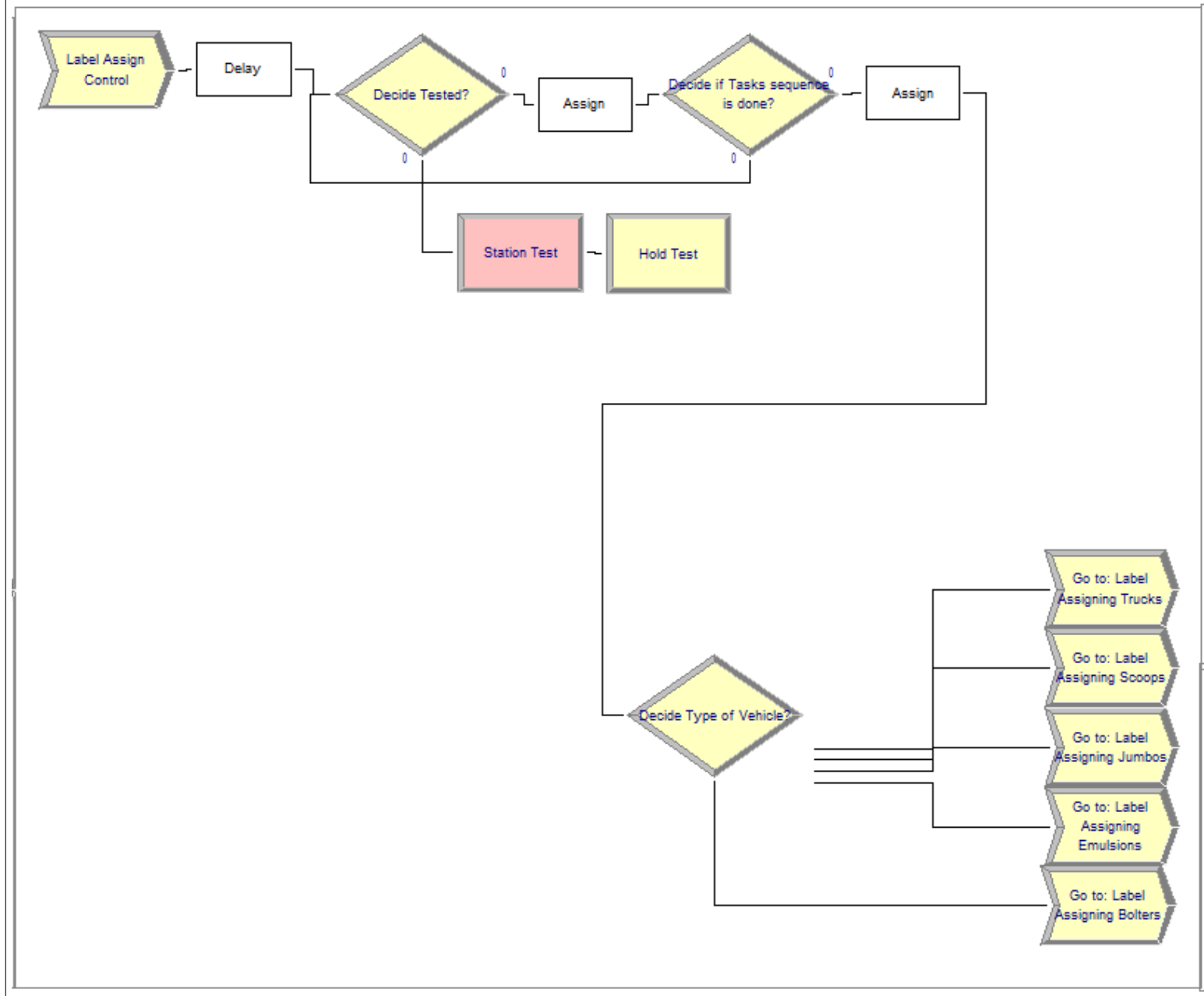
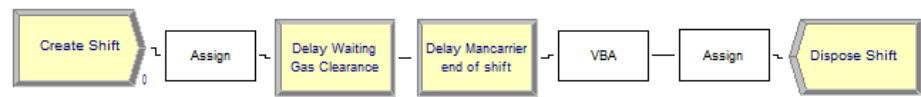


Figure A.2 Section 1.1 : Comptabilisation du nombre de tâches par entité.

1.2 : Shift change



1.3 : Excel Outputs

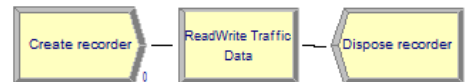


Figure A.3 Section 1.2 et 1.3 : Création des entités quart de travail et des entités d'enregistrement des résultats en format CSV.

2.0 : Assigning Scoops

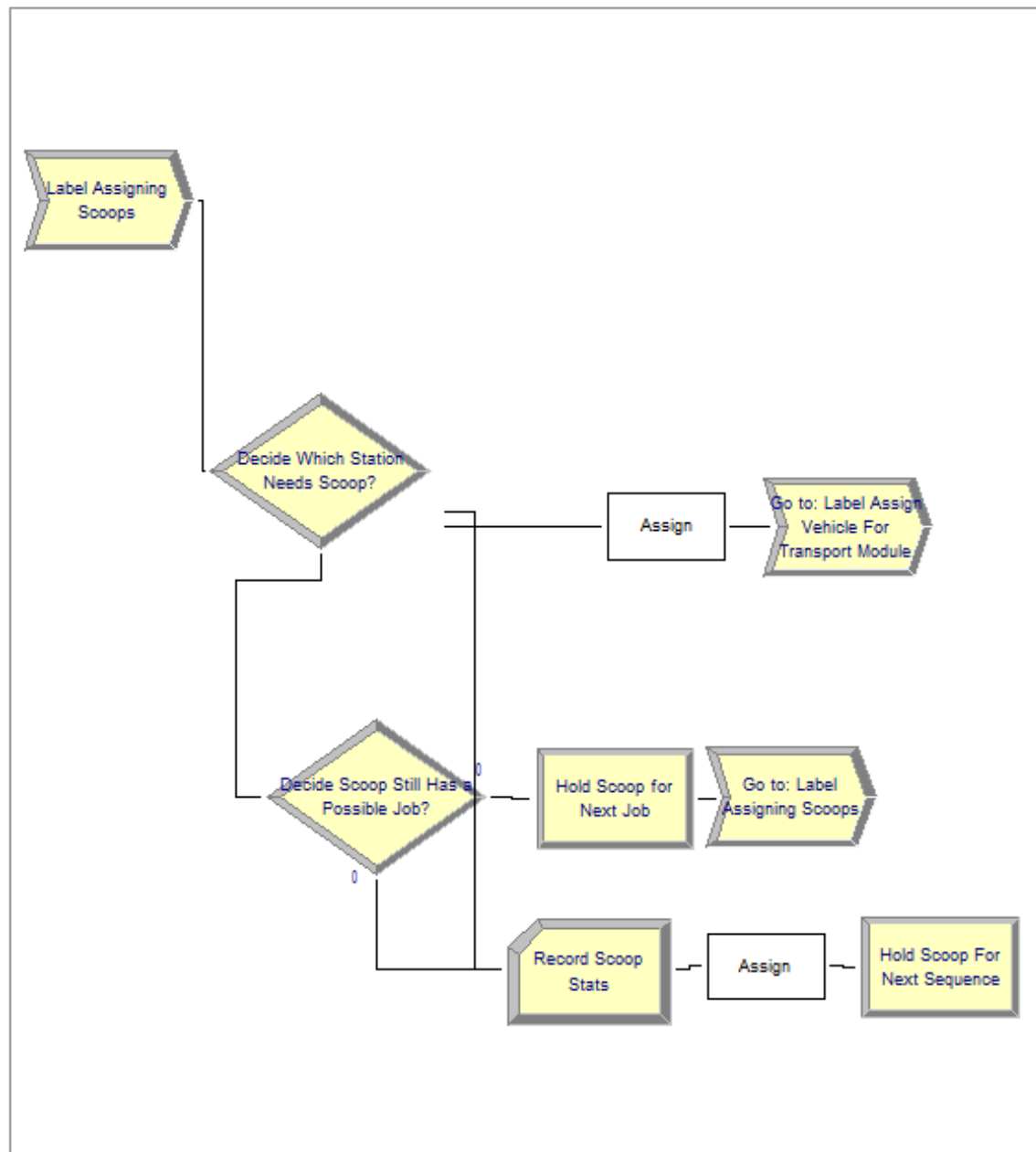


Figure A.4 Section 2.0 : Attribution des tâches pour les chargeuses.

2.1 : Assigning Bolter

2

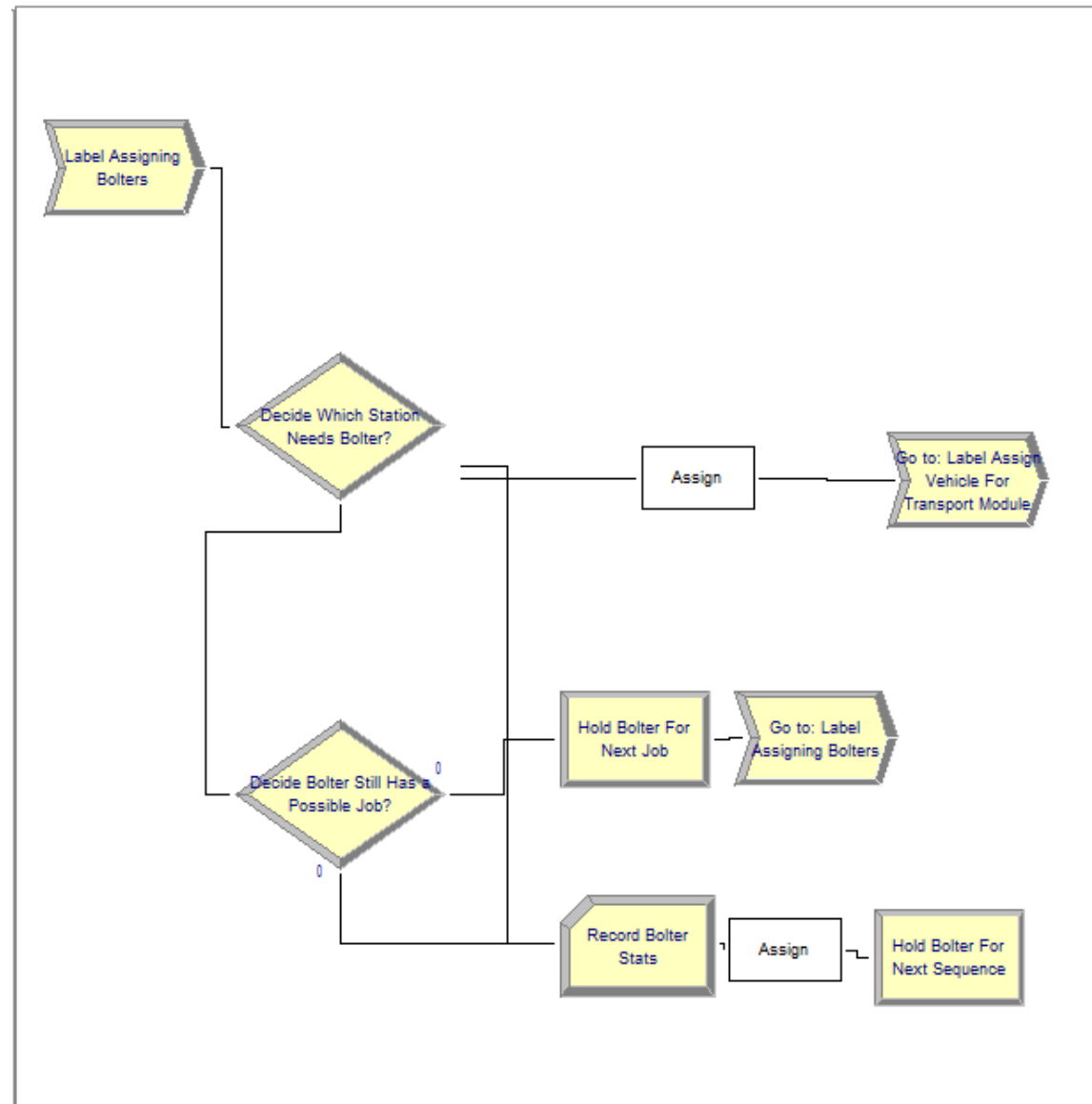


Figure A.5 Section 2.1 : Attribution des tâches pour les boulonneuses.

2.2 : Assigning Jumbo

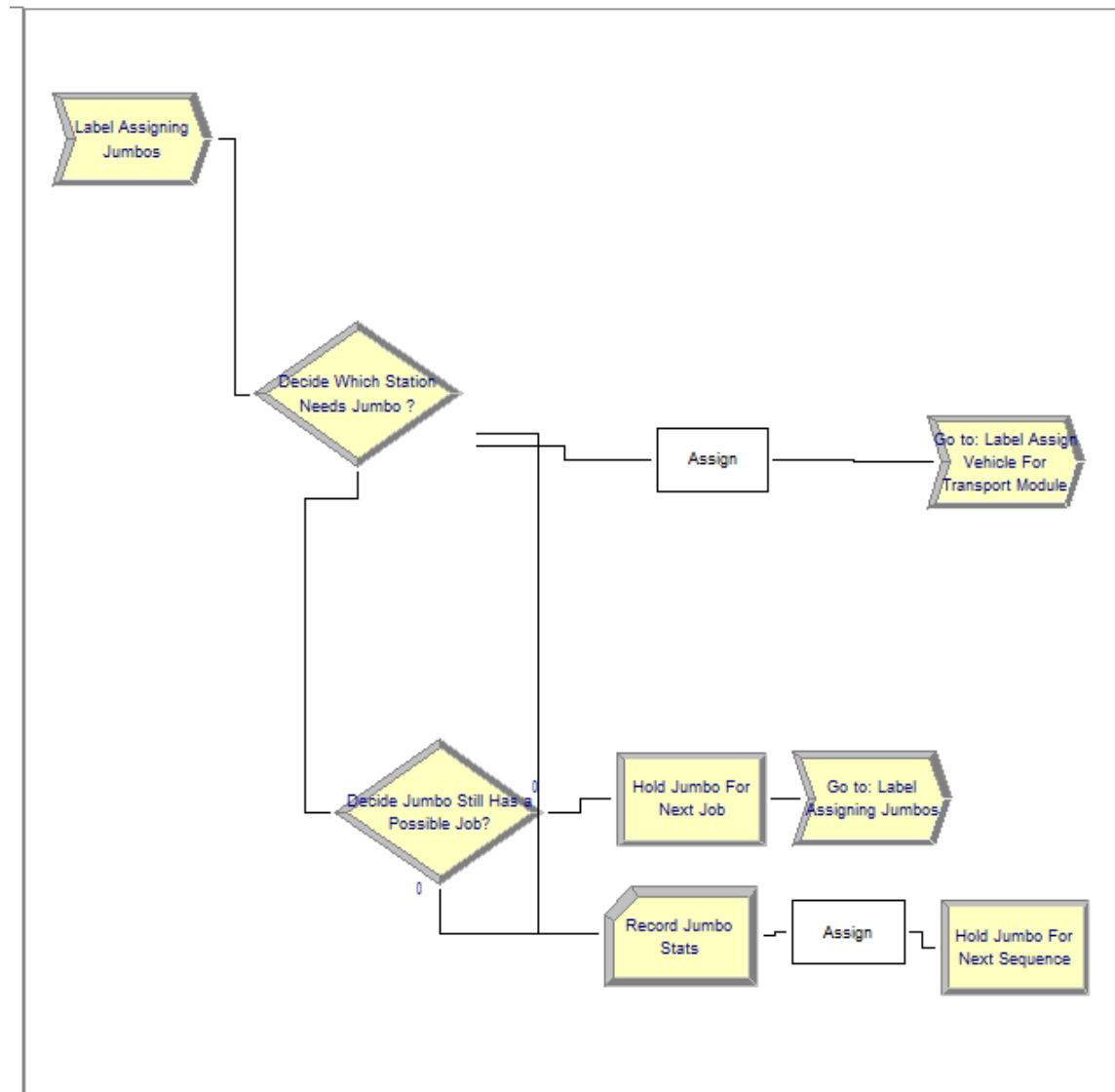


Figure A.6 Section 2.2 : Attribution des tâches pour les *jumbo*.

2.3 : Assigning Emulsion

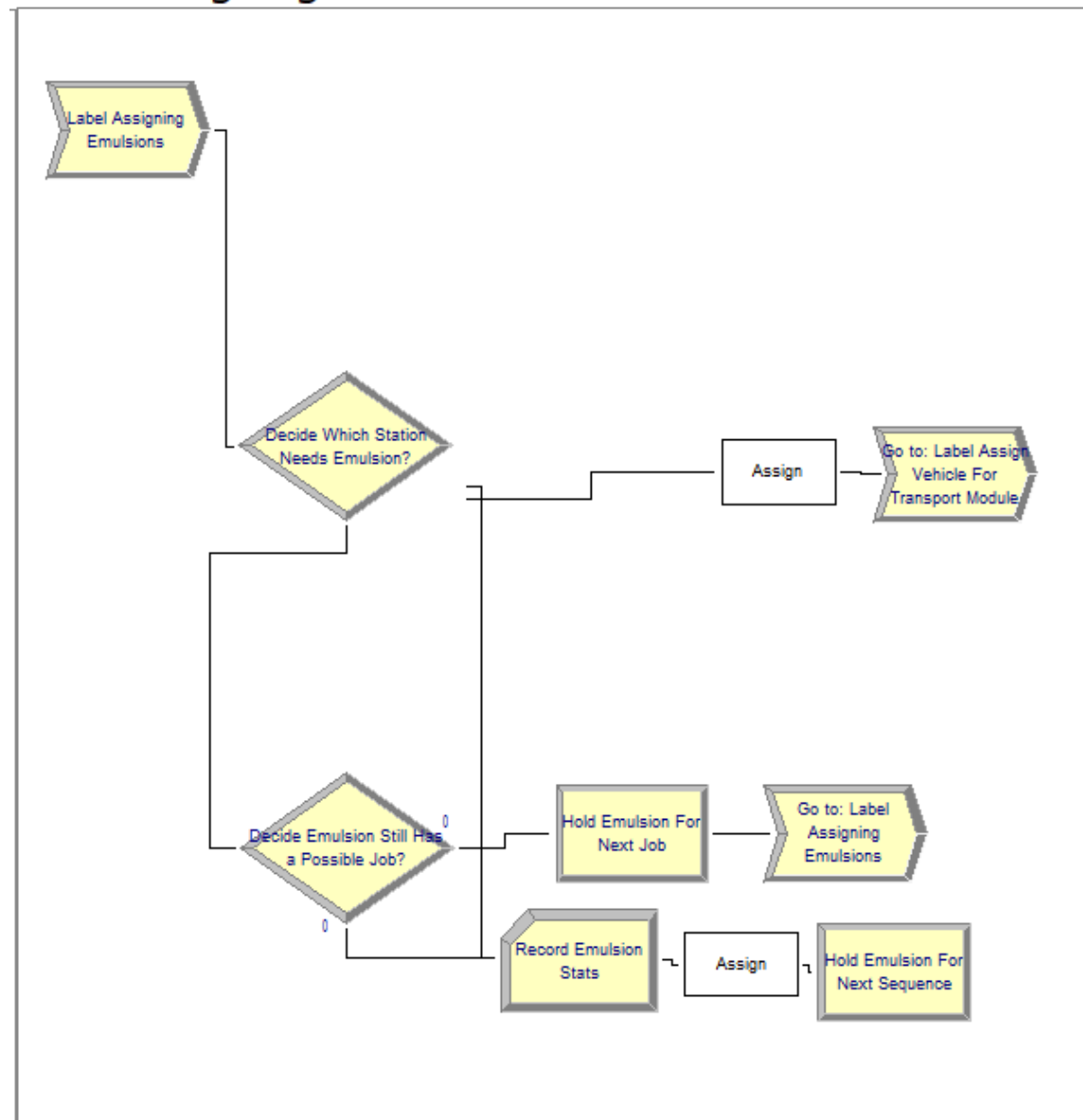


Figure A.7 Section 2.3 : Attribution des tâches pour les camions à émulsion.

2.4 : Assigning Trucks

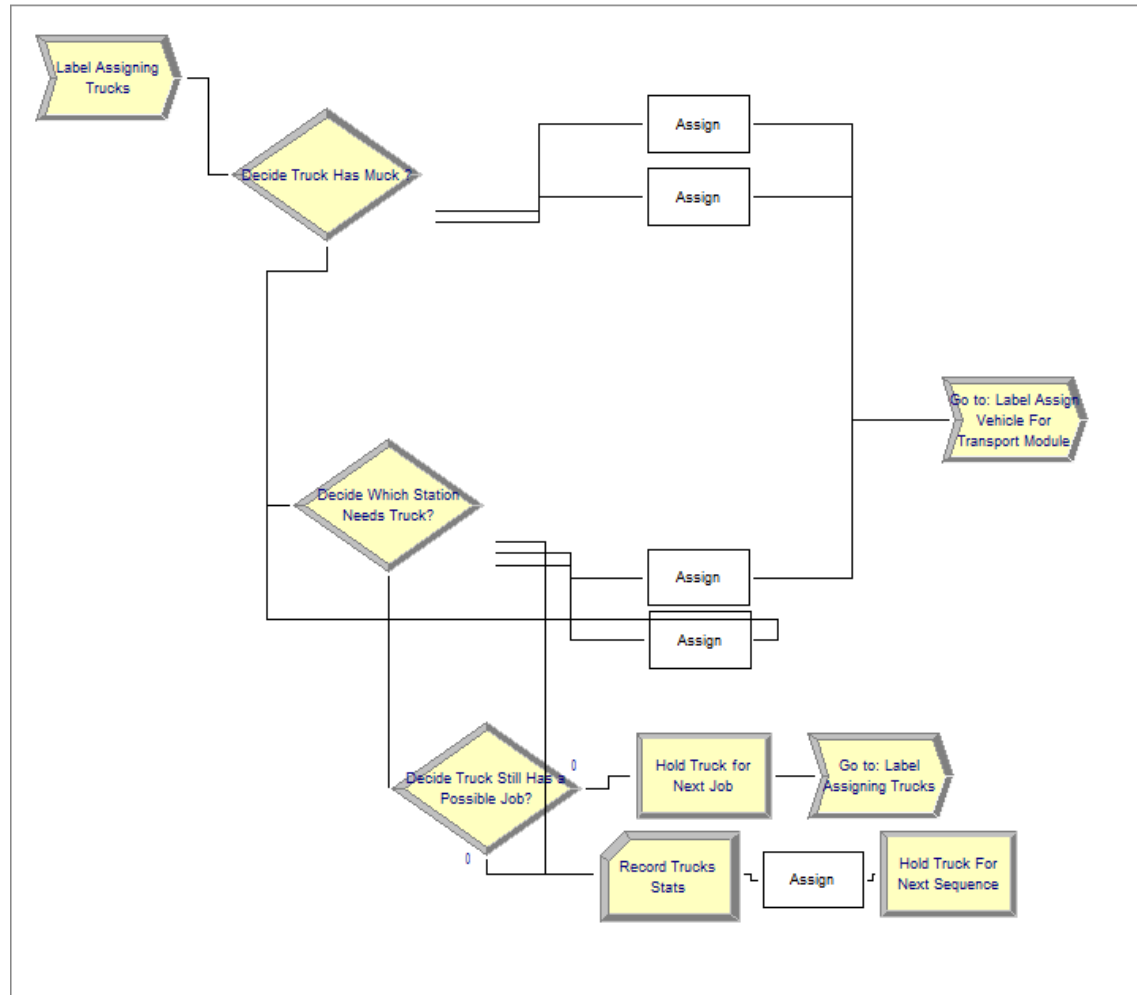


Figure A.8 Section 2.4 : Attribution des tâches pour les camions.

3.0 : Initiate transportation logics

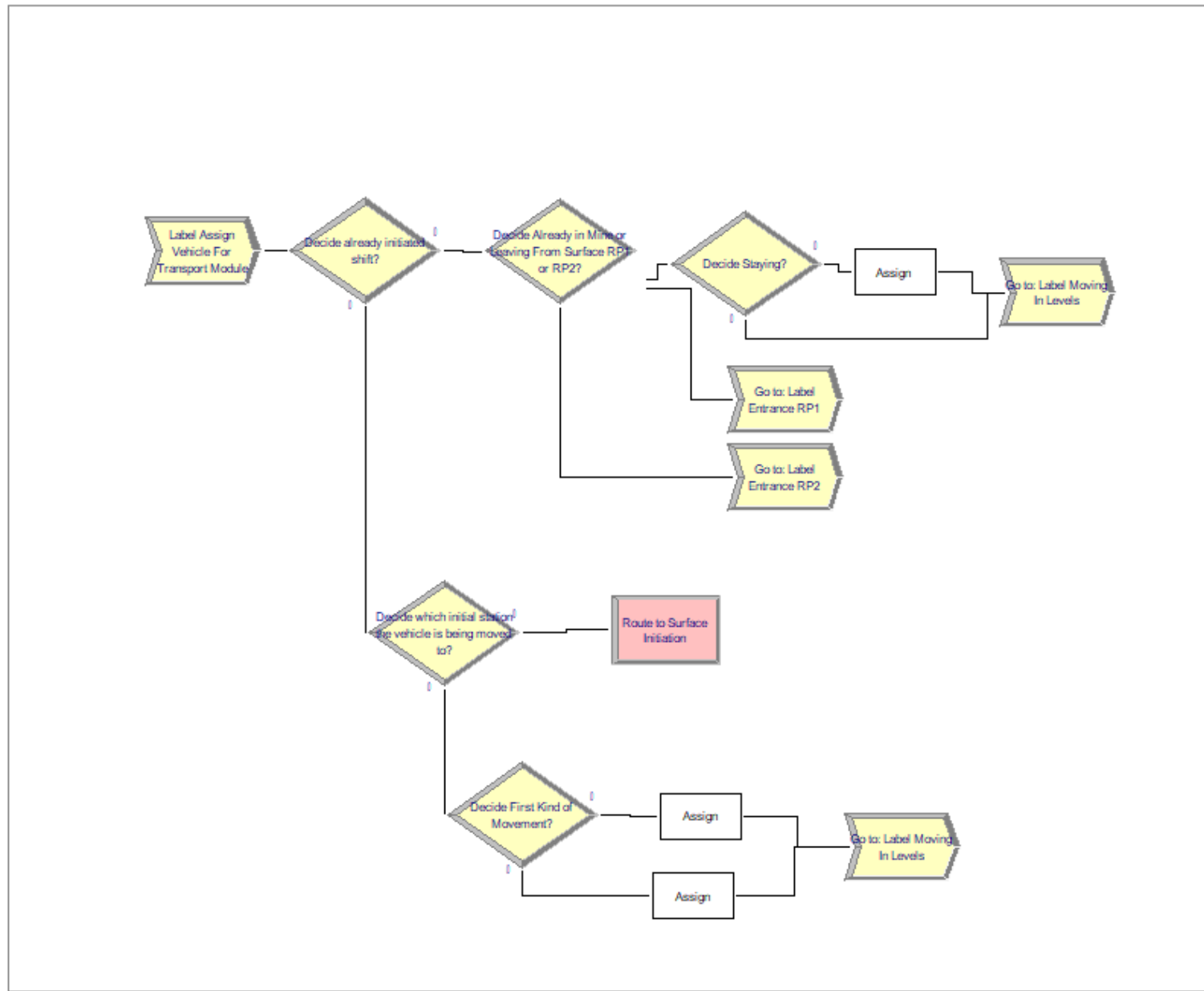


Figure A.9 Section 3.0 : Aiguillage des entités, qui ont une nouvelle tâche, vers la section de déplacement appropriée.

3.1 : Entering Ramps / Entering levels

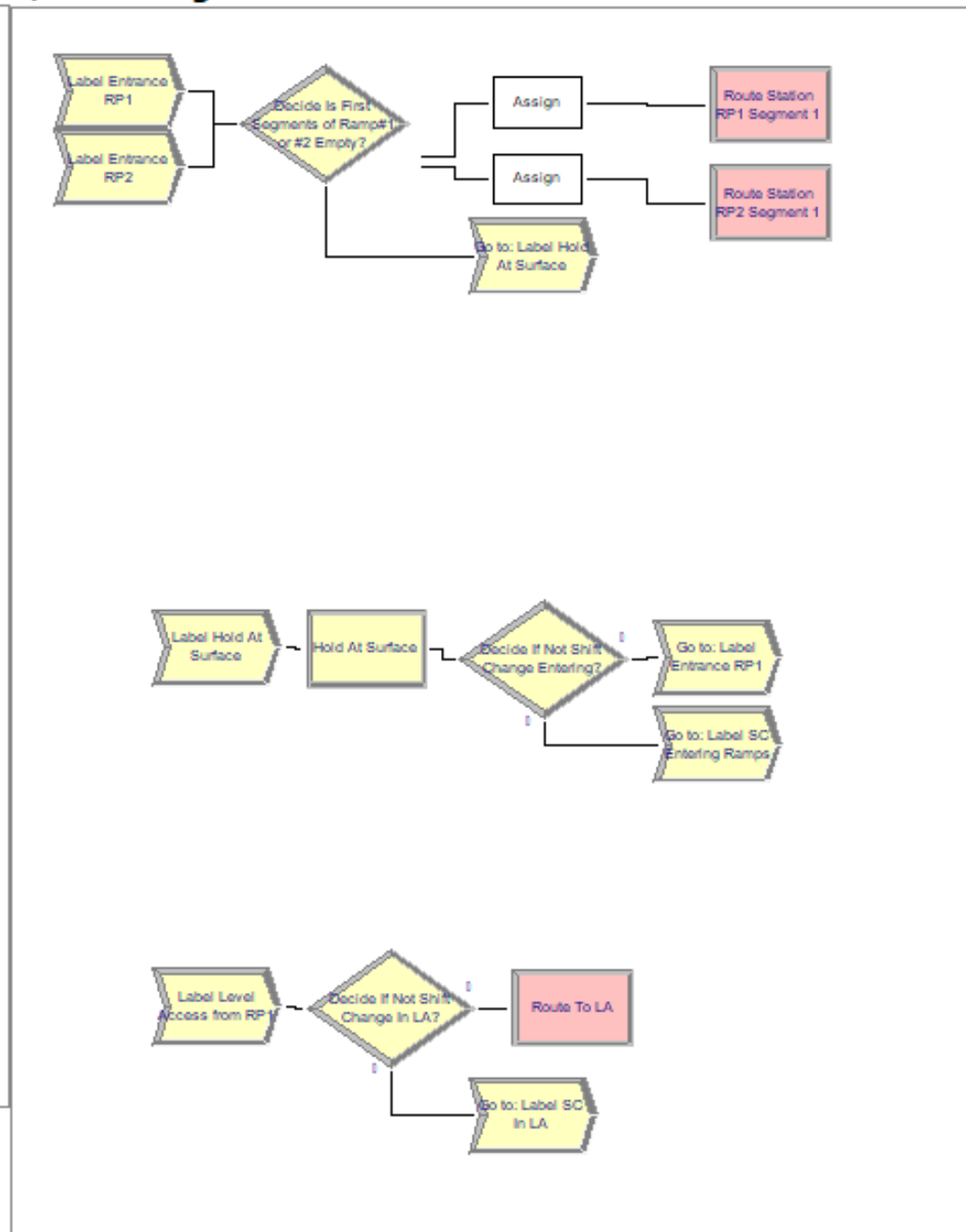


Figure A.10 Section 3.1 : Déplacements d'entrée de niveau ou de rampe.

3.2 : Moving In Ramp #2 (RP2)

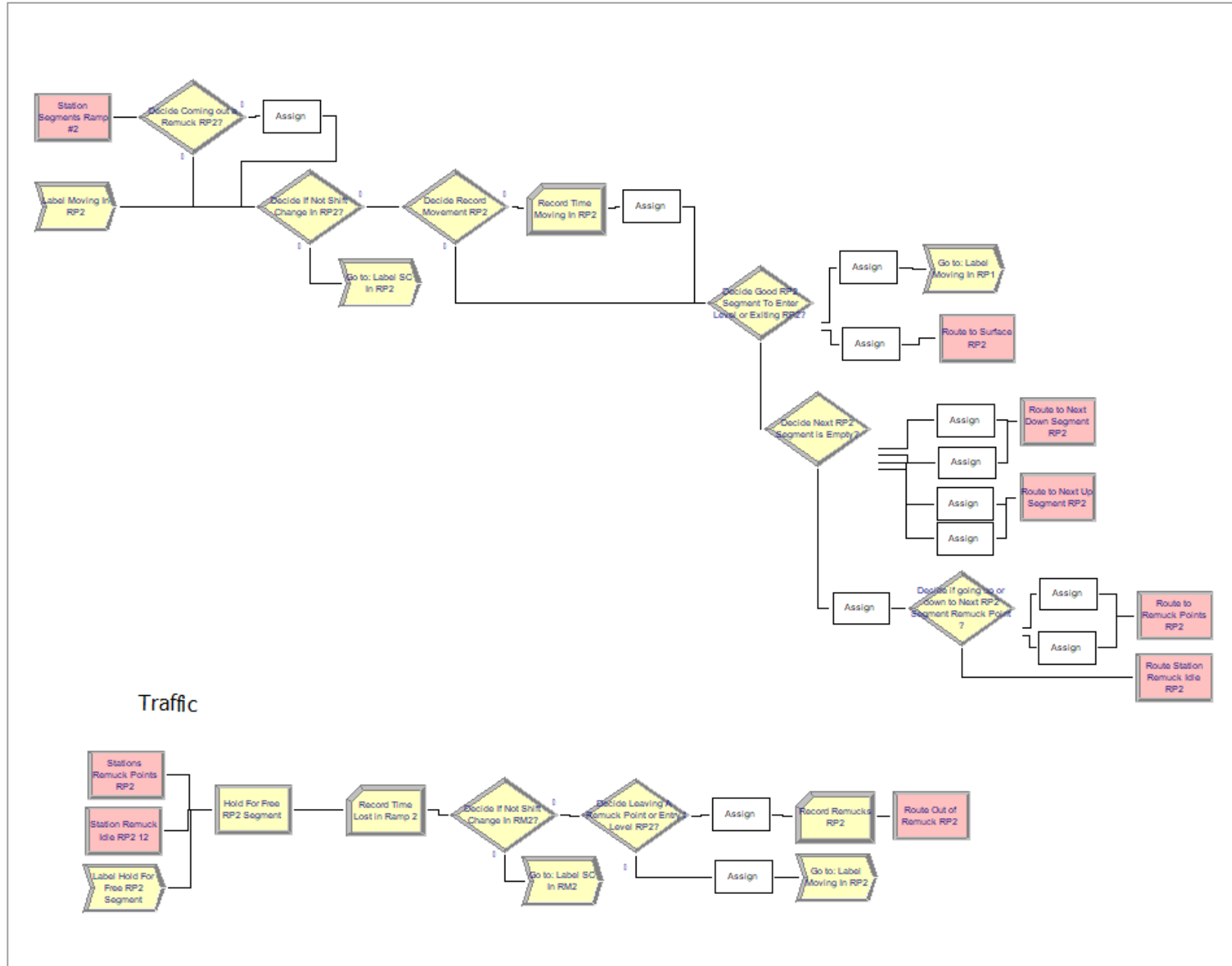


Figure A.11 Section 3.2 : Déplacements dans la rampe 2.

3.3 : Moving In Ramp #1 (RP1)

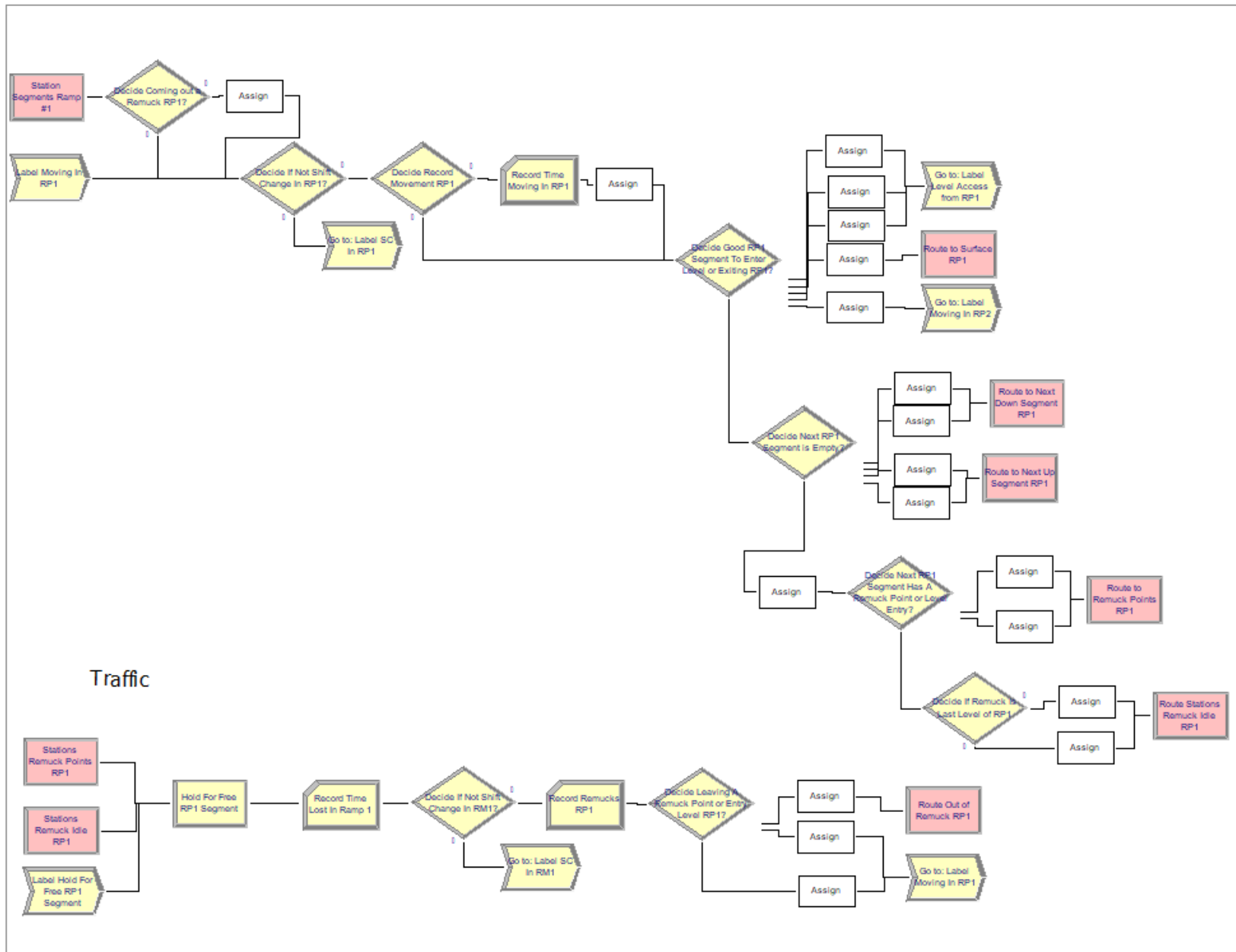


Figure A.12 Section 3.3 : Déplacements dans la rampe 1.

3.4 : Moving in Levels

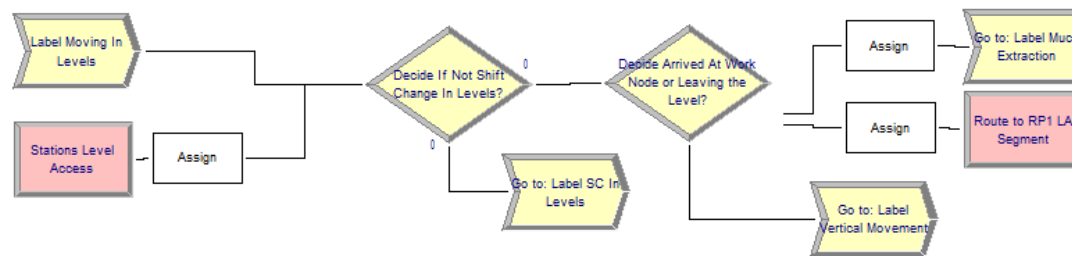


Figure A.13 Section 3.4 Partie 1 : Aiguillage des déplacements dans les niveaux.

Horizontal movement

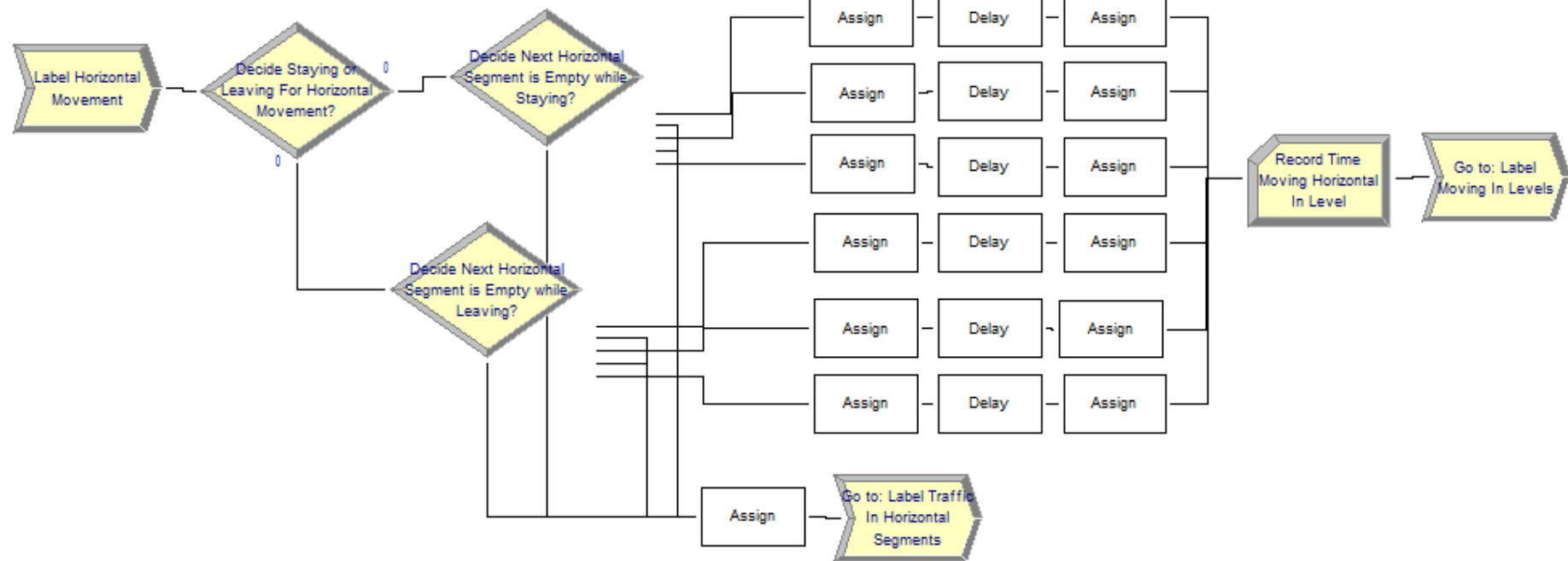


Figure A.15 Section 3.4 Partie 3 : Déplacements horizontaux dans les niveaux.

Traffic

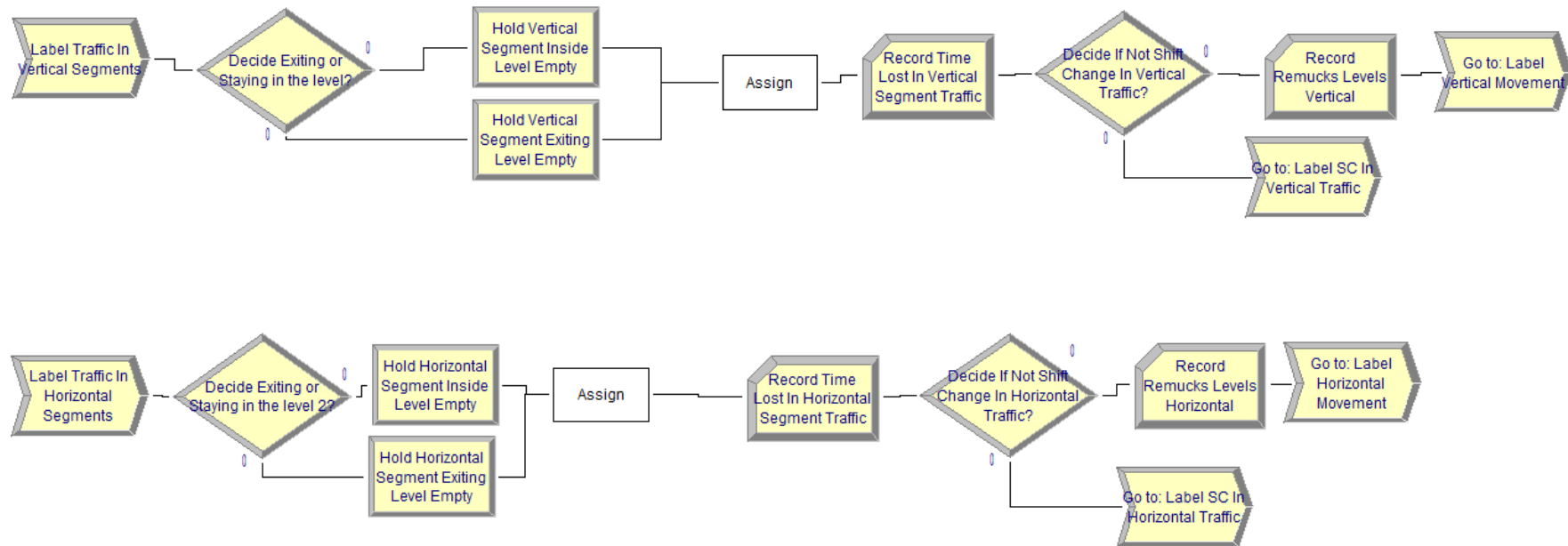


Figure A.16 Section 3.4 Partie 4 : Laissez-passer associé au trafic dans les niveaux.

4.0 : Muck extraction

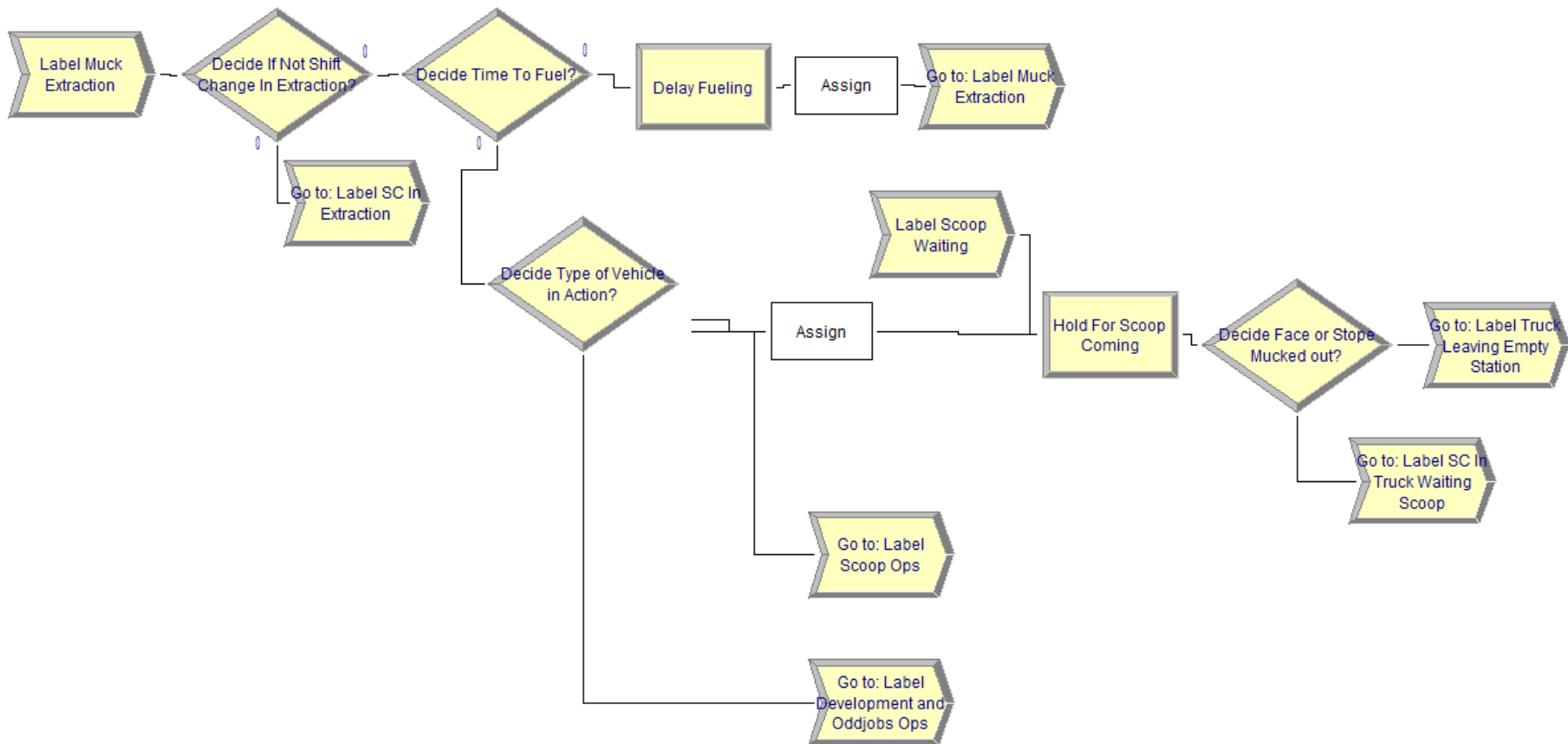


Figure A.17 Section 4.0 Partie 1 : Aiguillage des entités vers les logiques d'opérations appropriées.

Development/Backfill Operations

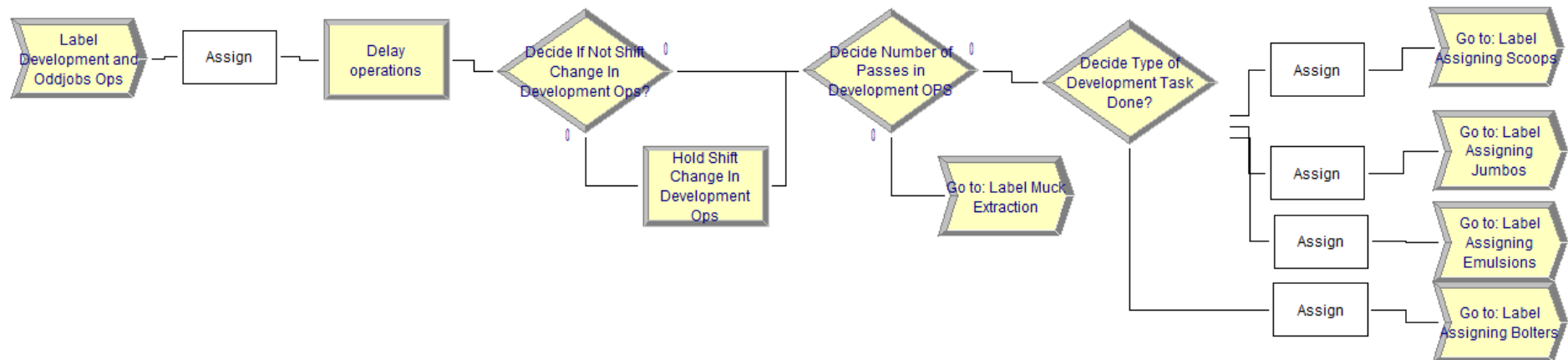


Figure A.18 Section 4.0 Partie 2 : Toutes les opérations de développement et de remblayage sauf le soutirage.

Mucking

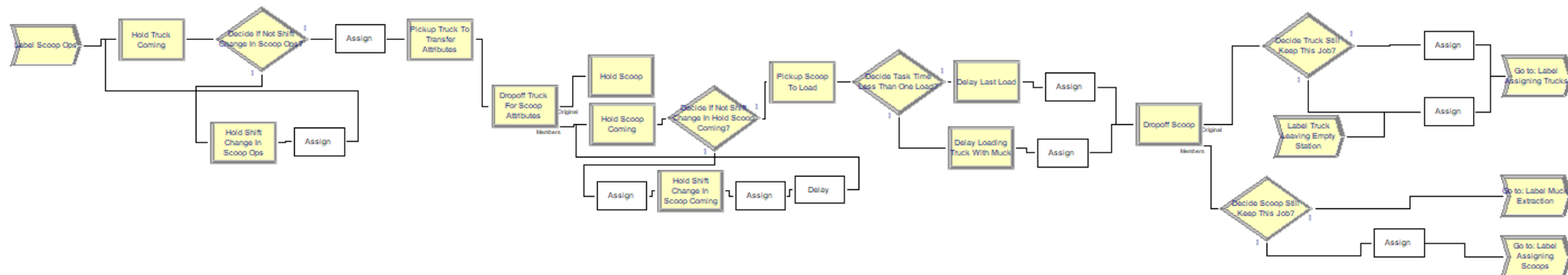


Figure A.19 Section 4.0 Partie 3 : Opérations de soutirage pour le développement et la production.

4.1 : Unloading On the Surface

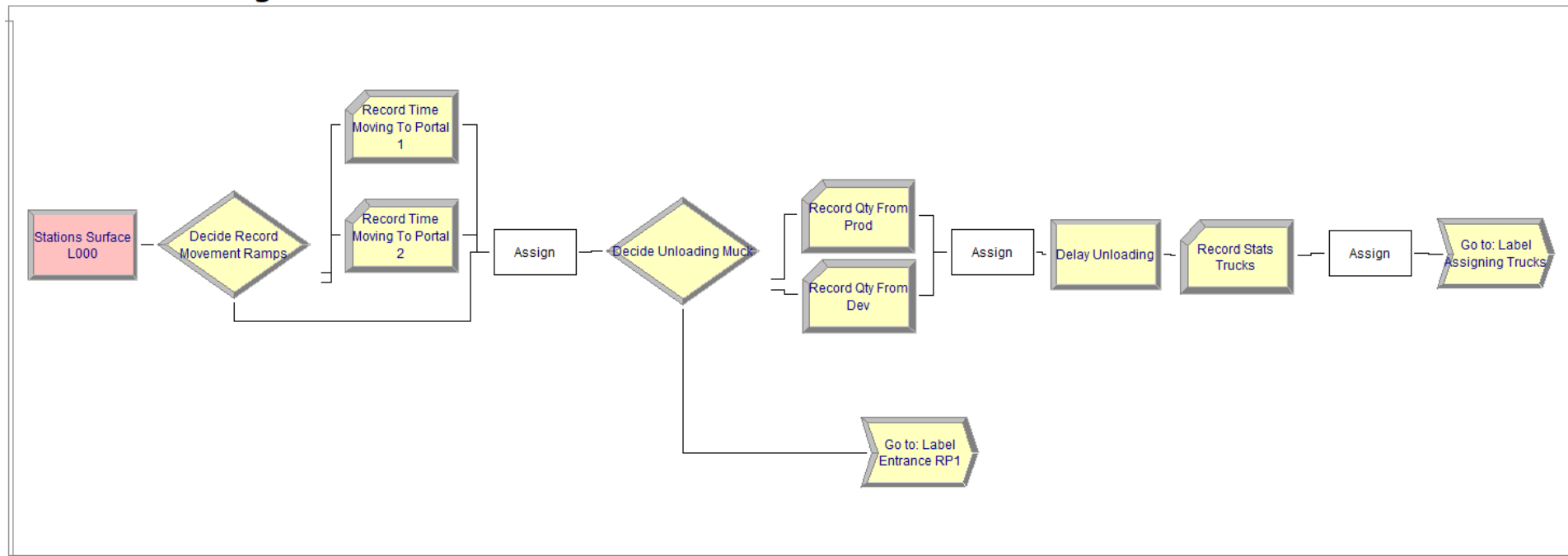


Figure A.20 Section 4.1 : Opérations de déchargement de la roche à la surface.

5.0 : Shift Change Labels

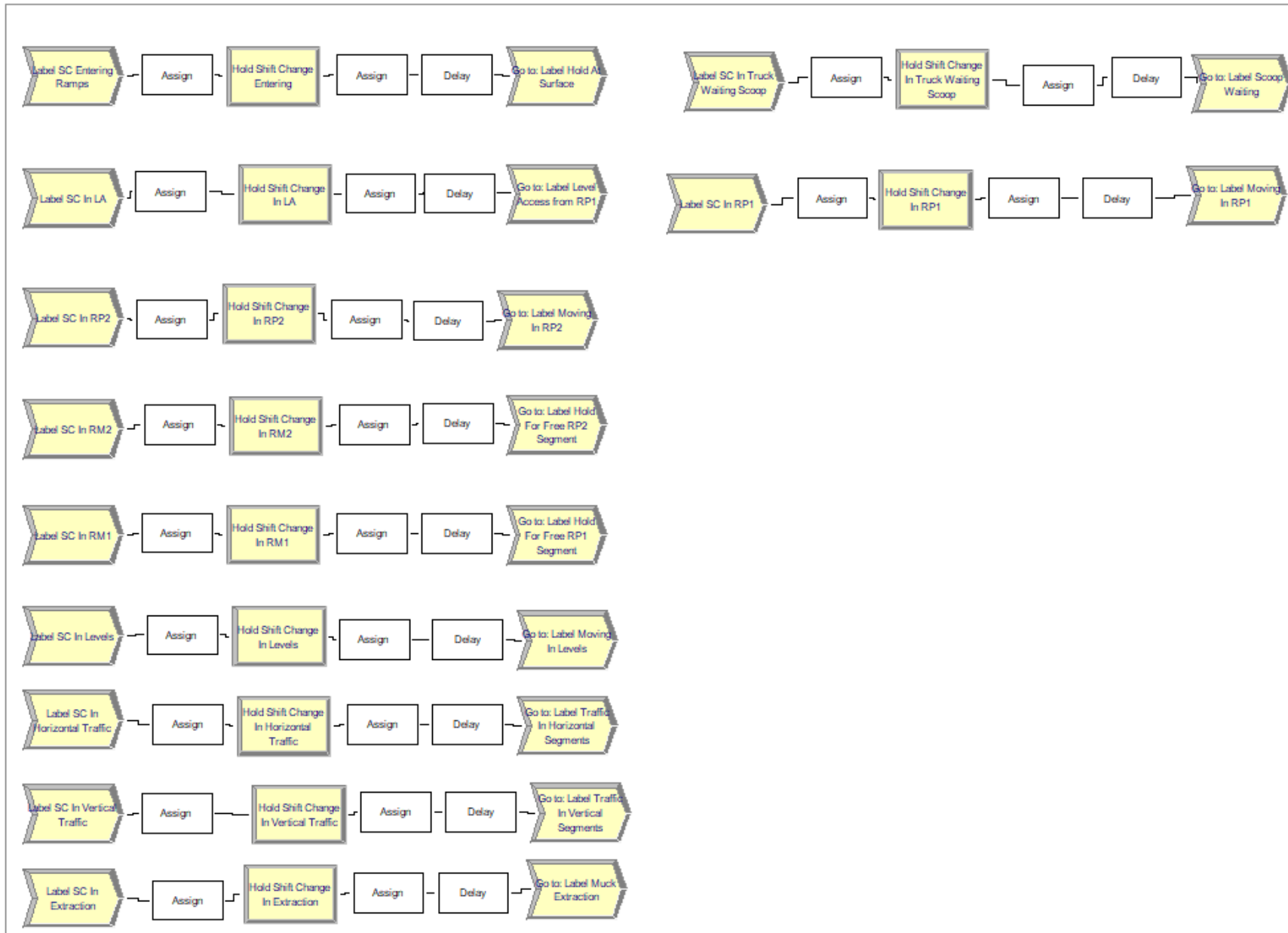


Figure A.21 Section 5.0 : Files d'attentes lors des changements de quart.

Date in Simulation	Time of the Day	Gas Clearance Occuring	Dev Qty (tons)	Prod Qty (tons)
15 January, 2021	05:30:00	0.00	0.00	0.00
			Qty Loads	Qty Blasts
			0.00	0.00

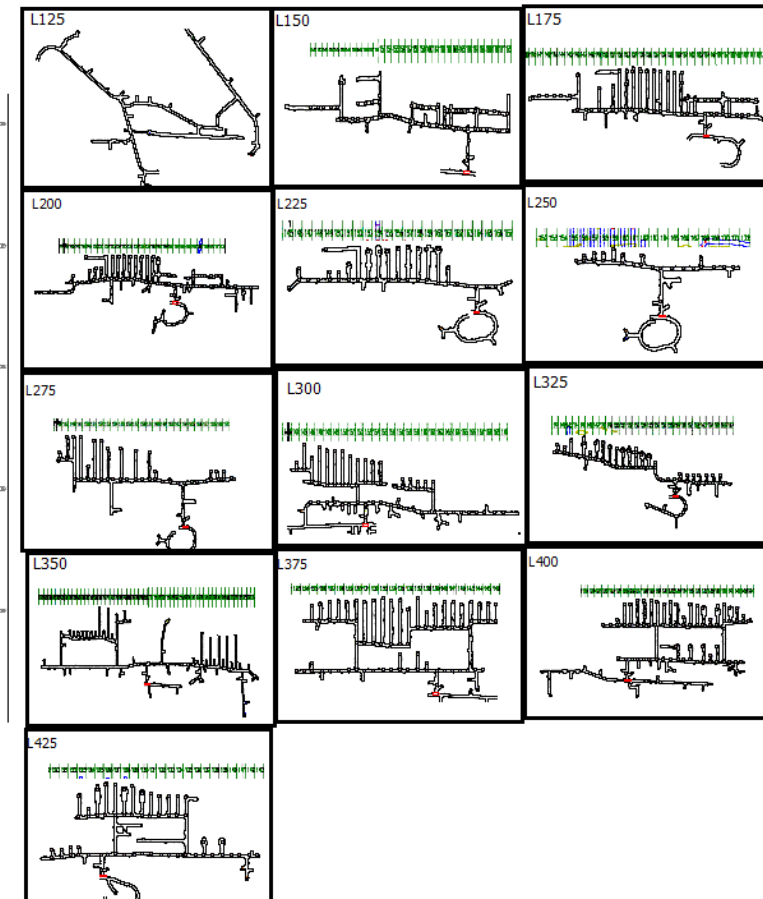
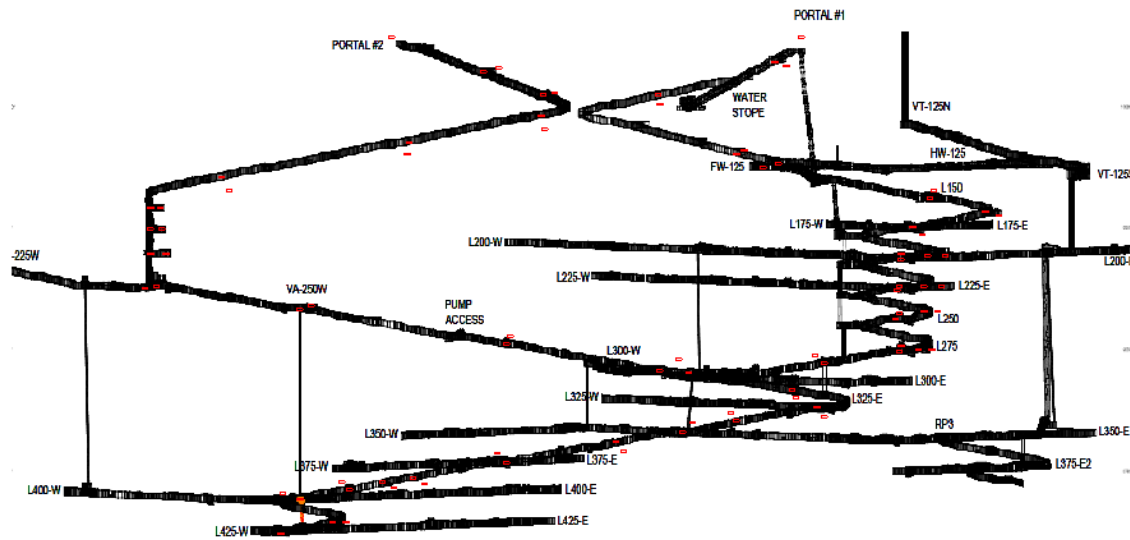


Figure A.22 Tableau de bord pour les simulations.

ANNEXE B MODÈLES DE PLANIFICATION

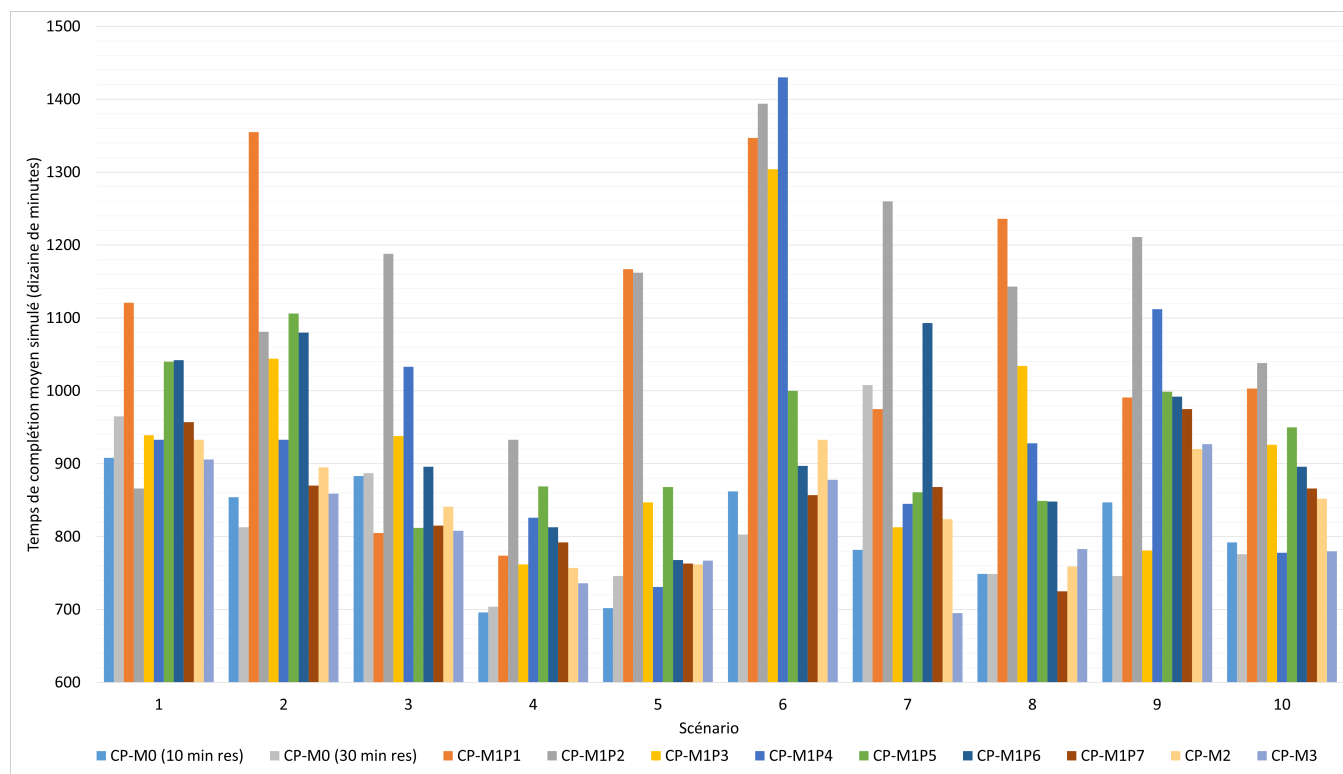


Figure B.1 Comparaison des temps de complétion moyens simulés provenant de tous les modèles.