

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE LYON

UNIVERSIDAD DE LOS ANDES

THÈSE DOCTORALE

**Optimisation d'Alignements d'un Réseau de Pipelines basée sur
les Algèbres Tropicales et les Approches Génétiques**

Mots clés : Modélisation des systèmes, optimisation multi-objectif, algèbres tropicales,
réseaux de flux, systèmes à événements discrets.

Par :

Karla QUINTERO

Pour obtenir

LE GRADE DE DOCTEUR

Jury

M. Eric NIEL	Directeur	Professeur (INSA Lyon)
M. José AGUILAR	Directeur	Professeur (ULA Mérida, Venezuela)
M. Philippe JEANNIN	Tuteur Industriel	Ing. Développement (Thales Group - TCS)
M. Sébastien LAHAYE	Rapporteur	Professeur (Université d'Angers)
M. Hassane ALLA	Rapporteur	Professeur (Université de Grenoble)
Mme. Audine SUBIAS	Examineur	MCF HDR (INSA Toulouse)
M. Addison RIOS	Examineur	Professeur (ULA Mérida, Venezuela)
M. Juan CARDILLO	Examineur	Professeur (ULA Mérida, Venezuela)
Mme. Juliette MATTIOLI	Examineur	Resp. Laboratoire de Recherche (Thales Group - TRT)

Laboratoire Ampère (UMR5005) - INSA Lyon
Laboratoire CEMISID - ULA Mérida, Venezuela

3 avril 2015

Résumé

Cette thèse porte sur l'optimisation d'opérations dans un terminal maritime pétrolier en vue d'assister les opérateurs de supervision. L'objectif est de fournir des solutions candidates pour la sélection d'alignements (chemins) de pipelines et pour l'ordonnancement d'opérations de transfert de pétrole et de maintenance de vannes. La principale difficulté de ce travail est la gestion d'un réseau à ressources limitées et conflictuelles. La prise de décision doit être réalisée en fonction des disponibilités des dispositifs, de la capacité opérative du réseau, d'aspects financiers (pénalités) liés au service, et d'activités de maintenance planifiées au préalable.

L'optimisation est abordée par des approches tropicales du fait que ces techniques appréhendent de manière concise et intuitive les phénomènes de synchronisation. Les propositions développées ici commencent par des modèles d'optimisation algébriques mono-objectif et se complexifient, lors de l'intégration de nouvelles variabilités, jusqu'à la proposition de modèles d'optimisation multi-objectif hybrides par approches d'intelligence artificielle et algébriques. Dans un premier temps, un modèle d'optimisation mono-objectif est proposé en algèbre $(\max, +)$ pour minimiser les pénalités, intégrant des phénomènes de nature différente dans une contrainte générique unique. Ce modèle est non linéaire, considère des alignements définis au préalable, et sa validation est réalisée à l'aide du logiciel d'optimisation LINGO.

Dans un deuxième temps, la linéarisation est résolue par une priorisation des opérations en conflit. Dans ce contexte, deux critères de linéarisation sont considérés : le premier concernant les pénalités potentielles pour les clients, et le deuxième concernant la criticité des opérations. Le modèle $(\max, +)$ non linéaire minimisant les pénalités est étendu pour la prise en compte de la recherche d'alignements et de la minimisation des retards des opérations de maintenance. Pour appréhender la dimension multi-objectif, une approche hybride d'algorithmes génétiques et des systèmes $(\max, +)$ linéaires est ensuite discutée. Enfin, dans un cadre plus formel, un nouveau produit synchrone exploitant les phénomènes de parallélisme au plus tôt est défini pour des automates tropicaux pour minimiser le makespan.

Les propositions sont validées par des données industrielles recueillies auprès de l'entreprise pétrolière PDVSA et du fournisseur de solutions de supervision Thales Group. Les principales

contributions à la recherche relèvent de la considération des approches tropicales dans la résolution de la problématique d'optimisation conduisant à des modèles concis et potentiellement linéaires, la proposition d'une approche hybride d'algorithmes génétiques et systèmes $(\max,+)$ linéaires exploitant les avantages de recherche distribuée des approches de l'intelligence artificielle avec les modèles concis issus de l'algèbre $(\max,+)$, et de la définition d'un nouveau produit synchrone d'automates tropicaux par une exploitation des phénomènes de parallélisme pour un comportement au plus tôt.

Abstract

This thesis addresses operations optimization in an oil seaport with the fundamental purpose of assisting supervision operators. The objective is to provide candidate solutions for pipeline alignment (path) selection and for scheduling of oil transfer operations, as well as maintenance operations, considering that the system has limited and conflicting resources. Informed decision making should consider operations scheduling and alignment selection based on : devices availability, operative capacity of the network, financial aspects (penalties) due to late service, and a predefined maintenance schedule.

The optimization problem is addressed by tropical approaches given their potential for yielding concise and intuitive representations when modeling synchronization phenomena. The proposals developed herein start by algebraic mono-objective optimization models. They subsequently become more complex, as new aspects are included, leading to the formulation of hybrid multi-objective optimization models based on artificial intelligence approaches as well as $(\max,+)$ -linear system theory.

Firstly, a mono-objective optimization model is proposed in $(\max,+)$ algebra for penalty minimization. This model integrates different nature phenomena into one single constraint. It is nonlinear, considers predefined alignments for transfer operations and is validated through an optimization solver (LINGO). Secondly, linearization of such model is introduced for prioritization of conflicting operations. Within this context, 2 criteria are addressed : potential penalties for clients and, on the other hand, operations criticality. The nonlinear $(\max,+)$ model minimizing penalties is extended in order to consider alignment search and delay minimization for maintenance operations. In order to address the multi-objective nature of the problem, an approach based on genetic algorithms and $(\max,+)$ -linear system theory is proposed.

Finally, in a more formal framework, a new synchronous product for tropical automata exploiting parallelism phenomena at the earliest is defined in order to minimize the makespan. The proposed models and methods herein have been validated by industrial data gathered from the oil company PDVSA and from the supervision solutions provider Thales Group.

The main contributions of this research are, firstly, the application of tropical approaches to this specific optimization problem, yielding concise and potentially linear models. Secondly, the proposal of a hybrid approach based on genetic algorithms and $(\max, +)$ -linear systems, which exploits the advantages of the distributed search of artificial intelligence approaches and the conciseness of the models stemming from $(\max, +)$ algebra. The final contribution focuses on the definition of the alphabet for a new synchronous product of tropical automata.

Remerciements

Table des matières

Résumé	i
Abstract	iii
Remerciements	v
1 Introduction	1
1.1 Contexte de la Recherche	2
1.1.1 Cadre de Collaboration	2
1.1.2 Cas d'Étude	2
1.2 Motivation et Objectifs	5
1.3 État de l'Art	6
2 Algèbres Tropicales et Modèles d'Optimisation Mono-objectif	21
2.1 Algèbres Tropicales - Motivation et Cadre Théorique	23
2.2 Modélisation du Cas d'Étude	25
2.2.1 Fonctionnement du Système	25
2.2.1.1 Gestion des Pénalités	25
2.2.1.2 Partage de Ressources	26
2.2.2 Modèle Non Linéaire d'Optimisation de l'Ordonnancement d'Opérations .	29
2.2.2.1 Modèle Non Linéaire	30
2.2.2.2 Résultats	34
2.2.3 Exploration des Modèles Linéaires d'Optimisation de l'Ordonnancement d'Opérations	37
2.2.3.1 Modèle Linéaire basé sur la Pénalité Totale Potentielle	37
2.2.3.2 Modèle Linéaire basé sur la Criticité	42
2.2.4 Modèle d'Optimisation de l'Ordonnancement d'Opérations et de Sélection d'Alignements	47
3 Modèle d'Optimisation Multi-objectif pour l'Ordonnancement d'Opérations	53
3.1 Cadre Général de la Problématique d'Optimisation Multi-Objectif	54

3.2	Optimisation Multi-Objectif et Algorithmes Génétiques - Notions Théoriques . . .	56
3.3	Formulation du Problème d'Optimisation Multi-Objectif sous une Structure d'Al- gorithmes Génétiques	58
3.3.1	Structure Génomique	60
3.3.1.1	Vue d'Ensemble sur les Structures Génomiques	60
3.3.1.2	Structure Génomique Proposée	62
3.3.2	Opérateur de Mutation	67
3.3.3	Opérateur de Croisement	69
3.4	Stratégie Évolutionnaire et Intégration des Modèles Algébriques (Max,+)	72
3.5	Résultats	79
4	Modélisation à Base d'Automates Tropicaux pour l'Optimisation de l'Or- donnement d'Opérations	83
4.1	Automates (Min,+) et (Max,+) : Notions Théoriques	85
4.2	Modélisation des Conflits	89
4.3	Bases Théoriques pour l'Alphabet Proposé	91
4.4	Alphabet du Nouveau Produit d'Automates Tropicaux	97
4.4.1	Alphabet pour une Paire d'Automates Synchronisés dans un Système à 3 Automates ($n = 3$)	99
4.4.2	Alphabet pour Trois Paires d'Automates en Synchronisation dans un Sys- tème à 3 Automates ($n = 3$)	103
4.4.3	Alphabet pour un Système à 3 Automates ($n = 3$) avec Tierce Synchronisée	106
4.4.4	Exemple pour un Système à 3 Automates ($n = 3$)	106
4.4.5	Extensions Envisagées pour $n > 3$	110
5	Conclusions et Perspectives	115
5.1	Conclusion Générale	115
5.2	Perspectives	117
	Glossaire	118
	Bibliographie	121

Table des figures

1.1	Schématisation d'un terminal maritime pétrolier (a) et son modèle sous forme de graphe non orienté (b).	3
1.2	Éléments identifiables dans le problème du plus court chemin à travers d'une représentation $(min, +)$.	12
1.3	Éléments identifiables dans le problème des canalisations au débits maximaux à travers d'une représentation (max, min) .	13
1.4	Éléments identifiables dans le problème des chemins les plus fiables à travers d'une représentation $(max, \times)$.	14
1.5	Dynamique d'un système à événements discrets par rapport à sa valeur propre.	15
2.1	Alignements indépendants (a) et en conflit (b) pour les opérations de transfert	27
2.2	Synchronisation et conflit dans les réseaux de Petri	29
2.3	Opérations à ordonnancer	35
2.4	Ordonnancement optimum - modèle $(max, +)$ non linéaire	36
2.5	Ordonnancement optimal pour le système $(max, +)$ linéaire basé sur le <i>TPP</i>	41
2.6	Exemple pour validation du modèle $(max, +)$ linéaire basé sur le <i>PCI</i> .	44
2.7	Ordonnancement optimum pour le modèle $(max, +)$ linéaire basé sur le <i>PCI</i>	46
2.8	Ordonnancements optima : a) Topologie de référence, b) ordonnancement optimum pour l'instance 1, c) Ordonnancement optimum pour l'instance 2.	50
3.1	Opérateurs génétiques.	57
3.2	Workflow général pour la formulation du problème multi-objectif par une approche d'algorithmes génétiques.	60
3.3	Schémas de représentation de l'information dans les algorithmes génétiques structurés.	61
3.4	Instance simple pour la définition de la structure génomique de la problématique	63
3.5	Structure génomique proposée.	64
3.6	Mutation des gènes de contrôle et propagation vers les gènes de paramétrage - exemple.	69
3.7	Opération de croisement avec des parents ayant des structures différentes appliquée à l'exemple de la Fig. 3.4.	70
3.8	Flux d'exécution de l'algorithme SPEA2 avec des opérations en algèbre $(max, +)$.	78

3.9	Exemple pour validation de l'approche intégrée SPEA2 + modélisation (max,+).	79
3.10	Solutions dans l'espace de critères pour l'instance de la Fig. 3.9.	80
3.11	Solutions représentées par les individus non dominés dans la Fig. 3.10.	81
4.1	Exemples d'automates tropicaux : a) automate à 1 état. b) automate à plusieurs états.	86
4.2	Produit synchrone 2 automates : l'automate c) c'est le produit des automates a) et b).	88
4.3	Produit synchrone de 3 automates : l'automate d) c'est le produit des automates a), b) et c). L'automate e) représente le produit souhaité pour le cas d'étude. . .	88
4.4	Exemple de modélisation des conflits par automates tropicaux mono-état : les opérations mutuellement exclusives sont modélisées par des transitions sortant du même état.	90
4.5	Exemple de modélisation de conflit par automates tropicaux : représentation multi-état pour restreindre l'exécution de chaque opération à une seule occurrence.	91
4.6	Exemple pour 2 conflits de partage de ressources.	93
4.7	Décomposition en micro-événements pour l'exemple dans la Fig. 4.6.	93
4.8	Représentation du produit synchrone $G_1 G_2 G_3$ pour l'exemple de la Fig. 4.7. .	95
4.9	Ordonnancements qualitativement optimaux pour des systèmes avec une paire d'automates synchronisés : 3 cas possibles.	101
4.10	Ordonnancements Qualitatifs Optimaux pour des Systèmes à 3 Automates et avec 3 Paires d'Automates en Synchronisation.	104
4.11	Système à 3 automates ($n = 3$) présentant tous les possibles types de synchronisation (représentation simplifiée mono-état pour faciliter la compréhension). . . .	107
4.12	Ordonnancement qualitatif optimal pour $n = 4$ et (G_1, G_2) comme la seule paire en synchronisation.	111
4.13	Exemple pour 4 automates locaux (représentation simplifiée à un seul état pour faciliter la compréhension).	112
4.14	Résultats d'ordonnancement pour l'exemple dans la Fig. 4.13.	112
4.15	Résultats pour l'ordonnancement qualitatif pour $n = 6$ et 3 paires d'automates en synchronisation : (G_1, G_4) , (G_2, G_5) et (G_1, G_6)	113

Liste des tableaux

2.1	Données d'entrée pour les opérations de transfert sur la Fig. 2.3.	35
2.2	Données d'entrée pour validation du modèle basé sur le <i>TPP</i>	39
2.3	Données de validation du modèle (max,+) linéaire basé sur le <i>PCI</i>	44
2.4	Instance 1 pour le problème de la Fig. 2.8.	50
2.5	Instance 2 pour le problème de la Fig. 2.8.	50
3.1	Synthèse du comportement proposé pour les opérateurs génétiques.	72
3.2	Données d'entrée pour les opérations de transfert dans l'instance de la Fig. 3.9. .	79

Dédié à.....

Chapitre 1

Introduction

Préambule

Ce premier chapitre introduit le contexte des travaux et la problématique soulevée au regard du cas d'étude, et des objectifs opérationnels imposés. Ceux-ci concernent l'optimisation de la gestion d'un réseau de pipelines en considérant l'engagement de ressources physiques, leur maintenance, les requêtes clients et les pénalités potentielles pour retard de service. Les verrous scientifiques qui émergent de ce projet de recherche sont exposés ainsi que les approches proposées par la communauté ad hoc pour pallier ces verrous. Un état de l'art argumente la motivation d'exploiter des approches tropicales, telles que l'algèbre $(\max, +)$ et les automates tropicaux.

Sommaire

1.1 Contexte de la Recherche	2
1.1.1 Cadre de Collaboration	2
1.1.2 Cas d'Étude	2
1.2 Motivation et Objectifs	5
1.3 État de l'Art	6

1.1 Contexte de la Recherche

1.1.1 Cadre de Collaboration

Cette thèse doctorale est développée dans le cadre du programme de coopération post-gradué PCP (en anglais : *Post-graduate Cooperation Program*) entre la France et le Venezuela. Ce projet implique la collaboration entre les institutions académiques : *Universidad de Los Andes* à Mérida, Venezuela et INSA Lyon en France, et les partenaires industriels PDVSA et Thales Group. PDVSA (en espagnol : *Petróleos de Venezuela Sociedad Anónima*) participe en tant que société vénézuélienne d'exploitation pétrolière et Thales Group en tant que fournisseur de solutions de supervision au niveau international.

Le financement de la thèse relève d'une convention CIFRE avec le soutien financier de l'Association Nationale de la Recherche et de la Technologie (ANRT) et de Thales Group.

1.1.2 Cas d'Étude

Le système étudié est un terminal maritime d'exportation de pétrole regroupant un ensemble de réservoirs de stockage reliés à un ensemble de bras de chargement positionnés sur des quais au travers d'un réseau de pipelines. Les caractéristiques générales du système ont été recueillies dans les travaux de [1] et ont été fournies par l'entreprise pétrolière vénézuélienne PDVSA. Les bras de chargement permettent de fournir le pétrole aux tankers (i.e. des navires) qui arrivent au terminal. Dans ce système, les tankers sont assimilés à des clients. Chaque client, demande un *batch* de pétrole, i.e. une quantité de barils de spécifications données. Ce batch doit être délivré du réservoir de stockage au bras de chargement appareillé à un tanker. Dans cette recherche, un seul bras de chargement est considéré par quai pour les modèles réalisés. Cette hypothèse peut être facilement levée si, pour un système spécifique, plusieurs bras de chargement sont placés sur un seul quai mais le temps de traitement de la requête de transfert peut être estimé, ce qui est normalement le cas dans la réalité.

La Fig. 1.1 illustre un exemple simple de l'abstraction à retenir par rapport aux éléments du terrain applicatif considéré qui sera appelé dorénavant "système". Dans la Fig. 1.1b), les vannes sont identifiées par des arcs labellisés et les segments de pipeline, par des nœuds. Les vannes représentent ainsi les dispositifs de contrôle de flux circulant dans le système.

Chaque opération de transfert ou requête (termes utilisés de manière indistincte) est réalisée en choisissant un alignement de pipelines dans le réseau reliant les éléments source (réservoirs) aux éléments puits (bras de chargement) à considérer. Un *alignement* est défini comme un chemin de pipelines résultant de la commande appropriée des dispositifs de contrôle de flux. Ici, le pétrole est supposé s'écouler par gravité, comme c'est le cas dans certains terminaux maritimes

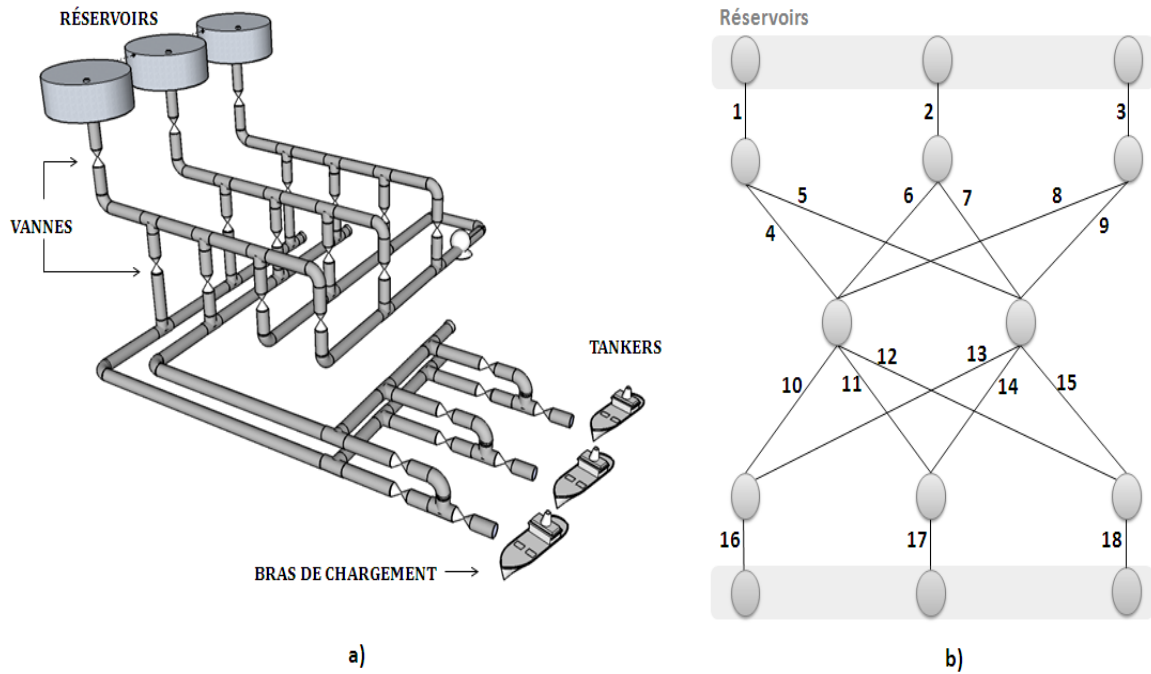


FIGURE 1.1: Schématisation d'un terminal maritime pétrolier (a) et son modèle sous forme de graphe non orienté (b).

au Venezuela. Un alignement est alors habilité lorsque toutes les vannes rattachées au parcours considéré commutent vers un état 'ouvert' et toutes les vannes adjacentes commutent vers un état 'fermé', pour isoler l'alignement du reste du réseau et ainsi éviter le mélange de batch différents.

Le mélange de deux types différents de pétrole générerait un batch de spécifications différentes, ce qui pourrait être étudié pour d'autres contextes de fonctionnement. Ici, le contexte retenu indique que le produit demandé par le client sera prélevé dans un réservoir unique et, afin d'interdire tout mélange de produit, deux alignements habilités simultanément seront toujours disjoints.

Les investigations ici s'inscrivent dans la poursuite de travaux précédents [1], [2], [3] et [4]. Les travaux réalisés [1], définissent la *capacité opérative* du réseau comme le nombre maximum d'alignements disjoints pouvant être habilités simultanément. La sélection d'alignements maximise la capacité opérative en reliant le maximum des chemins disjoints entre l'ensemble de réservoirs et l'ensemble de bras de chargement. Des chemins minimisant les interférences avec des chemins déjà envisagés ont été recherchés pour répondre simultanément à d'autres opérations de transfert [2]. Les travaux de [3] augmentent la difficulté de l'approche en considérant des chemins à sens de flux imposé par une pompe intermédiaire. Enfin, l'approche maximisant la capacité opérative proposée dans [1] est étendue par la minimisation des risques de défaillance des dispositifs de contrôle de flux dans [4].

Dans un contexte d'application proche au cas d'étude, les travaux de [5] présentent un positionnement intéressant sur l'ordonnancement d'opérations dans le cadre d'une raffinerie pétrolière par des approches relevant de la Recherche Opérationnelle. Sont considérés, le transfert de pétrole brut des pipelines de ravitaillement aux réservoirs de stockage, le transfert de ces réservoirs vers des réservoirs de chargement (pour des opérations de mélange de pétrole), et de ces derniers vers les unités de raffinage.

Les références dans la littérature sont peu nombreuses pour l'assistance à la supervision des terminaux maritimes. Néanmoins, la section 1.3 donne un état de l'art sur les travaux de modélisation pour des applications de différente nature transposables à la problématique abordée.

Ici, il est considéré que des opérations de maintenance peuvent être effectuées sur certaines vannes, et pour chacune de ces opérations, toutes les vannes adjacentes à la vanne en maintenance devront être fermées pour permettre l'isolement nécessaire à l'intervention. Ces dernières vannes sont nommées : *vannes d'isolement*.

Ainsi, le cas d'étude se rapporte à un réseau de pipelines piloté par des dispositifs de contrôle de flux (i.e. des vannes) et où des opérations de maintenance peuvent être effectuées sur ces dispositifs. D'ores et déjà, trois types de conflits de partage de ressources sont à considérer : le premier entre les requêtes clients (sélection d'un alignement, imposant un flux au travers des vannes affectées), le deuxième résultant des opérations de maintenance (vannes d'isolement ou de flux), et le troisième entre les opérations de transfert et de maintenance. En outre, le service tardif des clients entraîne des pénalités pour le terminal qui peuvent varier selon certaines priorités. La caractérisation de la gestion de pénalités sera présentée au chapitre suivant. Les hypothèses de fonctionnement retenues du réseau de pipelines sont les suivantes :

Hypothèse 1.1. Les canalisations sont considérées comme toujours pleines de pétrole. Ce pétrole stocké dans les canalisations est le produit résiduel des opérations de transfert précédentes.

Hypothèse 1.2. La quantité du produit résiduel stocké dans les canalisations est considérée comme négligeable par rapport aux quantités transférées aux clients. Ainsi, elles n'altèrent pas les spécifications du produit à fournir au client si le type de pétrole diffère.

Hypothèse 1.3. Les vannes de flux possèdent deux états possibles : ouverte ou fermée (i.e. on/off).

Hypothèse 1.4. Le phénomène de coup de bélier ¹ n'est pas considéré et les systèmes pertinents de protection des canalisations sont supposés être mis en place.

En termes de notation, dans les sections qui suivent, les *ressources* représentent les moyens physiques pour effectuer les opérations. Notamment, lorsque le contrôle de flux est réalisé par la commutation des vannes, l'ensemble de vannes dans le réseau représente l'ensemble de ressources à gérer. Lorsqu'une vanne est affectée à une opération de transfert, elle ne pourra pas être affectée

1. i.e. phénomène de surpression dans les canalisations dû à des commutations brusques des vannes.

à une autre opération si les états atteints après les commutations sont différents ou si cette affectation entraîne un mélange des produits à transférer.

1.2 Motivation et Objectifs

La recherche est motivée par le besoin de fournir une assistance à la décision pour les opérateurs de supervision devant établir un ordonnancement d'opérations, et l'affectation de ressources pour chacune d'entre elles, dans un réseau automatisé de pipelines où les ressources sont contraintes, et les retards dans les opérations impliquent des coûts financiers importants (i.e. pénalités).

L'intérêt majeur de ce travail est donc de déterminer une allocation de ressources pour satisfaire les opérations de transfert et de maintenance, minimisant les pénalités. L'originalité de ces travaux réside sur l'exploitation des informations de supervision au travers des approches algébriques tropicales. Telles approches, permettent d'obtenir des modèles non linéaires concis et intuitifs et, dans certaines conditions, facilement linéarisables. En outre, une telle diversité d'informations est à l'origine d'une prise de décision optimisée en accord aux objectifs globaux de performance du système.

Dans de nombreuses applications, les opérateurs de supervision réalisent une décision basée sur une expertise chèrement acquise et parfois trop focalisée sur la productivité. Ici, la prise de décision se veut multivariée, elle implique des informations de disponibilité des dispositifs, de l'ordre d'arrivée des clients, des planifications de maintenance, des pénalités subies dans certaines conditions opérationnelles (i.e. retards dans le service), et de l'exploitation appropriée de la capacité opérative du réseau.

L'intégration de l'ensemble de ces contraintes est hautement complexe pour un opérateur devant réagir en temps réel. Il s'agit alors de proposer une démarche d'optimisation de la prise de décision des opérateurs leur permettant d'inclure la valeur ajoutée de leur expertise. Cela est réalisable en lui permettant de choisir, parmi plusieurs alternatives représentant des différents compromis parmi les objectifs, l'alternative la plus adéquate par rapport aux conditions d'exploitation en temps réel. L'opérateur pourra ainsi prendre en compte toute information non prévue survenant en temps réel, ainsi que tout aspect supplémentaire de subjectivité pour décider parmi un ensemble de solutions optimales.

Dans ce cadre, les travaux académiques ont répondu au développement des approches formelles et heuristiques, PDVSA fournit des données industrielles liées au fonctionnement de ses terminaux maritimes, et la contribution de Thales Group relève de son expertise en système de supervision industrielle.

Les propositions et résultats obtenus sont suffisamment génériques pour qu'ils soient applicables à des systèmes de différente nature de celle du cas d'étude.

Ainsi, l'objectif général est de proposer une démarche d'assistance permettant d'optimiser l'allocation de ressources auprès des opérateurs de supervision dans un terminal maritime pétrolier considérant : la sélection d'alignements, l'ordonnancement d'opérations, la planification de la maintenance, et en cohérence avec une minimisation des pénalités financières. Ces pénalités seront encourues par l'exploitant du terminal dans les cas de retards dans le service des clients.

De manière plus spécifique, il s'agira de :

- Analyser les opérations de transfert du terminal maritime et identifier les besoins d'optimisation, en cohérence avec la gestion de la maintenance préventive existante et la capacité opérative du système.
- Définir des modèles d'intégration considérant les différents besoins d'optimisation (tels que minimisation des pénalités, gestion de la maintenance, affectation des ressources et détermination de l'ordonnancement d'opérations) en tenant compte des interactions des éléments pouvant être en conflit.
- Développer des tests de validation des modèles d'optimisation, et aboutir à une formulation générale intégrant les opérations de maintenance, la gestion des pénalités et la sélection d'alignements.
- Sur la base des données industrielles liées aux terminaux maritimes réels, valider les propositions retenues.

Les principaux verrous identifiés concernent l'intégration, au niveau de la modélisation, des informations de chacun des axes de l'optimisation tels que la sélection d'alignements, l'ordonnancement d'opérations de transfert et les aspects de disponibilité liés aux opérations de maintenance. Comme conséquence de ce défi d'intégration, l'exploitation formelle de certaines propriétés du système à travers de formulations mathématiques a permis d'aboutir à des résultats satisfaisants.

Les approches abordées par la communauté scientifique pour pallier ces verrous et les approches choisies sont présentées dans la section suivante.

1.3 État de l'Art

Cette section présente un panorama sur des approches classiques abordant des problèmes d'optimisation similaires, ainsi que les développements actuels de la communauté scientifique.

Étant donnée qu'il s'agit d'une problématique d'affectation de ressources et d'ordonnancement d'opérations et que, pour les propos des investigations, les dates de début et de fin des opérations

sont recherchées, les axes sur lesquels se centre l'état de l'art relèvent des systèmes à événements discrets. Ainsi, les approches liées aux réseaux de Petri, automates conventionnels, automates $(\max, +)$, théorie des jeux, algèbre $(\max, +)$, théorie des graphes et intelligence artificielle, seront abordées. En outre, des aspects d'intégration et/ou transposition entre les différentes techniques seront présentés du fait que, dans la recherche actuelle, elles peuvent être complémentaires.

Le problème abordé relève des systèmes à événements discrets (SED) du fait que, d'un point de vue macro (système), les opérations dans le réseau s'exécutent à des instants de temps précis, pour une durée déterminée. Dans un point de vue micro (éléments), chacune des opérations résulte de la commutation des vannes entre deux états possibles : ouverte/fermée. Ainsi, les dates de début d'opérations peuvent être considérées comme des événements suivis par des événements de fin d'opération après une certaine durée. Aussi, l'habilitation d'un alignement pour une opération résulte de plusieurs événements de commutation des vannes impliquées.

Dans un cadre général, l'intérêt se focalise sur la recherche des dates d'engagement de ressources partagées sous contraintes de pénalités financières et d'indisponibilité (dû aux opérations de transfert, ou de maintenance préventive ou corrective). Il s'agit ainsi d'une problématique classique d'allocation de ressources déjà abordée dans des secteurs aussi variés que le transport [6], la production manufacturière [7], l'énergie [8], et l'hospitalier [9]. La particularité dans ces travaux repose, d'une part sur une plate-forme de supervision où les opérateurs prennent des décisions en temps réel et, d'autre part, sur les caractéristiques de fonctionnement du système : conflits multiples (du fait du partage de ressources), synchronisation, et l'intégration des informations connues par l'opérateur (i.e. topologie et opérations de maintenance) avec des informations connues par des niveaux de planification et management telles que les pénalités financières dérivant du service tardif des clients.

Abord par Théorie des Graphes

Comme représenté dans la Fig. 1.1, le cas d'étude est facilement transposable sur la forme de graphe. La théorie des graphes est une des approches naturelles pour la recherche de chemins optimaux. Les travaux dans [1] se basent sur cette théorie pour maximiser la capacité opérative du réseau. Toujours en référence à la théorie des graphes, les approches les plus classiques, comme la recherche du chemin le plus court ou de fiabilité maximum, sont sources d'inspiration pour établir des analogies par rapport à des systèmes de diverses natures. Les travaux de [10], [11] et [12] décrivent des techniques classiques de recherche de chemins basées sur la théorie des graphes. Notamment, dans [10], la recherche du chemin le plus court est abordée avec des préférences fournies par l'utilisateur, ce qui montre le besoin naturel de conjuguer plusieurs objectifs en une formulation unique ; dans ce cas : une distance et des préférences rapportées

sur les liens du graphe. Les travaux de [11] abordent la recherche des chemins disjoints inspirés par des réseaux de télécommunication où une bande passante minimale doit être assurée pour traiter des requêtes en maximisant le nombre de requêtes traitées. [12] aborde la recherche du chemin le plus fiable dans un réseau avec des défaillances multiples.

La théorie des graphes représente ainsi un cadre de référence classique pour la recherche de chemins optimaux. Cependant, les assertions temporelles sont difficilement appréhendées. La théorie des graphes sera alors exploitée pour transposer la topologie du réseau vers d'autres approches tels que les automates et les approches algébriques et heuristiques plus adéquates à l'intégration des expressions temporelles.

Conventionnellement, les SED s'adressent à des systèmes à variables d'états discrets où la dynamique est décrite par une commutation d'état sur événements ou combinaison logique d'événements. Différentes techniques de représentation, quelles soient graphiques ou algébriques, sont utilisées selon les objectifs. En outre, la notion du temps peut être également intégrée dans le modèle et des indicateurs de performance peuvent être obtenus par résolution d'équations d'états ou à l'aide de simulations. Parmi ces outils de modélisation classiques dédiés à ce type de systèmes les plus répandus sont les réseaux de Petri et la théorie des langages.

Abord par les Réseaux de Petri

La difficulté de la problématique posée ici se caractérise par un nombre limité de ressources (vannes) qui sont sollicitées par un nombre important d'opérations. Ce phénomène de partage de ressources est classique pour les SED et a été abordé amplement dans la littérature à l'aide des réseaux de Petri qui, par des représentations intuitives, permettent d'obtenir, au travers d'expressions algébriques ou de techniques de simulation, des bonnes estimations de performance. De nombreuses applications visant les allocations de ressources dérivent de l'usage de modèles à base des réseaux de Petri dont la modélisation des systèmes de production. Ces systèmes ont tendance à comporter de nombreux phénomènes facilement représentables par les réseaux de Petri tels que la synchronisation d'opérations, le partage de ressources, le parallélisme d'opérations indépendantes et la dépendance des matières premières, parmi d'autres. Tous ces phénomènes sont représentables intuitivement à l'aide des réseaux de Petri et les simulations qui en dérivent permettent une estimation des indicateurs les plus pertinents de performance (e.g. délais, coûts). [13] aborde une application dans le domaine chimique en permettant d'établir des transpositions entre le modèle de production en réseaux de Petri et des algorithmes dédiés appelés PNS (en anglais : *Production Network Synthesis*) pour la détermination de l'affectation optimale des ressources. Par contre, les travaux dans [14] se centrent sur la modélisation et l'analyse d'inventaires dans les systèmes de production à l'aide d'une composition de différents sous-modèles basés sur

les réseaux de Petri tels que : des sous-modèles d'entrée, de sortie, d'inventaire, et de liaison. Des aspects liés à l'incertitude de certaines mesures peuvent être intégrés à ce type de modèles par des approches stochastiques ou de logique floue. Un exemple comportant les deux conforme les travaux de [15] qui renforcent l'intérêt de ces approches en faisant appel à la crédibilité. Dans le cadre du partage de ressources, les réseaux de Petri avec conflits sont d'intérêt particulier.

Étant donné que la notion du temps est intrinsèque pour les problèmes d'ordonnancement, les outils de modélisation discrète à considérer doivent pouvoir intégrer cette notion temporelle. Les réseaux de Petri temporisés (en anglais *Timed Petri Nets*) ou TPN et les automates temporisés répondent bien à cette extension. Notamment, les réseaux de Petri temporisés avec conflits abordent quelques aspects fondamentaux ici. Des analogies entre ce type de modèles et les automates conventionnels ou les automates $(\max, +)$ peuvent également être établies. Dans les travaux de [16], la modélisation par TPN analysés à temps discret détermine des dates minima et maxima de franchissement des transitions. Dans ce contexte, les TPN sont transformés sous la forme d'automates. Les travaux de [16] donnent une vue d'ensemble sur différentes techniques de modélisation permettant de représenter le comportement logique et temporel.

Les relations mathématiques décrivant le comportement dynamique représenté par les réseaux de Petri sont souvent hautement non linéaires dû aux phénomènes de synchronisation. Notamment, lorsque deux opérations a et b doivent se synchroniser, celle qui s'achève la première doit attendre à ce que la deuxième s'achève également pour enchaîner sur d'autres activités. Ainsi, au niveau temporel, la détermination du maximum des deux dates d'achèvement, i.e. la fonction $\max(a, b)$ est non linéaire. Les travaux de [17], [6] et [18] illustrent cette application des réseaux de Petri pour la modélisation des systèmes à événements discrets, ainsi que le besoin de représenter de manière linéaire ces types de comportement. L'algèbre $(\max, +)$ paraît dans ce contexte, très adéquate. Sans fournir de suite une description formelle et détaillée de cette algèbre, il peut être indiqué qu'elle permet d'obtenir des représentations algébriques linéaires directes pour la manipulation des phénomènes de synchronisation. Ainsi, dans le cadre des graphes à événements (réseaux de Petri sans conflits), une équation linéaire basée sur les opérateurs de maximisation et d'addition peut être définie pour chacun des franchissements des transitions du réseau. La résolution d'un système linéaire d'équations est alors très directe. En outre, comme dans l'algèbre conventionnelle, au travers d'un système d'équations linéaires, des artifices mathématiques formels permettent de prouver certaines propriétés comme l'existence d'une (ou plusieurs) solution(s), ainsi que la détermination des valeurs propres des matrices, pouvant décrire des indicateurs de performance dans le système. La puissance de modélisation et de résolution de cet outil mathématique est évidente ; néanmoins, le challenge fondamental lié à cette représentation concerne la modélisation des conflits. Le conflit de partage de ressources peut être résolu dans les réseaux de Petri par un choix de franchissement d'une seule transition parmi l'ensemble possible. Dans la littérature, de nombreux travaux, en particulier dans le domaine du transport, appliquent des techniques de modélisation basées sur des réseaux de Petri

déterminant des politiques de routage pour résoudre les phénomènes de conflit, et rendre ainsi les modèles linéaires. Une politique de routage définit essentiellement un critère de choix entre les transitions en conflit.

Les approches classiques par réseaux de Petri ne seront pas retenues pour cette recherche de manière directe. Néanmoins, les formulations mathématiques pouvant dériver de ces modèles seront exploitées dans le cadre des algèbres tropicales, notamment, l'algèbre $(\max, +)$. L'intérêt réside alors dans l'exploitation des modèles mathématiques de référence. La notion de politique de routage, classique dans les réseaux de Petri pour définir un critère de résolution de conflits, sera conservée.

Abord par Algèbres Tropicales

Concernant l'exploitation de l'algèbre $(\max, +)$ pour la gestion de conflits, les propositions de [6] établissent de manière convaincante une modélisation par systèmes d'équations linéaires $(\max, +)$ des tableaux horaires d'un réseau de transport routier. Les conflits sont gérés par une politique de routage avec l'établissement de préférences a priori. Des cas type sont définis et lorsqu'un conflit survient entre deux opérations 'a' et 'b', 'a' aura toujours la priorité. Dans d'autres applications, telle priorité ne peut être connue et d'autres approches doivent être prises en compte (e.g. l'affectation aléatoire de priorité). Les travaux de [18] abordent une application d'interaction de services où une modélisation croisée réseaux de Petri et algèbre $(\max, +)$ traite des conflits par une politique d'alternance à chaque itération périodique. Ainsi, à l'itération 1 la première opération du conflit a la priorité, à l'itération 2 l'opération 2, et ainsi de suite de manière périodique. Les travaux de [17] préconisent aussi une modélisation à l'aide des réseaux de Petri et d'automates en vue de la définition de la commande juste-à-temps d'un système manufacturier. Dans ces travaux, les phénomènes de synchronisation, i.e. début d'une opération dépendant de l'achèvement de plusieurs, sont étudiés à l'aide des graphes d'événements (i.e. réseaux de Petri où chaque place a au plus une transition entrante et une sortante) et de l'algèbre $(\max, +)$. Les conflits sont modélisés à l'aide des graphes d'états (i.e. réseaux de Petri où chaque transition a au plus une place entrante et une sortante). L'intérêt de l'application de l'algèbre $(\max, +)$ pour la description des phénomènes de synchronisation réside dans l'obtention des systèmes d'équations linéaires qui peuvent être analysés et résolus à l'aide des techniques algébriques classiques de produits matriciels. Autrement, le modèle serait hautement non linéaire et son analyse deviendrait beaucoup plus complexe et coûteuse. De ce fait, une représentation mathématique en algèbre classique ne sera pas retenue pour modéliser la problématique d'ordonnancement avec conflits, en considérant que des phénomènes de synchronisation font partie du comportement du système.

L'algèbre $(\max, +)$ appartient à la classe des algèbres tropicales exposées dans les travaux de Imre Simon (voir [19]). Ces méthodes de modélisation sont de plus en plus courantes et couvrent plusieurs domaines de recherche scientifique visant à exploiter leur caractère intuitif et concis pour modéliser des systèmes difficilement abordables en algèbre conventionnelle.

Au préalable, il s'agit d'utiliser une structure algébrique reposant sur deux opérateurs de base, et les propriétés mathématiques associées pour décrire un comportement dynamique. L'exploitation d'opérateurs dédiés spécifiquement à la description de certains types de phénomènes, comme par exemple de synchronisation (avec l'opérateur de 'maximisation'), aboutit à définir des modèles plus simples et intuitifs. Un large spectre d'applications est alors ouvert, dépendant de l'algèbre tropicale² choisie. Par exemple, l'algèbre $(\min, +)$, basée sur les opérateurs de minimisation et addition, est exploitée pour des applications du plus court chemin (ou des analogies qui peuvent en dériver telles que le chemin le moins coûteux) dû au fort lien existant entre les algèbres tropicales et la théorie des graphes. Considérant que dans un graphe orienté pondéré, le poids peut être associé à une durée de parcours, un temps, un coût, ou toute autre pondération, le problème de trouver le chemin le plus court, i.e. plus efficace, entre toute paire de nœuds peut être formulé avec l'algèbre $(\min, +)$. Cette algèbre possède comme opérations de base la minimisation (représentée par l'opérateur $\oplus$) et l'addition (représentée par l'opérateur $\otimes$), comme élément nul : $\varepsilon = +\infty$, et comme élément neutre : $e = 0$. Un réseau peut être alors représenté par une matrice A où chaque entrée a_{ij} représente l'arc doté d'un poids minimal du nœud i vers le nœud j et si aucun arc ne relie ces deux nœuds, alors $a_{ij} = \varepsilon = +\infty$.

Le produit matriciel $(\min, +)$ est défini comme $(A \otimes B)_{ij} = \bigoplus_{k=1}^n A_{ik} \otimes B_{kj}$ (consulter [20] pour plus de détails) et, comme dans l'algèbre conventionnelle, l'opérateur $\otimes$ peut être omis, e.g. $a \otimes b \Leftrightarrow ab$. La Fig. 1.2 illustre le lien entre le modèle du graphe, sa matrice d'adjacence, et la détermination des plus courts chemins à travers d'un produit matriciel dans l'algèbre $(\min, +)$. En termes de notation, A^n dénote la puissance n de la matrice A (e.g. $A^3 = A \otimes A \otimes A$), $Pmin_{ij}$ dénote le poids du chemin à poids minimal de i vers j , et σmin_{ij} le poids du cycle à poids minimal de i vers j . La Fig. 1.2 illustre par l'opération de puissance ' n ' de la matrice A , les chemins les plus courts entre toute paire de nœuds i, j . Dans la figure, les entrées égales à ε sont dénotées avec ' $'$ ' pour soulager la lecture. Les chemins les plus courts entre toute paire de nœuds i et j comportant au plus n arcs sont alors représentés par $A \oplus A^2 \oplus \dots \oplus A^n$. Ainsi, avec une technique algébrique formelle, les plus courts chemins pour toute paire de nœuds sont obtenus à l'aide d'un simple produit matriciel.

L'adaptation de la structure algébrique à certains phénomènes particuliers peut donner lieu à d'autres algèbres. Par exemple, l'algèbre $(\max, \min)$, aussi nommée 'algèbre goulot' (en anglais : *bottleneck algebra*), utilise comme opérateurs fondamentaux la maximisation (opérateur $\oplus$) et

2. Dans la littérature, des confusions peuvent survenir sur la définition d'algèbre tropicale. Ici, les algèbres tropicales sont toute une classe d'algèbres groupant les algèbres $(\max, +)$ et $(\min, +)$, parmi d'autres.

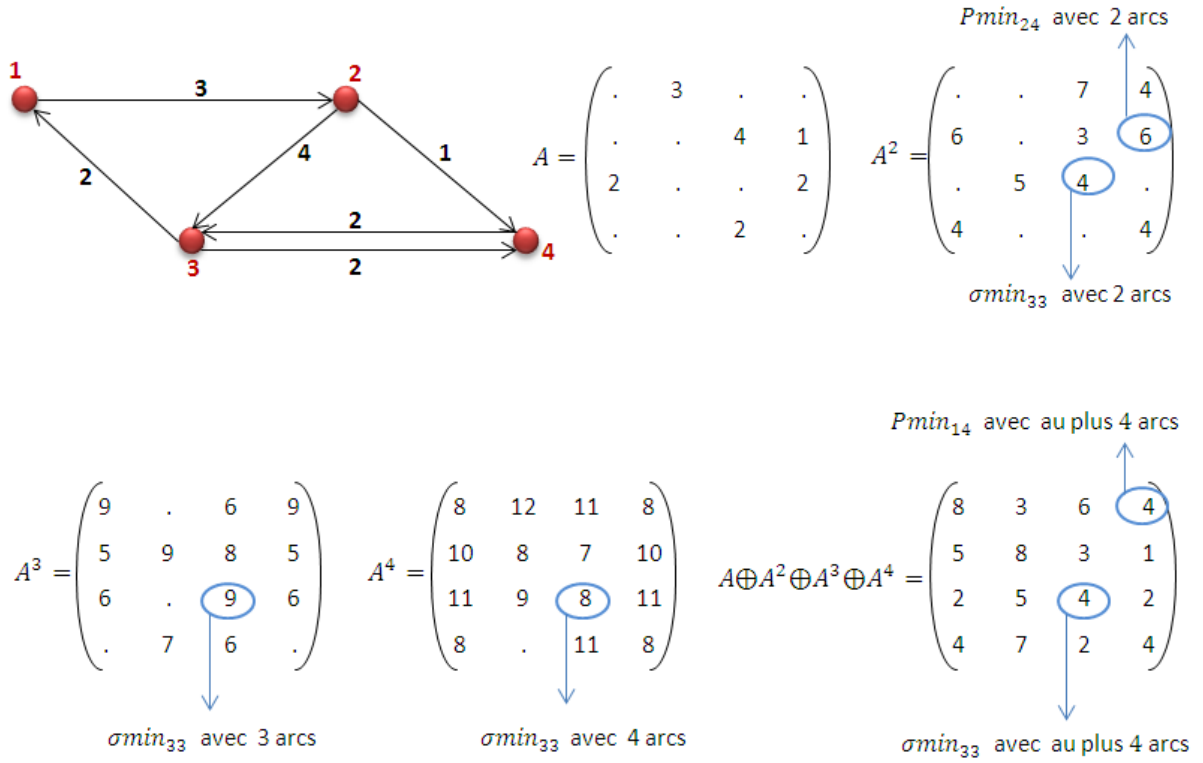


FIGURE 1.2: Éléments identifiables dans le problème du plus court chemin à travers d'une représentation $(min, +)$.

la minimisation (opérateur $\otimes$) appliqués à l'ensemble $S = \mathbb{R}^+ \cup +\infty$, où les éléments nul et neutre sont $\varepsilon = 0$ et $e = +\infty$, respectivement. Comme dans toute algèbre, ces éléments sont sélectionnés afin de respecter des propriétés de base comme l'associativité, commutativité et distributivité, en cohérence avec les opérateurs mathématiques de base. Dans cette algèbre, comme dans l'algèbre conventionnelle, l'addition d'un élément avec l'élément nul donne comme résultat le premier élément, i.e. $a \oplus \varepsilon \Leftrightarrow \max(a, \varepsilon) = \max(a, 0) = a$. Le produit, par contre, donne comme résultat l'élément nul, i.e. $a \otimes \varepsilon \Leftrightarrow \min(a, \varepsilon) = \min(a, 0) = 0$. Les travaux de [21] peuvent être consultés pour une description plus détaillée de la théorie. Dans [22], la pertinence de l'application de l'algèbre $(\max, \min)$ pour les problèmes de cheminement est présentée ainsi : "Le débit d'une canalisation étant limité par le minimum des débits des tuyaux qui la composent, calculer A^n dans l'algèbre max-min revient à trouver les canalisations de débit maximal parmi celles de longueur n ". La Fig. 1.3 présente un exemple illustratif où $C(DebMax)_{ij}$ dénote une canalisation à débit maximal entre les nœuds i et j . Parmi les 3 canalisations reliant le nœud 1 au nœud 4, une comporte 2 sections à des débits de 5 et 8 m^3/min , la deuxième comporte une seule section de débit de 4 m^3/min , et la troisième deux sections à 10 et 2 m^3/min . Chacune de ces canalisations ou chemins est physiquement restreinte par la section de débit minimal. La canalisation de débit maximal possible entre les nœuds 1 et 4 est celle passant par le nœud 2. Cela est directement calculé par le produit matriciel pour toute paire de nœuds. La figure montre qu'aucune canalisation à 2 arcs ne permet de relier les nœuds 3 et 4 (voir matrice A^2).

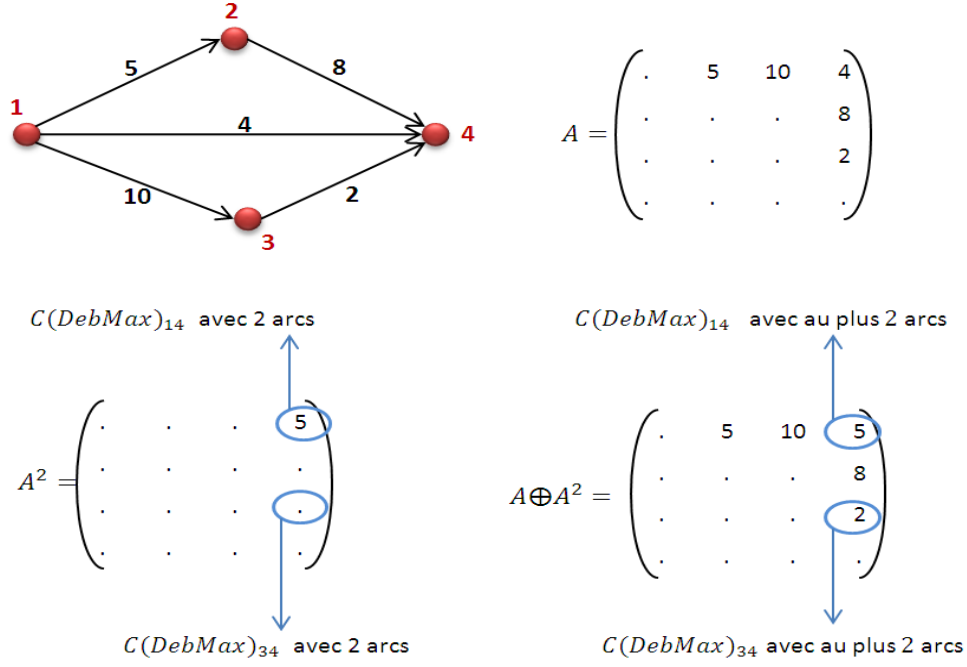


FIGURE 1.3: Éléments identifiables dans le problème des canalisations au débits maximaux à travers d'une représentation (max,min).

Néanmoins, pour le scénario plus général avec *au plus* 2 arcs dans la canalisation, le résultat optimal obtenu est une canalisation à 1 arc, ce qui est directement trouvé avec $A \oplus A^2$.

Les algèbres tropicales s'appliquent également pour la recherche des chemins les plus fiables à l'aide des opérateurs de base de maximisation et multiplication dotés des éléments nul et neutre : $\varepsilon = 0$ et $e = 1$, respectivement. Ainsi, le calcul de A^n , où A est la matrice de fiabilité de tous les segments reliant une paire de nœuds dans un graphe, permet d'obtenir l'ensemble des chemins les plus fiables reliant toute paire de nœuds. La Fig. 1.4 illustre ce principe en considérant un réseau de canalisations où les poids sur les arcs représentent la fiabilité de chaque section, et $C(FiabMax)_{ij}$ dénote la canalisation de fiabilité maximale entre les nœuds i et j .

Un des travaux précurseurs pour l'exploitation des algèbres tropicales pour la modélisation des procédés industriels est présenté dans [23]. Les travaux de [22] montrent que lorsque ces systèmes atteignent un régime 'périodique', les transitions (qui modélisent le passage d'une phase de production à l'autre dans un réseau de Petri) sont franchies à des intervalles temporels égaux. La valeur de l'intervalle est la valeur propre du système d'équations linéaires. Sans simulation, et avec une approche analytique, les valeurs et vecteurs propres du système peuvent être obtenus grâce à la linéarité du modèle donnée par l'algèbre. Cela permet d'estimer des indicateurs de performance tels que le taux de production en régime 'périodique'. Connaissant les valeurs propres, des cycles critiques (i.e. circuits avec les taux de production les plus bas) peuvent être identifiés permettant de cibler les phases dans lesquelles des ressources peuvent être ajoutées pour améliorer la performance globale du système. La Fig. 1.5 montre un exemple d'un réseau de transport

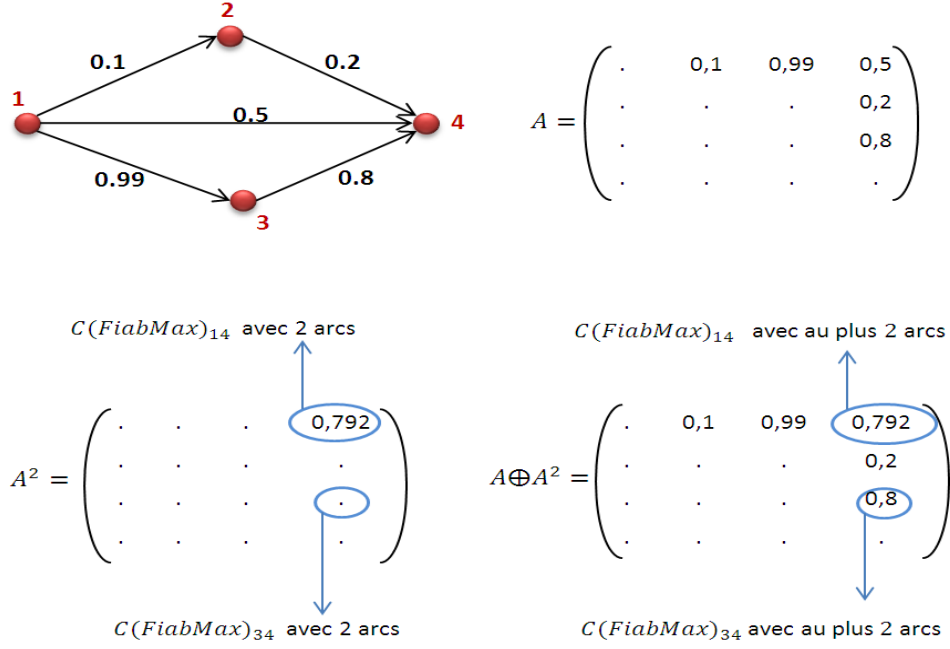


FIGURE 1.4: Éléments identifiables dans le problème des chemins les plus fiables à travers d'une représentation $(\max, \times)$.

reliant 3 villes où chaque parcours a une durée déterminée, e.g. le trajet de la ville '1' à la ville '2' est de 2 heures. Ainsi, le système est représenté avec un modèle $(\max, +)$ linéaire de la forme $X(t) = AX(t - 1)$ où $X(t) = (x_1(t), x_2(t), x_3(t))^T$ représente les dates de départ de chaque ville, qui dépendent des dates des dernières correspondances dans la même ville. Ainsi, en supposant que la date initiale pour chaque parcours est $x_i(0) = 0$, alors $X(1) = AX(0) = (2, 3, 2)^T$, $X(2) = A^2X(0) = (5, 6, 4)^T$, etc. Après la 5-ème itération, le système atteint un régime périodique remarqué par les expressions de A^6 , A^7 , et ainsi de suite. Notamment, $X(5) = (14, 15, 13)^T$, $X(6) = (17, 18, 16)^T = 3 \otimes X(5)$, et $X(7) = (20, 21, 19)^T = 3 \otimes X(6)$. Le système évolue vers un régime périodique de 3 unités de temps obtenu par la valeur propre de la matrice A où $\lambda = 3$. Les travaux [24] et [25] constituent des références pédagogiques pour les bases et méthodes liées au calcul algébrique de λ , et [26] peut être consulté pour une application à la modélisation et évaluation de performance des systèmes cycliques de type job shop.

La notion d'indépendance linéaire tropicale pour les matrices $(\max, +)$ est exploitée dans les travaux de [27] pour l'étude d'existence des solutions pour certains problèmes d'optimisation basés sur $(\max, +)$.

Pour plus d'information sur l'algèbre $(\max, +)$ pour la modélisation des chaînes de production ainsi que des réseaux de transport, le lecteur se rapportera à la référence [22]. Les travaux de [20] et [28] des excellentes références de base sur la théorie des systèmes $(\max, +)$ linéaires. [29] peut être consulté pour la théorie $(\max, +)$ appliquée au contrôle du trafic, et [30] pour une application de maintenance.

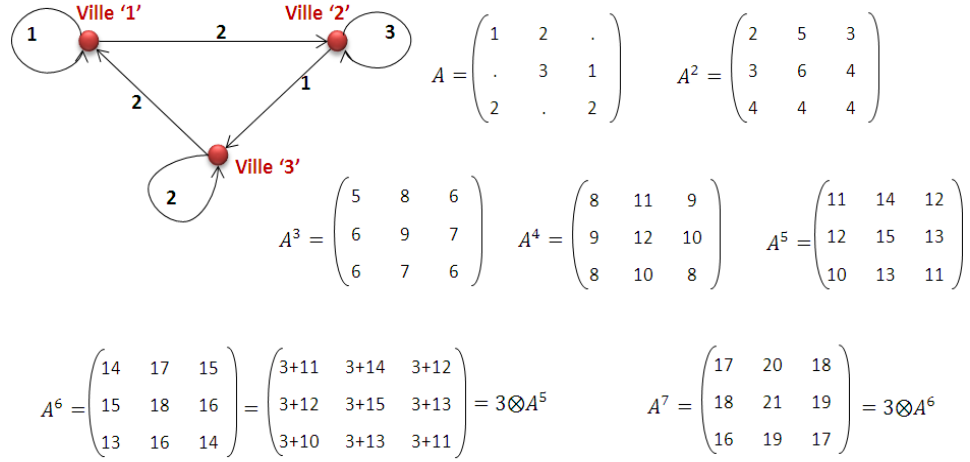


FIGURE 1.5: Dynamique d'un système à événements discrets par rapport à sa valeur propre.

L'algèbre $(\max, +)$ peut avantageusement devenir complémentaire à des approches de modélisation conventionnelles. Par exemple, les travaux de [27] portent sur les transpositions entre les systèmes de type $AX \leq BX$ ou de type $AX \oplus C \leq BX \oplus D$ (dénotés comme linéaire et affine, respectivement, dans le sens $(\max, +)$) et la théorie des jeux. Notamment, ces travaux démontrent que ces problèmes de décision, modélisés à l'aide de l'algèbre $(\max, +)$, peuvent être transposés vers des modèles de la théorie des jeux. De manière générale, la théorie des jeux est l'étude formelle des situations de prise de décision entre joueurs de manière stratégique. Pour le cas le plus simple de 2 joueurs, chacun peut à son tour prendre une décision qui peut lui rapporter un bénéfice quantifiable. Ensuite, son opposant effectue sa manœuvre avec aussi une conséquence quantifiable et le but du jeu pour chacun est de maximiser le bénéfice acquis. Pour une description des principes de la théorie des jeux, le lecteur peut se rapporter à [31]. Dans les investigations ici, aucune interaction de stratégies n'est considérée, notamment la prise de décision s'effectue dans un seul sens : prise de décision de l'opérateur de supervision. Ainsi, les formulations mathématiques deviennent plus simples et, par conséquent, des approches de la théorie de jeux ne seront pas retenues.

Les travaux de [7] portent sur l'optimisation de l'ordonnancement d'opérations dans les systèmes manufacturiers, et représentent une référence très intéressante ici. Une analogie peut effectivement être déduite du comportement aux réseaux de flux ; ainsi, l'algèbre $(\max, +)$ fera partie de nos principales investigations.

Par ailleurs, la transposition des réseaux de Petri vers des représentations sous forme d'automates a toujours été de grand intérêt dû au cadre formel des langages. L'ouvrage [32] est une référence dans ce contexte.

Abord par Automates

Les automates temporisés ([33] et [34]), conçus à l'origine pour résoudre des problèmes de vérification, sont bien entendu aptes à procurer des modèles pour la détermination d'ordonnancement ([35], [36]). Les travaux de [37] illustrent le lien entre la théorie de langages et l'analyse des systèmes de commande, en traduisant des formules de logique temporelle propositionnelle vers des modèles à base d'automates pour définir des spécifications de commande. Ainsi, la théorie de langages, très riche formellement, représente un corpus consistant au niveau mathématique pour représenter le comportement des systèmes à événements discrets.

D'autres techniques alternatives qui n'ont pas été détaillées dans cette étude incluent les automates cellulaires, pour lesquels le lecteur peut se rapporter à [38] et [39]. Notamment, ces modèles exploitent des notions de parallélisme en considérant un ensemble d'entités ou cellules communicantes, chacune avec un état qui évolue selon les états des cellules dans son voisinage par rapport à certaines règles prédéfinies. [39] aborde une application à l'ordonnancement de tâches prévues pour un algorithme dans un réseau multi-processeur. Cependant, la définition des règles d'évolution est souvent coûteuse. Quelques approches considèrent l'application d'algorithmes évolutionnaires comme prometteuse dans la détermination des meilleures règles tel que des instances similaires pour le même modèle d'automates cellulaires puissent converger vers l'optimum. Dans ce contexte, l'objectif d'optimisation est typiquement la minimisation du temps d'exécution de l'algorithme.

Ici, les automates cellulaires n'ont pas été retenus du fait de la disposition de techniques plus intuitives de représentation permettant d'éviter une détermination coûteuse des règles d'évolution. Néanmoins, l'exploitation de cette technique pour des problématiques similaires à celle adressée ici pourrait être envisagée pour comparaison de résultats, ainsi que pour exploration d'une possible intégration avec les algèbres tropicales.

Les automates temporels utilisés dans [17] résultent d'une transposition entre les réseaux de Petri et les automates type tas (en anglais : *heaps of pieces*). Ces automates permettent de représenter des opérations ou engagement de ressources sur des créneaux disponibles. Une pièce à traiter est alors placée dans un créneau si telle machine est affectée à cette opération. La représentation considère un empilement des pièces lorsque les opérations sont exécutées dans le temps, et cette hauteur de tas de pièces (i.e. *heaps*) permet de déterminer le makespan pour le système.

Ces représentations sous forme d'automates conduisent à une modélisation formelle et exacte. Notamment, ils conjuguent une représentation à l'aide de la théorie classique d'automates pour le comportement logique, et l'exploitation de l'algèbre $(\max, +)$ pour la représentation du comportement temporel. À ce jour, un effort considérable porte sur le développement de cette théorie

et l'application à des systèmes de différente nature. Quelques références fondamentales [40] définissent les bases pour l'application des automates $(\max, +)$ pour la modélisation des systèmes à événements discrets. Les travaux [41], [42] et [43] peuvent être consultés pour des développements sur la modélisation modulaire à base d'automates $(\max, +)$, ainsi que des compositions d'automates et la synthèse de superviseur discrets. Dans ces travaux, la modélisation modulaire d'automates vise à obtenir une représentation de chaque sous-système en termes d'un automate $(\max, +)$. Des événements communs reflètent l'interaction des sous-systèmes au travers de la synchronisation. En outre, différents types de compositions d'automates $(\max, +)$ ont été étudiés pour générer un comportement global. Ainsi, ces approches visent à faciliter la modélisation par modularisation. Sur le modèle global obtenu, des contrôleurs peuvent être synthétisés. Considérant les automates $(\max, +)$, une approche de commande juste-à-temps consiste ainsi à déterminer un superviseur qui fonctionne comme un retardateur d'exécution d'opérations, générant un système contrôlé tel que chaque opération est réalisée au plus tard, tout en satisfaisant des dates limites pour des séquences d'opérations, e.g. des dates limites d'achèvement.

La proposition dans [44] est fondamentale ici, elle apporte la définition d'un produit synchrone d'automates $(\max, +)$. L'aspect primordial de ce travail se centre sur des modèles considérant l'exécution parallèle d'opérations où le produit synchrone est défini sur une extension de l'alphabet des automates originaux.

Le modèle global de [44] définit que les opérations exécutées en parallèle parviennent à une date d'achèvement égale au maximum des chacune des dates d'achèvement locales. Cette approche est concise et directe ; néanmoins, la synchronisation de deux (ou plusieurs) opérations en parallèle implique que si l'une d'elles s'achève, les ressources qui lui sont dédiées à cet instant resteront oisives jusqu'à l'achèvement des autres. Le besoin de pallier cette limite, a inspiré ici le développement d'un nouveau produit synchrone d'automates $(\max, +)$.

L'intérêt des approches formelles à l'aide d'automates, liées à une modélisation algébrique en vue d'obtenir des modèles linéaires est clair. Néanmoins, l'usage des méthodes approximatives reste importante et parfois nécessaire. Ainsi, les modèles linéaires algébriques ne peuvent être obtenus que si des priorités sont établies a priori pour les conflits de partage de ressources. Cela n'est pas toujours le cas dans la réalité de l'exécution des opérations industrielles et des heuristiques doivent être établies pour déterminer des solutions lorsque aucune autre information ne permet de faire une priorisation. Notamment, l'application des approches heuristiques est justifiée lorsqu'il est nécessaire de réduire le temps de recherche de solution et lorsque le nombre d'objectifs d'optimisation à gérer augmentent. En outre, elles permettent de trouver une solution aux modèles $(\max, +)$ non linéaires puisque des solutions analytiques ne sont pas possibles. Les approches approximatives réduisent également la taille et la complexité des modèles formels.

Abord par Intelligence Artificielle et Optimisation Multi-Objectif

L'usage des approches approximatives reste appréciable. Dans la pratique, une de ses avantages est qu'une fois réalisée la transposition du problème au contexte heuristique (i.e. techniques basées sur les algorithmes génétiques, colonies de fourmis ou autres), l'obtention de la solution est transparente pour le modélisateur puisqu'il s'agit d'un ajustement des paramètres d'évolution qui sont liés à la technique de résolution. Cette avantage peut aussi devenir un inconvénient si l'objectif est d'analyser un comportement au travers d'un modèle mathématique.

De nombreuses contributions dédiées aux systèmes de production présentent des analogies directes avec la problématique abordée ici. À titre d'exemple, [45] présente une approche basée sur colonie de fourmis pour l'optimisation d'ordonnancement dans un job shop. [46] présente une application d'optimisation par *colonie de fourmis* et par *essaims particulières* à l'intégration des modèles de manufacture flexible et les modèles de gestion de matériaux. Les travaux de [47] portent sur l'optimisation des coûts dans la conception d'un atelier de production.

Les techniques approximatives sont très bien adaptées pour aborder des problèmes d'optimisation multi-objectif. Dans [48], les bases de l'optimisation multi-objectif sont présentées : les approches, leur classement et leur évolution sur les deux dernières décades.

Les approches évolutionnaires se basent sur une émulation de l'évolution naturelle des espèces en abordant une résolution au travers d'une vue de populations qui évoluent dans le temps. Au début, une population est générée où chaque individu représente une solution au problème. Ensuite, une sélection et des modifications génétiques, telles que la mutation et le croisement, sont appliquées à la population à chaque itération. Au bout de plusieurs générations, la population n'est plus identique et les meilleures solutions (i.e. individus) ont été conservées par ce processus d'évolution. Afin d'illustrer le principe, dans un contexte binaire, si une solution à un problème quelconque est identifiée comme $P_1 = '1\ 0\ 1\ 1\ 1'$, où 5 variables prennent des valeurs 0 ou 1, et une autre solution est représentée comme $P_2 = '1\ 1\ 0\ 0\ 0'$, une opération de mutation de la première solution par exemple dans la position 4 donnerait comme résultat $P_{1M} = '1\ 0\ 1\ 0\ 1'$. D'autre part, le croisement des solutions P_1 et P_2 dans la position 3 donnerait comme résultat deux solutions, aussi appelées descendants en sachant que P_1 et P_2 jouent le rôle de parents, qui correspondent à : $D_1 = '1\ 0\ 1\ 0\ 0'$ et $D_2 = '1\ 1\ 0\ 1\ 1'$. Le principe est alors de partir d'une population initiale et de générer des populations successives à travers l'application d'opérations génétiques de modification des solutions et d'évaluation de leur qualité pour que le processus converge vers un ensemble de solutions avec une qualité optimale. Les travaux de [49] et [50] peuvent être consultés pour une introduction aux algorithmes génétiques et ses applications parmi lesquelles se trouvent : la conception des réseaux, des problèmes de logistique, d'engagement de ressources, de télécommunications, de réseaux de flux, de transport, d'ordonnancement de processeurs et de vérification de logiciels (voir [51]).

Lorsque plusieurs objectifs sont considérés, une solution optimisant tous les objectifs n'est pas forcément garantie, i.e. un objectif est amélioré au détriment d'un autre. Ainsi, la prise de décision implique la considération d'un compromis dans la qualité obtenue pour tous les objectifs. Cette approche s'adapte aux spécificités de notre problématique puisque les objectifs d'intérêt (i.e. minimisation des pénalités, retard des opérations de maintenance et sélection d'alignements conflictuels) sont non commensurables ; i.e. n'ont pas une même mesure et, par conséquent, ne peuvent pas être évalués de manière groupée. D'autres approches non retenues abordent une intégration de tous les objectifs dans une seule fonction objectif dont une des plus connues est l'agrégation ou somme pondérée de la forme $Min (\alpha_1 obj_1 + \alpha_2 obj_2 + \dots + \alpha_n obj_n)$. Tous les objectifs sont composés considérant qu'ils sont commensurables et que chacun a un poids associé α_i par rapport à son importance dans l'objectif global. Ainsi, le problème est réduit à un de type mono-objectif ; néanmoins, le choix des poids pour chaque objectif n'est pas évident. D'autres méthodes connues dans le traitement des problèmes multi-objectif incluent la méthode d'optimisation par contraintes, connue en anglais par ε -*constraint*, qui vise à retenir un seul objectif en transformant tous les autres sous forme de contraintes, i.e. $f_k(x) \leq \varepsilon_k$ où $\varepsilon_k \in \mathbb{R}^p$. Des méthodes hybrides proposent une fonction objectif représentant la somme pondérée des objectifs avec des contraintes sur chacun. La méthode des contraintes élastiques flexibilise les contraintes sur les objectifs en permettant d'être violées et en pénalisant ces violations dans la valeur de la fonction objectif. Les méthodes d'optimisation lexicographiques permettent de modéliser des objectifs incommensurables où, pour des raisons qui ne dépendent pas du décideur, ces objectifs sont classés de manière hiérarchique. Pour plus de détails sur ces méthodes et autres dans la littérature, le lecteur peut se rapporter à l'ouvrage [52].

Dans le contexte de l'optimisation d'alignements, les objectifs ne sont pas commensurables et la finalité des travaux est de permettre une prise de décision parmi plusieurs solutions candidates représentant des compromis entre les objectifs. Une technique courante est la frontière de Pareto. Dans [48] et [53], une description des bases de cette approche est présentée. Notamment, une notion de dominance est définie pour comparer des individus (i.e. solutions). Un individu est dénoté comme 'non dominé' si aucun autre individu n'arrive pas à améliorer certains objectifs sans en détériorer d'autres. Ainsi, à la fin de l'application d'un algorithme évolutionnaire, plusieurs individus peuvent être non dominés. Ils représentent différents compromis ou solutions optimales pour la problématique. L'ensemble d'individus non dominés est nommé la frontière de Pareto.

Plusieurs techniques permettent de gérer l'évolution des individus en vue de trouver la frontière de Pareto : MOMGA (en anglais : *Multi-objective Messy Genetic Algorithm*), PAES (en anglais : *Pareto Archived Evolution Strategy*), PESA (en anglais : *Pareto Envelope-based Selection Algorithm*) et SPEA (en anglais : *Strength Pareto Evolutionary Algorithm*) sont quelques exemples. Le lecteur se rapportera à l'ouvrage [54], et aux travaux de [55] et [56], pour une description générale de ces techniques qui se différencient dans la manière dont elles

gèrent l'évaluation des individus au cours des générations. Par exemple, l'algorithme SPEA, ou algorithme évolutionnaire de force de Pareto, est relativement récent comme le montre le travail dans [57], et propose l'évaluation de chaque individu par rapport à sa force (*'strength'*). Pour chaque individu, cet indicateur comptabilise le nombre d'individus dominés par l'individu en question. Dans le contexte applicatif, [58] aborde le SPEA2, qui est une amélioration du SPEA pour la recherche du chemin le plus court. La définition ou amélioration des techniques évolutionnaires est un large domaine de recherche actuel. Ici, le SPEA sera considéré largement suffisant pour valider les premiers résultats.

Conclusions Partielles

La problématique d'engagement de ressources partagées dans le contexte d'un réseau de flux comportant des contraintes de lancement d'opérations de maintenance et de pénalités causées par un service tardif a été présentée. La littérature identifie de nombreuses approches relevant des SED pour aborder des problèmes similaires. Les approches tropicales sont exploitées du fait de leur potentiel de modélisation apportant des résultats concis et formels permettant l'exploitation de propriétés largement étudiées de l'algèbre linéaire conventionnelle.

Le document comporte cinq chapitres dont le premier est introductoire. Le chapitre 2 se centre sur la modélisation à l'aide des algèbres tropicales, et présente plusieurs modèles mono-objectif linéaires et non linéaires. L'optimisation globale est décomposée en abordant premièrement l'ordonnancement des opérations de transfert minimisant des pénalités, puis se complexifie pour progressivement arriver à un problème d'optimisation multi-objectif. Le chapitre 3 aborde l'optimisation multi-objectif minimisant les pénalités et les retards dans les opérations de transfert pour l'ordonnancement et la sélection d'alignements optimum. Dans ce chapitre une proposition intégrant l'approche algébrique $(\max, +)$ et un algorithme génétique est présentée. Le chapitre 4 propose une modélisation à base d'automates tropicaux pour la recherche d'un ordonnancement optimal pour un comportement au plus tôt motivant la définition d'un nouveau produit. Enfin, le Chapitre 5 présente les conclusions et perspectives.

Chapitre 2

Algèbres Tropicales et Modèles d'Optimisation Mono-objectif

Préambule

Ce chapitre aborde la modélisation des problèmes d'ordonnancement dans un réseau de pipelines à l'aide des algèbres tropicales. Il présente les modèles et les résultats obtenus validant la résolution des problèmes d'optimisation mono-objectif; notamment de minimisation des pénalités subies par le terminal maritime pour un ensemble d'opérations de transfert à traiter. Une synthèse est présentée sur les algèbres tropicales en vue de répondre à la problématique de datation abordée. Ensuite, le système est modélisé à l'aide de cette formalisation en explorant des représentations $(\max, +)$ non linéaires, puis linéarisés, pour des alignements connus a priori. Le chapitre se termine par la proposition d'un modèle étendu permettant également d'optimiser la sélection d'alignements.

Sommaire

2.1 Algèbres Tropicales - Motivation et Cadre Théorique	23
2.2 Modélisation du Cas d'Étude	25
2.2.1 Fonctionnement du Système	25
2.2.1.1 Gestion des Pénalités	25
2.2.1.2 Partage de Ressources	26
2.2.2 Modèle Non Linéaire d'Optimisation de l'Ordonnancement d'Opérations	29
2.2.2.1 Modèle Non Linéaire	30
2.2.2.2 Résultats	34

2.2.3 Exploration des Modèles Linéaires d'Optimisation de l'Ordonnancement d'Opérations	37
2.2.3.1 Modèle Linéaire basé sur la Pénalité Totale Potentielle	37
2.2.3.2 Modèle Linéaire basé sur la Criticité	42
2.2.4 Modèle d'Optimisation de l'Ordonnancement d'Opérations et de Sélection d'Alignements	47

2.1 Algèbres Tropicales - Motivation et Cadre Théorique

La phase de modélisation d'un problème nécessite souvent un effort et un temps considérable avant d'aborder les étapes de commande et d'optimisation. Dans le cadre des systèmes à événements discrets, de nombreux phénomènes ne peuvent pas être décrits de manière linéaire ou même intuitive à l'aide de l'algèbre conventionnelle (en se limitant aux opérateurs d'addition et de multiplication). Une proposition prometteuse est de transposer ces phénomènes au travers de structures mathématiques plus adaptées¹. Cette transposition permet de mieux représenter les propriétés formelles, telles que la linéarité par rapport à certaines variables décrivant son évolution, à travers d'opérateurs qui leur sont dédiés. L'exploitation des algèbres tropicales fournit les moyens de transposition pour modéliser ce type de systèmes par des expressions mathématiques à la fois intuitives, concises et voire linéaires.

Avant de présenter une description formelle mathématique, la motivation pour l'utilisation des algèbres tropicales est exposée.

Le potentiel applicatif des algèbres tropicales a été présenté dans [22]. Notamment, ces travaux abordent 'une nouvelle manière de compter', i.e. une structure algébrique différente, pour modéliser des systèmes dynamiques. Un exemple très intuitif présenté dans cette référence est le suivant : "pour l'employé d'une boulangerie qui fabrique des sandwiches jambon-fromage, un morceau de baguette 'plus' une tranche de jambon 'plus' une lamelle de fromage font un sandwich. Dans cette situation, l'addition de trois quantités de natures différentes donne un résultat aussi de nature distincte. Selon cette opération, ' $1 \oplus 1 \oplus 1 = 1$ '. De surcroît, si l'employé garnit le sandwich de deux tranches de jambon au lieu d'une, le résultat reste un sandwich unique. Autrement dit, ' $2 \oplus 1 \oplus 1 = 1$ '".

Ce genre d'abstraction est l'origine de l'usage des algèbres tropicales basées sur deux opérateurs fondamentaux différents de ceux de l'algèbre conventionnelle. Ces opérateurs introduisent des nouvelles propriétés afin de mieux appréhender le comportement des systèmes à événements discrets de divers points de vue tels que le comportement temporel et la gestion de ressources. Leurs origines datent des travaux de Imre Simon [19].

Le même type d'abstraction (i.e. un sandwich ne peut être produit que si ses ingrédients sont disponibles) peut être représenté par des réseaux de Petri. Néanmoins, c'est la formalisation par l'algèbre $(\max, +)$ qui exploite explicitement des propriétés mathématiques issues de l'algèbre linéaire conventionnelle (e.g. existence de solution pour des systèmes d'équations, valeurs propres, produits matriciels) pour décrire un comportement dynamique souvent basé sur des phénomènes de synchronisation.

1. où les opérateurs de base $\oplus$ et $\otimes$ décrivent directement les phénomènes fondamentaux du système étudié.

Dans l'exemple présenté, l'objet produit (i.e. 1 sandwich) résulte du minimum des opérandes. C'est un concept bien appréhendé par l'algèbre $(\min, +)$ dotée des 2 opérateurs de base que sont la minimisation et l'addition. Par contre, pour l'appréhension des phénomènes de synchronisation, où il s'agit de considérer le dernier des événements en termes de datation, dans un ensemble d'exécution simultanée, l'algèbre $(\max, +)$ est plus adaptée. Cette algèbre est basée sur les opérateurs de maximisation et d'addition. Un deuxième exemple classique pour l'application de cette algèbre est la gestion ferroviaire où le départ d'un train est conditionné aux arrivées d'autres trains dont les passagers doivent effectuer une correspondance. Ceci est évidemment extrapolable à tout autre réseau avec des ressources à interactions.

Dans le même esprit, d'autres règles peuvent émerger selon les besoins, par exemple l'algèbre $(\max, \min)$ ou l'algèbre $(\min, +)$. L'ouvrage [20] donne une introduction complète, ici sont rappelées les notions les plus utiles à la compréhension du propos de la recherche.

L'algèbre $(\max, +)$ est une structure mathématique dénotée comme $\mathbb{R}_{\max}$, formée par l'ensemble $\mathbb{R} \cup \{-\infty\}$ et deux opérations binaires $\oplus$ et $\otimes$, qui correspondent à la maximisation et l'addition, respectivement (voir (2.1)).

$$\forall a, b \in \mathbb{R}_{\max}, \begin{cases} a \oplus b = \max(a, b) \\ a \otimes b = a + b \end{cases} \quad (2.1)$$

Deux aspects importants de l'algèbre sont à noter : le manque de l'inversion additive et la propriété d'idempotence. Pour l'exemple précédent concernant le trafic ferroviaire, les dates de départ peuvent être notées par x_1 et x_2 pour les trains 1 et 2 (arrivant en gare), et x_3 pour le train 3, qui est conditionné par l'arrivée des deux premiers. La durée de chaque parcours (incluant le temps pour effectuer les correspondances) pour les trains 1 et 2 est notée par τ_1 et τ_2 , respectivement. Ainsi, la date de départ du troisième train peut être décrite en algèbre conventionnelle par $x_3 = \max(x_1 + \tau_1, x_2 + \tau_2)$, et sera décrite dans la notation de l'algèbre $(\max, +)$ par $x_3 = x_1 \otimes \tau_1 \oplus x_2 \otimes \tau_2$. Par conséquent, il s'agit d'un simple produit scalaire dans le sens $(\max, +)$ des vecteurs (x_1, x_2) et $(\tau_1, \tau_2)^T$. Si la date de départ du troisième train est imposée, comme une constante b par exemple, alors les dates de départ des autres deux trains peuvent être déterminées telles que b soit respectée. Ce résultat est obtenu en résolvant la simple équation $(\max, +)$ linéaire : $x_1 \otimes \tau_1 \oplus x_2 \otimes \tau_2 = b$. Dans le cadre $(\max, +)$, x_i est dénoté comme 'dateur' et représente la date de début d'une opération.

Formellement, cette structure algébrique est un semi-anneau commutatif idempotent. D'après [20], un semi-anneau $\mathcal{K}$ est un ensemble doté avec deux opérations de base : $\oplus$ et $\otimes$, qui respectent certaines propriétés algébriques classiques. L'élément nul est $\varepsilon = -\infty$, et l'élément neutre est $e = 0$. Les principales propriétés de cette structure (similaires à celles de l'algèbre conventionnelle) sont les suivantes :

Opération $\oplus$:

- est associative (e.g. $a \oplus (b \oplus c) = (a \oplus b) \oplus c$),
- est commutative (e.g. $a \oplus b = b \oplus a$),
- possède un élément nul ε (e.g. $a \oplus \varepsilon = a$),
- est idempotent (i.e. $a \oplus a = a$; $\forall a \in \mathcal{K}$).

Opération $\otimes$:

- est distributive par rapport à $\oplus$ (e.g. $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$),
- est inversible. Par exemple, si $2 \otimes 3 = 5$ alors $2 = 5 \oslash 3$ ou en notation conventionnelle : si $2 + 3 = 5$ alors $2 = 5 - 3$ (ici, l'opérateur $\oslash$ dénote l'inverse de l'opération $\oplus$),
- possède un élément neutre e qui satisfait $\varepsilon \otimes e = e \otimes \varepsilon = \varepsilon$.

Les expressions équivalentes en algèbres $(\max, +)$ et conventionnelle sont les suivantes :

$$\begin{array}{ll} a \oplus b \Leftrightarrow \max(a, b) & a \otimes b \Leftrightarrow a + b \\ a \oplus \varepsilon \Leftrightarrow \max(a, \varepsilon) = a & a \otimes \varepsilon \Leftrightarrow a + \varepsilon = \varepsilon \\ a \oplus e \Leftrightarrow \max(a, e) = a & a \otimes e \Leftrightarrow a + e = a \end{array}$$

On peut intuitivement reconnaître l'intérêt de l'application de cette technique de modélisation algébrique pour la description des phénomènes de partage de ressources et des contraintes de dépendances temporelles entre différents constituants du système. Ceci fait l'objet de la section suivante.

2.2 Modélisation du Cas d'Étude

2.2.1 Fonctionnement du Système

Cette section aborde la description du fonctionnement du réseau de pipelines en termes de la gestion des pénalités, ainsi que des aspects de partage de ressources, permettant de déterminer des modèles basés sur l'algèbre $(\max, +)$ représentant ces aspects de fonctionnement.

2.2.1.1 Gestion des Pénalités

Dans le cadre de la gestion de pénalités, chaque client contractualise une date limite d'achèvement de service. En cas de dépassement de cette date, des pénalités financières sont adressées à l'exploitant du terminal maritime. Le coût de la pénalité par heure de retard est spécifié par client, et la pénalité globale subie par le terminal pour chaque client est proportionnelle au retard total du service. Ainsi, l'objectif majeur de l'exploitant du terminal est de minimiser les pénalités globales pour un ensemble de clients dans un horizon de temps donné.

D'autre part, s'il est fait référence à la disponibilité des moyens de transfert impactant les retards, il est nécessaire de considérer certaines évaluations de sûreté de fonctionnement. Par hypothèse, l'échéancier des opérations de maintenance est connu².

Pour chaque client, une négociation est conclue au préalable avec l'exploitant du terminal pour s'accorder sur la date de traitement de la commande. Le client définit (dans des conditions qui ne sont pas pertinentes dans ce rapport) la pénalité par heure de retard de remplissage de son tanker. Le terminal propose une fenêtre de temps de trois jours dans laquelle le tanker peut arriver. En cas d'arrivée tardive, le terminal sera exempt de toute pénalité en cas de retard de service. Par contre, si le client arrive dans la fenêtre de temps prédéfinie, le client 'a le droit' d'être servi immédiatement. Dans la pratique, une pénalité sera exigée si le terminal n'a pas effectué le service dans les 36 heures. Plus précisément, les délais maxima de service sont de 36 heures pour chargement et de 4 heures pour documentation administrative. Ici, seule la phase de chargement est considérée dans l'ordonnancement sans aucune perte de généralité. Aucune autre information n'est retenue concernant d'autres scénarios d'arrivée des tankers. Cependant, si le service est retardé (i.e. dépasse les 36 heures) du fait du tanker, une pénalité de sur-occupation de quai peut être demandé au client et si, par contre, le service est retardé du fait de conditions météorologiques aucune pénalité n'est exigée pour l'une ou l'autre des parties.

2.2.1.2 Partage de Ressources

Selon le type et le séquençement des alignements demandés, les vannes, ressources du réseau, seront sollicitées de manière différente. Les alignements réservoir-tanker sont obtenus du fait des états d'ouverture/fermeture des vannes du pipeline. Des conflits surviennent lors de leur sollicitation pour des opérations simultanées et lors de demandes antagonistes de commutation. Ces notions sont pertinentes pour une large gamme de systèmes interconnectés et pose des problèmes critiques de partage de ressources.

A. Conflits entre plusieurs opérations de transfert

Définition 2.1. Plusieurs alignements affectés à des opérations de transfert sont dits en conflit s'ils partagent pendant une même période au moins une vanne ouverte commune ou si cette vanne doit commuter vers un état différent pour l'ensemble des alignements à considérer.

Autrement dit, si une vanne commune pour différents alignements doit commuter à l'état de fermeture pour tous les alignements considérés aucun conflit n'est identifié car cette commutation permet d'exécuter les opérations de manière simultanée. Dans tout autre cas, un conflit de partage de ressources existe. La Fig. 2.1a) présente deux alignements disjoints visant à satisfaire les requêtes (R_1 et R_3). Les lignes continues dénotent les vannes qui doivent être ouvertes et les

2. cet ordonnancement est stratégique à résoudre et il n'est pas l'objectif de cette recherche.

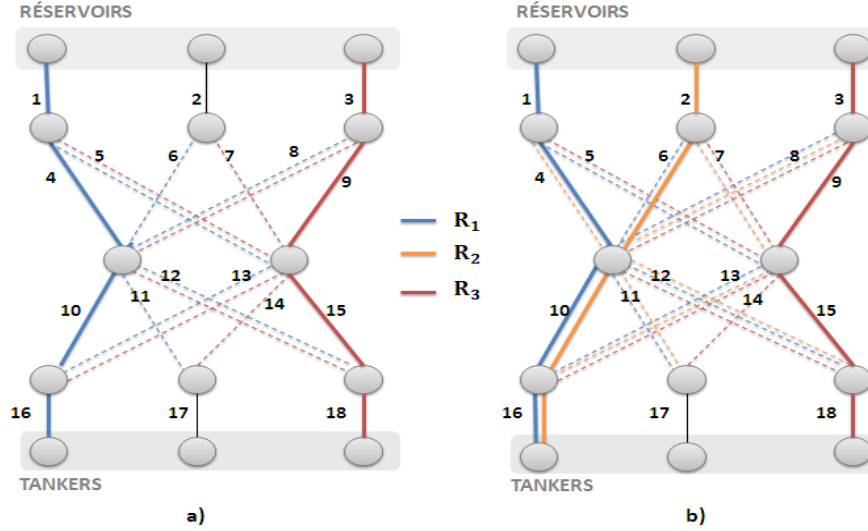


FIGURE 2.1: Alignements indépendants (a) et en conflit (b) pour les opérations de transfert

lignes segmentées les vannes qui doivent être fermées pour isoler les alignements. Par exemple, pour habilitier l'alignement pour R_1 les vannes 1, 4, 10, et 16 doivent être ouvertes et les vannes 5, 6, 8, 12, 11, et 13 doivent être fermées. Sur la Fig. 2.1a) aucun conflit n'existe car les ressources partagées par les opérations (i.e. vannes 5, 8, 12 et 13) sont toutes des vannes à fermer; par conséquent, elles peuvent permettre l'habilitation des deux alignements simultanément. Sur la Fig. 2.1b) une troisième requête est rajoutée et des conflits surviennent pour les vannes 10 et 16, puisqu'elles doivent être ouvertes pour deux opérations de transfert (conduisant à un mélange des produits) et pour les vannes 4 et 6 puisque les commutations requises sont différentes pour les deux opérations (ce qui est physiquement impossible). Par conséquent, R_1 et R_2 ne peuvent être traitées simultanément. Naturellement, il est primordiale de servir autant de clients que possible au plus tôt, ce qui se traduit par l'exécution simultanée d'opérations chaque fois que possible avec l'objectif de minimiser les pénalités.

B. Conflits entre les opérations de transfert et les opérations de maintenance

Définition 2.2. Une vanne est dite en conflit si habilitant le flux de produit sur un alignement ou l'isolant elle est également l'objet d'une opération de maintenance.

Dans la Fig. 2.1b), si la vanne 6 était l'objet d'une opération de maintenance, cela entraînerait un conflit avec l'opération R_1 (la vanne doit être exploitée en état de fermeture) et aussi avec l'opération R_2 (la vanne est requise en état d'ouverture). Ainsi, en aucun cas une même vanne ne peut être mise en maintenance et servir pour des opérations de transfert.

Pour qu'une vanne puisse être mise en maintenance, elle doit être isolée du reste du réseau en impliquant la fermeture de toutes les vannes adjacentes. Dans cette situation, il faut garantir qu'aucun conflit n'existe entre les vannes d'isolement pour l'opération de maintenance, et

les vannes d'ouverture pour un alignement d'une opération de transfert. Cela est garanti en respectant la définition 2.3.

Définition 2.3. Pour une vanne, le conflit entre sa sollicitation comme vanne d'isolement pour une opération de maintenance et comme vanne ouverte pour une opération de transfert générera toujours un conflit entre la vanne en maintenance et une vanne d'isolement pour l'opération de transfert.

Pour illustrer le principe de la définition 2.3, sur la Fig. 2.1b) si la vanne 7 devait être mise en maintenance, alors les vannes 2, 6, 5, 9, 13, 14 et 15 seraient les vannes d'isolement de cette opération et, par exemple, la vanne 2 ne pourrait pas être sollicitée simultanément pour satisfaire R_2 . Néanmoins, ce conflit d'opérations entraîné par la vanne 2 (de type fermeture pour maintenance / ouverture pour opération de transfert) est implicitement modélisé par le conflit entraîné par la vanne adjacente 7 (de type fermeture pour l'opération de transfert R_2 / mise en maintenance). Ce dernier type de conflit a déjà été décrit dans la définition 2.2.

Comme mentionné dans le chapitre 1, de nombreuses approches relevant des SED pourraient modéliser ces phénomènes. Dans ce contexte, les réseaux de Petri permettent une représentation graphique très intuitive des phénomènes de synchronisation.

Dans la Fig. 2.2, deux phénomènes de base (synchronisation et conflit) sont représentés à l'aide des réseaux de Petri. Graphiquement, les réseaux de Petri sont constitués par d'éléments interconnectés de type 'place' et 'transition'. Les jetons dans les places peuvent s'assimiler à des ressources ou des étapes en cours de traitement dans un système quelconque lorsque les transitions représentent l'occurrence d'événements³.

La Fig. 2.2a) modélise un phénomène de synchronisation sur la transition x_4 . La synchronisation peut modéliser des contraintes de séquence d'opérations, des dépendances de matière première dans les systèmes de production, ou des phénomènes similaires transposables à d'autres problématiques. Le franchissement de la transition x_4 dépend de la disponibilité de ressources (jetons) disposées dans les places en amont. Dans le cas considéré, la transition n'est pas franchissable puisqu'une place en amont ne contient pas de ressources disponibles. Ainsi, ce modèle représente la dépendance d'une ou plusieurs opérations, ressources ou événements. Ce même phénomène peut être représenté comme une contrainte mise sous la forme d'une équation ou inéquation (selon les besoins de modélisation). Si x_i est la date de franchissement de chaque transition, l'une des possibles représentations de cette contrainte serait $x_4 = \max(x_1, x_2, x_3)$ si aucun temps de séjour n'est exigé dans les places en amont. En l'algèbre conventionnelle, la fonction de maximisation est une fonction non linéaire, qui dans l'algèbre $(\max, +)$ devient linéaire, i.e.

3. des modèles plus élaborés basés sur les réseaux de Petri peuvent inclure des temporisations, le déplacement de plusieurs jetons lors du franchissement d'une transition, des phénomènes stochastiques ou des contraintes supplémentaires. La finalité de cette étude n'est pas une analyse exhaustive dans ce sujet et, ainsi, la version la plus simple de ce type de modèle est abordée pour l'illustration sans perte de généralité.

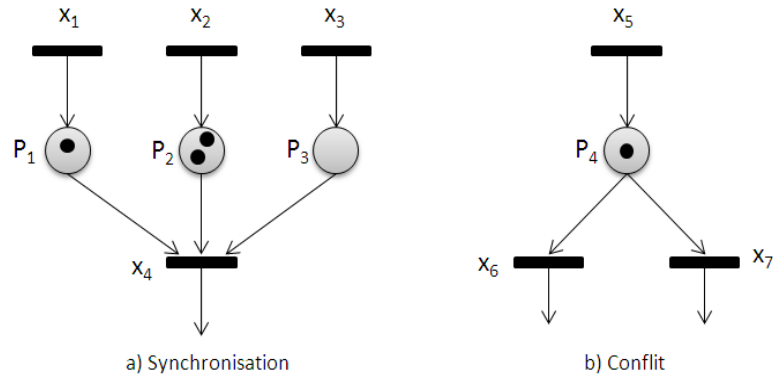


FIGURE 2.2: Synchronisation et conflit dans les réseaux de Petri

$x_4 = x_1 \oplus x_2 \oplus x_3$. Sachant que dans la pratique les modèles d'optimisation sont complexes et présentent de nombreuses non linéarités, des heuristiques sont utilisées pour leur résolution permettant d'obtenir des solutions approximées acceptables dans un délai raisonnable. Le phénomène de conflit, représenté dans la Fig.2.2b), modélise le fait que si la place P_4 contient un jeton, une seule des transitions en aval peut être franchie avec cette ressource. Néanmoins, ce choix n'est pas évident. Le choix correspondra mathématiquement à l'affectation d'une variable de décision jugé sur un critère spécifique. Pour ces deux phénomènes, les avantages d'une modélisation basée sur l'algèbre $(\max, +)$ seront une représentation mathématique simple et concise du problème d'optimisation de l'ordonnancement d'opérations.

2.2.2 Modèle Non Linéaire d'Optimisation de l'Ordonnancement d'Opérations

Ce modèle, basé sur l'algèbre $(\max, +)$, décrit de manière intuitive et concise les conflits de partage de ressources ainsi que la dépendance des dates d'arrivée des tankers sur la forme d'une synchronisation.

Pour le terminal maritime, chaque alignement (réalisé par la commutation de plusieurs vannes) répond à une opération de transfert. Dans un horizon de temps donné, l'exploitant du terminal peut être amené à gérer simultanément n opérations de transfert (soit n alignements). Ainsi, en terme d'allocation de ressources puisqu'il s'agira de répondre aux requêtes dans les temps impartis, les principes du modèle se basent sur une représentation de type dateur par vanne ou de type dateur par opération de transfert. Par la suite, le modèle d'un dateur par transfert sera retenu. La représentation d'un dateur par vanne a par ailleurs été objet de publication dans [59]. Ces deux représentations sont équivalentes, donnent les mêmes résultats, et les différences se situent au niveau de granularité retenu qui peut s'ajuster de manière plus adéquate en fonction de l'application et des besoins spécifiques.

2.2.2.1 Modèle Non Linéaire

Hypothèse 2.1. L'alignement affecté à chaque opération est connu au préalable, et, par conséquent, les conflits de partage de ressources sous-jacents.

Le dateur pour une opération de transfert est dénoté par x_i où i est la requête à traiter. Le dateur pour une opération de maintenance est dénoté par x_{m_k} où k est la vanne à être maintenue. $\mathcal{M}$ dénote l'ensemble des opérations de maintenance programmées au préalable (i.e. si une vanne k doit subir une opération de maintenance, alors $k \in \mathcal{M}$), $\mathcal{J}$ dénote l'ensemble des opérations de transfert (i.e. si i est une opération de transfert à être exécutée, alors $i \in \mathcal{J}$), et finalement ISO_k dénote l'ensemble des vannes d'isolement permettant d'exécuter la maintenance de la vanne k .

$$x_i = \max\left(t_0 ; u_i ; \max_k(x_{m_k} + tm_k + ztm_k + V_{i,k}) ; \max_{i'}(x_{i'} + p_{i'} + zp_{i'} + zc_{i'} + V_{i,i'})\right) \\ \forall i, i' \in \mathcal{J} | i \neq i', \forall k \in \mathcal{M} \quad (2.2)$$

$$x_i = t_0 \oplus u_i \oplus \left(\bigoplus_k (x_{m_k} \otimes tm_k \otimes ztm_k \otimes V_{i,k})\right) \oplus \left(\bigoplus_{i'} (x_{i'} \otimes p_{i'} \otimes zp_{i'} \otimes zc_{i'} \otimes V_{i,i'})\right) \\ \forall i, i' \in \mathcal{J} | i \neq i', \forall k \in \mathcal{M} \quad (2.3)$$

La première contrainte du modèle d'optimisation correspond à la date de début des opérations de transfert. Elle est représentée par (2.2) en algèbre conventionnelle et (2.3) correspond à son équivalent en notation d'algèbre $(\max, +)$. Dans ce qui suit, la notation $(\max, +)$ sera employée et dans le cas contraire il sera précisé. Cette contrainte détermine la date de début x_i , aussi nommée 'dateur' dans le contexte $(\max, +)$, pour l'opération i de transfert de pétrole qui s'exécute sur un alignement qui est supposé comme connu au préalable. La connaissance de l'alignement pour chaque opération implique que lorsque l'opération i débute, un ensemble de commutations de vanne sont réalisées pour habilitier l'alignement dans le réseau en cohérence avec l'hypothèse 2.2.

Hypothèse 2.2. Toutes les vannes dans chaque alignement commutent simultanément.

Les éléments dans (2.3) sont :

x_i : dateur pour l'opération de transfert i .

x_{m_k} : dateur pour une opération de maintenance sur la vanne k . Cette opération de maintenance est en conflit avec l'opération de transfert i si la vanne k est commune aux deux opérations.

$x_{i'}$: dateur pour une opération de transfert i' en conflit avec l'opération de transfert i .

$V_{i,k}$: variable de décision binaire qui résout le conflit d'affectation de ressources entre les opérations i (opération de transfert) et k (opération de maintenance) où $V_{i,k} \in \{\varepsilon, e\}$. Autrement dit, dans (2.3), si $V_{i,k} = \varepsilon = -\infty$ alors le terme $(x_{m_k} \otimes tm_k \otimes ztm_k \otimes V_{i,k})$ tend vers $-\infty$ et, par conséquent, il est négligeable dans l'équation (puisqu'il s'agit d'une maximisation). Cela implique que x_i ne dépend pas de la date d'achèvement de l'opération de maintenance de la vanne k , i.e. l'opération de transfert i s'exécutera avant. Le cas opposé est représenté pour $V_{i,k} = e$. Pour toute variable $V_{i,k}$ il existe une variable complémentaire $V_{k,i}$.

$V_{i,i'}$: variable de décision binaire qui résout le conflit d'affectation des ressources entre les opérations de transfert en conflit i et i' . $V_{i,i'} \in \{\varepsilon, e\}$. Son impact temporel est identique au cas précédent, et pour toute $V_{i,i'}$, il existe une variable complémentaire $V_{i',i}$.

u_i : date d'arrivée pour le tanker demandeur de l'opération de transfert i .

$ztm_k, zp_{i'}$, et $zc_{i'}$ représentent respectivement les retards possibles dans l'opération de maintenance, dans le service au client dû à des difficultés techniques dans le terminal, et dans le service du client du fait de difficultés techniques du tanker.

Enfin, les paramètres : t_0, tm_k , et $p_{i'}$ correspondent respectivement à la date de début de l'horizon d'ordonnancement, et aux durées nominales pour l'activité de maintenance et l'opération de transfert. Ici, t_0 prend sa valeur par défaut, i.e. $t_0 = 0$ dans tous les modèles.

L'équation (2.3) modélise que le fait d'habilitier un alignement pour répondre à une opération de transfert i dépend du début de l'horizon du temps d'ordonnancement, de l'arrivée du client au terminal, et de l'achèvement des opérations en conflit (de transfert ou de maintenance qui doivent s'exécuter avant l'opération ciblée i). Ainsi, le dateur x_i dépend du fait que toutes ces conditions aient eu lieu, il faut donc attendre que la dernière des conditions soit vraie pour pouvoir démarrer l'opération de transfert i . Ce comportement est modélisé par l'opérateur de maximisation des éléments de l'équation.

La structure de la contrainte de date (2.3) est la base de la modélisation. Les valeurs affectées aux variables de décision permettent de quantifier les dateurs, qui équivalent aux ordonnancements. L'équation est ainsi non linéaire dans l'algèbre $(\max, +)$ étant donné qu'il existe un produit des variables de décision et des dateurs ([7]), i.e. le produit du type dateur-variable de décision, e.g. $x_{i'} \otimes V_{i,i'}$.

Lors de l'exécution de l'opération i , plusieurs types d'interruption dans le service peuvent avoir lieu et permettront d'ajuster les données dans le modèle et de recalculer l'ordonnancement par rapport aux conditions opérationnelles actuelles du réseau.

Les variables de décision sont binaires et peuvent prendre les valeurs nulle ($\varepsilon = -\infty$) ou neutre ($e = 0$) dans l'algèbre $(\max, +)$. En vue de valider le modèle, les valeurs gérées sont e et B , tel que B soit un nombre réel grand et négatif (i.e. $B \rightarrow -\infty$). En outre, il est rappelé que chaque

variable de décision a une variable complémentaire (i.e. si $V_{i,i'} = e$, alors $V_{i',i} = B$ ou vice versa). L'effet des contraintes par rapport aux valeurs des variables de décision est le suivant : dans (2.3), si $V_{i,k} = B$ alors le 3^{eme} terme de la maximisation devient négligeable, ce qui implique que l'achèvement de l'opération de maintenance n'a aucune influence sur le début de l'opération de transfert i . Ainsi, l'opération i s'exécute avant l'opération de maintenance k . Au contraire, si $V_{i,k} = e$, alors le 3^{eme} terme de la maximisation est la date d'achèvement de l'opération de maintenance, ce qui implique que l'opération i dépend de cette date, i.e. l'opération i s'exécute après l'opération k .

$$x_{m_k} = x_{fixed_k} \oplus \left(\bigoplus_i (x_i \otimes p_i \otimes z p_i \otimes z c_i \otimes V_{k,i}) \right) \oplus \left(\bigoplus_{k'} (x_{m_{k'}} \otimes t m_{k'} \otimes z t m_{k'} \otimes V_{k,k'}) \right) \\ \forall i | i \in \mathcal{J}, \forall k \in \mathcal{M}, \forall k' \in ISO_k \quad (2.4)$$

De manière analogue, l'expression (2.4) permet de déterminer le dateur pour une opération de maintenance. Le résultat est le maximum de trois termes, dont le premier est une date (x_{fixed_k}) supposée fixée pour l'activité de maintenance (échancier). Dans le meilleur des scénarios, l'ordonnancement global d'opérations permet l'exécution des activités de maintenance dans les dates prévues sans changement. Le deuxième terme de la maximisation modélise le conflit de l'activité de maintenance avec des opérations de transfert sollicitant des ressources communes, et dont les commutations requises ne permettent pas d'effectuer les opérations de manière simultanée. Le troisième terme représente aussi des conflits de partage de ressources ; dans ce cas il s'agit des conflits entre la vanne en maintenance k et les possibles opérations de maintenance sur les vannes d'isolement pour k .

Hypothèse 2.3. Lorsqu'une opération de transfert débute, le client commence à recevoir le produit de manière instantanée, puisqu'il est considéré que les canalisations sont toujours pleines de pétrole (hypothèse 1.1).

Dans le cas où un délai important doit être considéré entre l'habilitation de l'alignement et la réception du produit par le client, il peut être facilement inclus dans les temps de traitement des requêtes dans (2.3) et (2.4).

$$V_{i,i'} \otimes V_{i',i} = B \quad \forall i, i' \in \mathcal{J} \quad (2.5)$$

$$V_{i,i'} \oplus V_{i',i} = e \quad \forall i, i' \in \mathcal{J} \quad (2.6)$$

Les équations (2.5) et (2.6) restreignent les valeurs des variables de décision à zéro ou B pour les conflits potentiels entre 2 opérations de transfert. De manière analogue, les équations (2.7) et (2.8) restreignent les valeurs des variables de décision pour les conflits entre une activité de maintenance et une opération de transfert.

$$V_{i,k} \otimes V_{k,i} = B \quad \forall i \in \mathcal{J}, \quad \forall k \in \mathcal{M} \quad (2.7)$$

$$V_{i,k} \oplus V_{k,i} = e \quad \forall i \in \mathcal{I}, \quad \forall k \in \mathcal{M} \quad (2.8)$$

$$D_i = \begin{cases} u_i \otimes p_{max} & \forall i | u_i \in tw_i \\ x_i \otimes p_{max} & \forall i | u_i > utw_i \\ ltw_i \otimes p_{max} & \forall i | u_i < ltw_i \end{cases} \quad (2.9)$$

L'équation (2.9) exprime la date limite de service D_i pour une requête i où $tw_i = [ltw_i, utw_i]$ définit la fenêtre de temps autorisée pour l'arrivée du tanker correspondant. Si le tanker arrive dans cet intervalle, sa date limite de service sera de $p_{max} = 36$ heures après l'arrivée. Si le tanker arrive après cette fenêtre, aucune pénalité n'est adressée au terminal pour le temps d'attente, qui sera recalculé en tenant compte des conditions opérationnelles du terminal. Aucune autre information supplémentaire n'a été obtenue directement des terminaux visités. Par conséquent, sans aucune perte de généralité, afin de valider le modèle proposé il est supposé que, dans ce scénario, la date limite de service est de $p_{max} = 36$ heures juste après le début de transfert. Finalement, il est supposé que si le tanker arrive avant la fenêtre autorisée, la date limite de service est également de 36 heures après le début de cette fenêtre.

$$dpr_i = (x_i \otimes p_i \otimes zp_i \otimes zc_i \oslash D_i) \oplus 0 \quad \forall i \in \mathcal{I} \quad (2.10)$$

Le retard par requête dpr , en anglais : *delay per request*, est déterminé par (2.10). Pour chaque requête, le dpr est la différence (opérateur $\oslash$) entre la date d'achèvement de la requête (considérant les possibles retards causés par le terminal et/ou le client) et sa date limite de service. En termes de politique de gestion des pénalités, les actions à appliquer lorsque les deux parties causent des retards dans le service simultanément ne sont pas encore connues. Afin de valider les modèles, l'hypothèse 2.4 est retenue, ce qui permet de modéliser par (2.11) le retard pénalisé pour le terminal pds , en anglais *penalized delay for the seaport*, pour chaque requête.

Hypothèse 2.4. La pénalité de sur-occupation de quai par heure (subie par chaque client) est considérée égale à la pénalité par heure subie par le terminal pour service tardif pour le même client.

$$pds_i = \begin{cases} \oslash [(\oslash zu_i \oslash zp_i \otimes zc_i) \oplus (\oslash dpr_i)] & \forall (zu_i \otimes zp_i) > zc_i \\ 0 & \text{sinon} \end{cases} \quad (2.11)$$

En algèbre conventionnelle, (2.11) indique que si le retard entraîné par le terminal (incluant le temps d'attente du tanker $zu_i = x_i - u_i$ et les interruptions de chargement) est supérieur aux interruptions causées par le client (i.e. $zu_i + zp_i > zc_i$), alors le terminal encourt une pénalité. Cette pénalité est calculée comme $\min((zu_i + zp_i) - zc_i, dpr_i)$ qui est le minimum entre la mesure dans laquelle le retard du terminal dépasse le retard du client et la mesure dans laquelle la date

limite de service est dépassée. Cette minimisation est exprimée sous la forme de maximisation par (2.11) en algèbre $(\max, +)$. Si le retard causé par le tanker est supérieur ou égal au retard causé par le terminal, alors le terminal n'encourt aucune pénalité.

$$\text{Min } TCP = \bigotimes_i \left(\bigotimes_{n=1}^{pds_i} c_i \right) \quad \forall i \in \mathcal{J} \quad (2.12)$$

L'équation (2.12) représente la fonction objectif dans le problème d'optimisation. Le coût total entraîné par les pénalités *TCP*, en anglais : *Total Cost due to Penalties*, est déterminé pour toutes les requêtes dans l'horizon de temps considéré. L'équation (2.12) décrit en algèbre $(\max, +)$ ce qui en algèbre conventionnelle serait la somme des produits pour chaque retard pénalisé (en heures) et sa pénalité correspondante (en \$/heure), i.e. $\sum_i pds_i \times c_i$.

2.2.2.2 Résultats

Le modèle proposé a été publié dans [59] avec la représentation d'un dateur par vanne (au lieu d'un dateur par alignement). Les instances ont été considérées sur une topologie simple, comme celle de la Fig. 1.1, pour une assimilation simple du processus de prise de décision de l'ordonnancement d'un ensemble d'opérations avec partage de ressources. Ainsi, il est possible d'estimer l'augmentation de la complexité de la prise de décision lorsque la taille du réseau augmente et que la topologie se complexifie.

L'instanciation a été réalisée avec le logiciel d'optimisation LINGO (voir [60]). La fonctionnalité de 'solver global' a été choisie pour garantir un optimum global et les contraintes et fonction objectif ont été directement codées dans cet environnement. Le solver teste des valeurs pour les variables de décision (qui par conséquent génèrent des valeurs pour les dateurs des opérations), et évalue la fonction objectif jusqu'à ce qu'elle ne puisse plus être améliorée, tout en respectant toutes les contraintes.

L'instance de la Fig. 2.3, qui correspond à un réseau de topologie simple, représente l'exécution de sept opérations de transfert, dénotées par R_i , $i = 1, \dots, 7$, et 2 activités de maintenance, dénotées par M_k , $k = 13, 15$, sur les vannes 13 et 15 planifiées aux dates $t = 100$ et $t = 130$ de durée de 12 et 10 heures, respectivement. Les alignements considérés sont réalisés par les vannes qui doivent être ouvertes pour l'écoulement et les vannes isolantes (omisées de la figure pour une meilleure compréhension). Pour les activités de maintenance, les vannes d'isolement sont également omises. Cette instance couvre l'ensemble des conflits possibles et les données d'entrée sont présentées sur le Tableau 2.1.

Sachant qu'une fenêtre de temps est autorisée pour l'arrivée de chaque tanker (représentée dans le Tableau 2.1 à partir de $t_0 = 0$), un ordonnancement de référence peut être obtenu en supposant

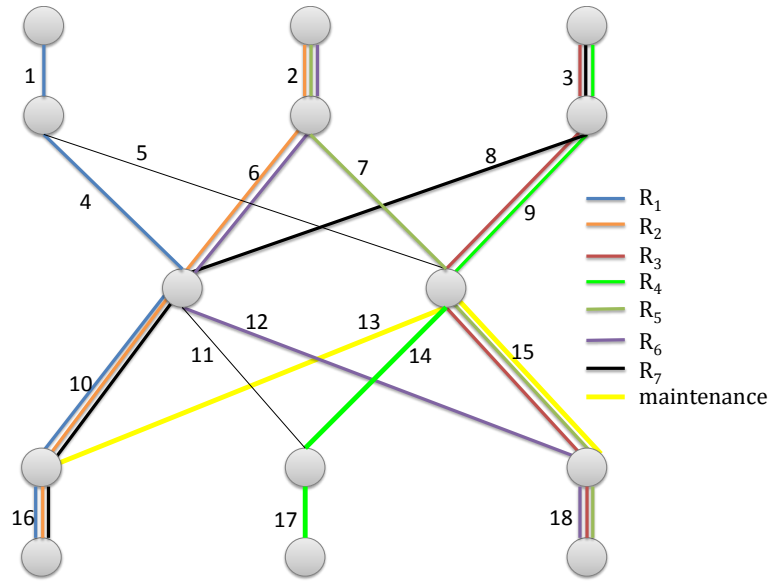


FIGURE 2.3: Opérations à ordonnancer

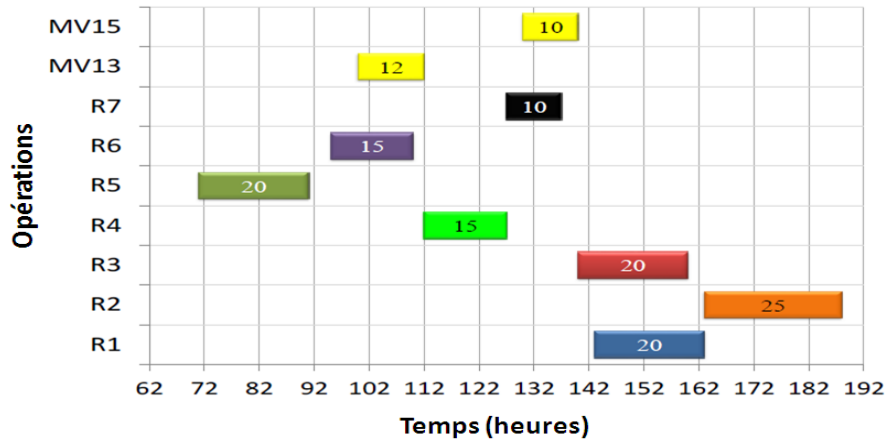
Requête	Temps de Traitement (heures)	Pénalité (\$/ heure)	Fenêtre de Temps pour Arrivée (jours)
R_1	20	4000	[4,6]
R_2	25	2500	[2,4]
R_3	20	3000	[2,4]
R_4	15	2500	[1,3]
R_5	20	2500	[1,3]
R_6	15	3000	[2,4]
R_7	10	2000	[3,5]

TABLE 2.1: Données d'entrée pour les opérations de transfert sur la Fig. 2.3.

certaines dates d'arrivée (dans ou hors la fenêtre) et basé sur les estimations actuelles par rapport aux échanges avec les clients. Pour valider la pertinence du modèle d'optimisation proposé, il est supposé que tous les tankers, sauf celui qui correspond à l'opération R_2 , arrivent dans leurs fenêtres de temps à la dernière heure du dernier jour permis, ce qui entraîne de nombreux conflits. Il est aussi supposé que le tanker pour R_2 arrive après sa fenêtre de temps, à 10 heures du matin du jour 5. Aucune interruption pouvant entraîner un retard n'est prévue, ce qui implique que $zp_i = zc_i = ztm_i = e$. Il est supposé que l'horizon de temps pour la planification commence à $t_0 = 0$.

L'ordonnancement optimum obtenu par LINGO est présenté sur la Fig. 2.4, il implique un TCP de \$137000.

Selon les données du Tableau 2.1, R_4 et R_5 sont en conflit puisque les tankers arrivent simultanément (à $t = 71$, début de la dernière heure de leur fenêtre) et les ressources communes qui doivent être sollicitées ne permettent pas une exécution simultanée (voir en référence à la Fig. 2.3 où la vanne 9 doit être ouverte pour R_4 mais fermée pour R_5). Dans l'ordonnancement obtenu, il peut être vérifié que la résolution optimale de ce conflit, minimisant les pénalités, est

FIGURE 2.4: Ordonnancement optimum - modèle $(\max, +)$ non linéaire

d'exécuter R_5 avant R_4 . Les opérations restantes peuvent être commentées de manière analogue. Par exemple, en considérant la fenêtre de temps $[2, 4]$ (i.e. arrivée entre les jours 2 et 4) dans le Tableau 2.1 il peut être identifié que les opérations R_3 et R_6 sont en conflit du fait des ressources communes et du fait que les deux tankers arrivent à $t = 95$ (dernière heure du dernier jour de leur fenêtre). R_2 n'est pas en conflit avec R_3 et R_6 puisqu'il a été considéré que le tanker arrive à 10 heures du matin du jour 5 ; i.e. à $t = 119$, hors la fenêtre de temps autorisée.

Ainsi, l'instance présentée force le terminal à subir des pénalités étant donné le nombre important de ressources en conflit, dates d'arrivée définies et des temps de traitement. Considérant la topologie relativement simple de la Fig. 2.3, il peut être vérifié manuellement qu'aucun autre ordonnancement ne génère un TCP inférieur. Naturellement, les opérations en conflit ne peuvent pas être exécutées simultanément et les dates des activités de maintenance doivent être respectées. Les requêtes qui entraînent des pénalités pour le terminal sont R_3 (retard de 29 heures) et R_4 (retard de 20 heures). Le terminal n'encourt aucune pénalité pour R_2 puisque le tanker arrive après sa fenêtre autorisée.

L'objectif de la formulation $(\max, +)$ a été de pouvoir minimiser le TCP pour répondre au besoin d'ordonnancement des terminaux maritimes. Sa validation a utilisé des données recueillies auprès de l'entreprise pétrolière PDVSA et un ordonnancement satisfaisant a été obtenu. Le modèle algébrique $(\max, +)$ proposé est concis puisqu'il permet d'agréger facilement dans une seule équation une diversité de contraintes. Cependant ce modèle est non linéaire car il n'existe pas d'a priori, ni sur les clients ni sur les ressources. Dans la section suivante, des modèles linéaires basés sur cette représentation de base seront proposés pour les cas où des informations supplémentaires posent des priorisations entre chaque paire des requêtes en conflit. Ainsi, à partir des modèles déjà proposés, une linéarisation résultera de l'affectation des variables de décision.

2.2.3 Exploration des Modèles Linéaires d'Optimisation de l'Ordonnement d'Opérations

Dans la section précédente, le modèle d'optimisation a été proposé sous un approche $(\max, +)$ tel que les contraintes fondamentales sont celles de l'affectation de ressources lors d'un conflit. Le modèle est non linéaire du fait du produit entre les dateurs et les variables de décision (voir (2.3)). Par exemple, le terme $(x_{i'} \otimes p_{i'} \otimes zp_{i'} \otimes zc_{i'} \otimes V_{i,i'})$ modélise une non linéarité en sachant que $x_{i'}$ et $V_{i,i'}$ sont des variables et que $p_{i'}$, $zp_{i'}$ et $zc_{i'}$ sont des paramètres. Ainsi, le produit $x_{i'} \otimes V_{i,i'}$ rend l'expression non linéaire. Dans ce cas, l'affectation d'une valeur à la variable de décision $V_{i,i'}$ rendrait l'expression linéaire ; i.e. si tous les paramètres sont regroupés dans un seul paramètre global $pg = p_{i'} \otimes zp_{i'} \otimes zc_{i'} \otimes V_{i,i'}$, alors l'expression linéaire est synthétisée comme $pg \otimes x_{i'}$.

La motivation de cette section est due à l'importance de l'exploitation des propriétés mathématiques des modèles $(\max, +)$ linéaires en général. L'affectation des variables de décision est abordée en considérant certains critères de priorisation des opérations de transfert en conflit. Ainsi, les opérations de maintenance ne sont pas considérées dans cette section. Pour les considérer il faudrait disposer des informations telles que l'ensemble de conséquences financières entraînées, dans la gestion de maintenance, par le décalage d'une opération de maintenance de vanne, informations qui ne sont pas connues dans cette étude. Dans le cas où ces informations seraient connues, une affectation des variables de décision de précedence entre les opérations de transfert et de maintenance en conflit pourrait être effectuée et ainsi le modèle deviendrait linéaire du type $X = AX \oplus b$ avec $X = (X_t, X_m)^T$ où X_t serait le vecteur ligne des dateurs des opérations de transfert et X_m correspondrait au vecteur ligne des dateurs des opérations de maintenance.

L'affectation des variables de décision est traité en terme de *politique de routage* dans les réseaux de Petri. Dans ces modèles, le routage fixe la priorité d'exécution de opérations qui ne peuvent pas être exécutées simultanément du fait des ressources partagées. Cette politique de routage peut résulter d'une intégration de plusieurs critères ou indicateurs liés aux opérations en conflit, ou aux conditions d'exploitation du réseau. Pour explorer des représentations linéaires du modèle $(\max, +)$ proposé, et en analogie avec la terminologie des réseaux de Petri, une politique de routage sera appliquée pour déterminer la précedence entre deux opérations quelconques en conflit. Deux possibles politiques ont été étudiées, elles sont présentées dans les sections qui suivent.

2.2.3.1 Modèle Linéaire basé sur la Pénalité Totale Potentielle

Cette représentation linéaire est inspirée par la possibilité d'estimer la précedence entre deux opérations de transfert en conflit référant aux pénalités potentielles que ces opérations pourraient

entraîner pour le terminal. La pénalité totale potentielle ou *TPP* (en anglais : *Total Potential Penalty*) est déterminée pour chaque opération de transfert comme le produit en algèbre classique du temps de traitement et du montant de la pénalité par unité de temps en cas de service tardif. Elle représente la pénalité due par le terminal dans le cas où le traitement de chaque requête débute après sa date limite.

Supposant que les tankers pour deux opérations de transfert en conflit i et i' arrivent simultanément au terminal et, suivant la même notation que précédemment, $u_i \in tw_i$, $u_{i'} \in tw_{i'}$ et $u_i = u_{i'}$, alors la pénalité totale potentielle s'exprime en algèbre conventionnelle par : $TPP_i = p_i \times c_i$ et $TPP_{i'} = p_{i'} \times c_{i'}$. La requête i précède la requête i' si et seulement si $TPP_i \geq TPP_{i'}$. Autrement dit, le risque potentiel de retarder le traitement de la requête i est supérieur au risque potentiel de retarder la requête i' .

Supposant qu'une priorisation peut être réalisée pour toute paire d'opérations en conflit, par exemple en comparant les *TPP*, l'affectation des valeurs aux variables de décision peut être effectuée, conduisant à un système $(\max, +)$ linéaire. La proposition du *TPP* comme différenciateur pour des opérations en conflit n'est pas restrictive et peut être ajustée aux critères d'intérêt selon l'application.

Pour illustrer la propriété de linéarité $(\max, +)$ du modèle, le même exemple de la Fig. 2.3 est considéré où seul les opérations de transfert sont prises en compte. Ainsi, considérant (2.3) et en supposant que l'horizon de planification est initialisé à $t_0 = 0$, les expressions des contraintes fondamentales non linéaires de partage de ressources pour cette instance sont définies par (2.13-2.19). L'opérateur du produit est omis (comme il est souvent fait dans l'algèbre classique) pour soulager la notation. Par exemple, dans (2.13), le dateur x_1 pour l'opération de transfert R_1 dans la Fig. 2.3 dépend de la date d'arrivée du tanker (u_1), et des conflits avec les opérations R_2 , R_6 et R_7 , pour lesquels l'affectation des variables de décision $V_{1,2}$, $V_{1,6}$ et $V_{1,7}$ déterminera les précédences correspondantes.

$$x_1 = u_1 \oplus x_2 p_2 V_{1,2} \oplus x_6 p_6 V_{1,6} \oplus x_7 p_7 V_{1,7} \quad (2.13)$$

$$x_2 = u_2 \oplus x_1 p_1 V_{2,1} \oplus x_5 p_5 V_{2,5} \oplus x_6 p_6 V_{2,6} \oplus x_7 p_7 V_{2,7} \quad (2.14)$$

$$x_3 = u_3 \oplus x_4 p_4 V_{3,4} \oplus x_5 p_5 V_{3,5} \oplus x_6 p_6 V_{3,6} \oplus x_7 p_7 V_{3,7} \quad (2.15)$$

$$x_4 = u_4 \oplus x_3 p_3 V_{4,3} \oplus x_5 p_5 V_{4,5} \oplus x_7 p_7 V_{4,7} \quad (2.16)$$

$$x_5 = u_5 \oplus x_2 p_2 V_{5,2} \oplus x_3 p_3 V_{5,3} \oplus x_4 p_4 V_{5,4} \oplus x_6 p_6 V_{5,6} \quad (2.17)$$

$$x_6 = u_6 \oplus x_1 p_1 V_{6,1} \oplus x_2 p_2 V_{6,2} \oplus x_3 p_3 V_{6,3} \oplus x_5 p_5 V_{6,5} \oplus x_7 p_7 V_{6,7} \quad (2.18)$$

$$x_7 = u_7 \oplus x_1 p_1 V_{7,1} \oplus x_2 p_2 V_{7,2} \oplus x_3 p_3 V_{7,3} \oplus x_4 p_4 V_{7,4} \oplus x_6 p_6 V_{7,6} \quad (2.19)$$

Pour illustration et validation, le pire des scénarios est considéré. C'est le cas où les tankers arrivent tous en même temps et dans leurs fenêtres de temps autorisées, e.g. $u_i = e$ et $u_i \in tw_i \forall i \in J$. Les données d'entrée sont celles proposées dans le Tableau 2.2. Pour une meilleure appréhension des résultats, les données ont été finxées telles que $\forall i | 0 < i < 7 \Rightarrow TPP_i < TPP_{i+1}$. L'affectation des valeurs des variables de décision est établie sur la priorisation des requêtes avec les TPP les plus importants et par rapport à la capacité opérative du réseau.

Il est rappelé [1] que la capacité opérative représente l'ensemble maximum d'alignements disjoints simultanés dans le réseau. Puisqu'il est important d'exécuter autant d'opérations simultanément que possible, les opérations avec un grand TPP et avec des possibilités d'exécution parallèle sont privilégiées dans l'ordonnancement.

TABLE 2.2: Données d'entrée pour validation du modèle basé sur le TPP

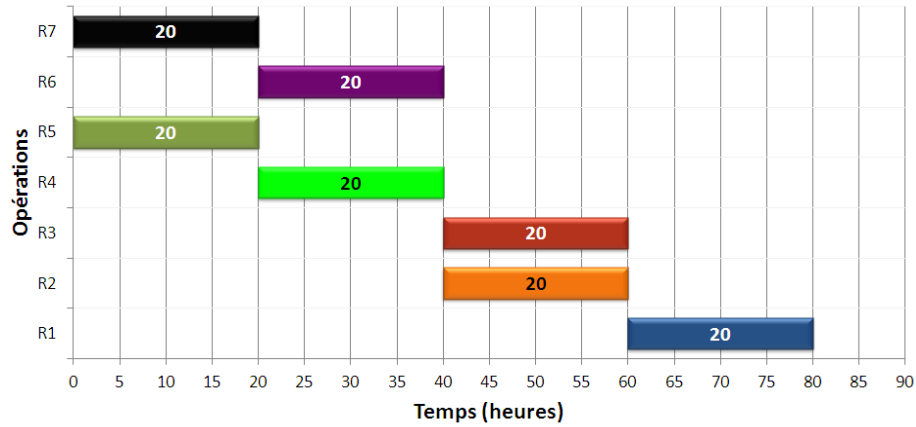
Requête	p_i (heures)	c_i (\$/heure)	TPP_i
R_1	20	1000	20000
R_2	20	2000	40000
R_3	20	3000	60000
R_4	20	4000	80000
R_5	20	5000	100000
R_6	20	6000	120000
R_7	20	7000	140000

Il est facile de déduire que la topologie de la Fig. 1.1 définit une capacité opérative de 2 alignements. Ainsi, l'ordonnancement démarrerait avec l'exécution de R_7 (opération du plus grand TPP) en parallèle avec l'opération R_5 (opération simultanée possible et de TPP élevé), en remarquant que si R_6 possède un TPP plus grand, elle ne peut être lancée simultanément avec R_7 .

La priorisation des opérations R_7 et R_5 conduit à affecter l'ensemble des variables de décision dans (2.17) et (2.19) par la valeur $\varepsilon = -\infty$. Ainsi, R_7 et R_5 ne dépendent de l'achèvement d'aucune autre opération. Cette affectation se propage dans le système d'équations en rappelant que pour chaque variable de décision $V_{i,i'}$ une variable $V_{i',i}$ prend sa valeur complémentaire. Par exemple, si dans (2.17) $V_{5,6} = \varepsilon$, alors dans (2.18) $V_{6,5} = e$.

Ensuite, les variables restantes sont déterminées par rapport au critère du TPP . Par exemple, dans (2.14) $V_{2,6} = e$ indique que R_2 dépend de la date d'achèvement de R_6 , puisque $TPP_6 > TPP_2$, ce conduit à ce que $V_{6,2} = \varepsilon$ dans (2.18). Après l'affectation des variables, le système non linéaire $(\max, +)$ (2.13-2.19) devient le système linéaire $(\max, +)$ (2.20-2.26).

$$x_1 = x_2 p_2 \oplus x_6 p_6 \oplus x_7 p_7 \oplus u_1 \quad (2.20)$$

FIGURE 2.5: Ordonnancement optimal pour le système $(\max, +)$ linéaire basé sur le TPP

$$A^* = \begin{pmatrix} e & 20 & . & . & 60 & 40 & 60 \\ . & e & . & . & 40 & 20 & 40 \\ . & . & e & 40 & 40 & 20 & 40 \\ . & . & . & e & 20 & . & 20 \\ . & . & . & . & e & . & . \\ . & . & . & . & 20 & e & 20 \\ . & . & . & . & . & . & e \end{pmatrix}$$

Par conséquent, la solution au système $(\max, +)$ linéaire, réponse à l'équation $X = A^*b$, est $X = (60 \ 40 \ 40 \ 20 \ e \ 20 \ e)^T$ et correspond à l'ordonnancement illustré par la Fig. 2.5.

Ces dateurs minimisent le TCP et sont obtenus par une simple manipulation d'algèbre linéaire. Le TCP est de \$204000, puisque toutes les dates limites de service sont égales à 36 heures, i.e. $D_i = 36 \ \forall i \in \{1, 2, \dots, 7\}$, et la capacité opérative reste exploitée au maximum (2 opérations exécutées simultanément par des alignements disjoints). La simplicité de l'exemple permet de valider intuitivement que les résultats obtenus correspondent bien à ceux souhaités. Pour des systèmes plus complexes, la procédure reste directe et permet d'obtenir facilement des dateurs (par des produits matriciels $(\max, +)$) pour un large ensemble d'opérations avec des conflits de partage de ressources. Naturellement, et à différence de l'exemple présenté, l'estimation empirique des dateurs pour des systèmes plus complexes n'est pas envisageable par un opérateur de supervision et c'est là que réside l'attrait de l'approche proposée.

Ainsi, le modèle présenté exploite une approche intuitive relevant de l'algèbre $(\max, +)$, pour la prise en considération des dates d'arrivée des clients et de nombreux conflits de partage de ressources. Le modèle obtenu intègre ainsi une formulation unique de contraintes. En outre, à partir d'une estimation des pénalités potentielles, le modèle linéaire réduit le problème d'optimisation à un produit matriciel $(\max, +)$. La proposition non linéaire nécessite d'un algorithme

déterminant la meilleure affectation des variables par rapport à la fonction objectif. Le logiciel LINGO (voir [60]) a été utilisé dans ce sens.

Les résultats de cette section sur le modèle $(\max, +)$ linéaire basé sur le *TPP* ont été publiés dans [62].

2.2.3.2 Modèle Linéaire basé sur la Criticité

Cette section propose un modèle linéaire basé sur le critère de criticité⁴ de la réalisation des alignements pour l'affectation des variables de décision visant à linéariser les contraintes exprimées par (2.3).

Ici, la “politique de routage” vise à intégrer les conséquences financières directes induites à la fois de l'ordre d'exécution d'opérations en conflit, mais aussi de la disponibilité des ressources impliquées. L'indisponibilité de chaque alignement affecté à une opération est considérée pour estimer les coûts qui pourrait résulter des pénalités potentielles d'un service tardif. Ainsi, une priorisation pourra être établie entre toute paire d'opérations en conflit en se basant sur la criticité.

La criticité d'une indisponibilité dépend de sa probabilité d'occurrence ainsi que de sa gravité [63]. Dans le contexte abordé, la gravité d'indisponibilité d'un alignement (pour une opération de transfert spécifique) dépend des conséquences financières liées aux pénalités de service tardif pour cette opération. [63] peut être consulté pour une synthèse sur les notions de fiabilité, disponibilité, maintenabilité et sécurité des systèmes industriels. L'intérêt se focalise sur l'aptitude de l'alignement à fonctionner dans l'instant de temps dans lequel il est sollicité. Notamment, son aptitude à fonctionner sachant que le système n'est pas dans sa phase initiale d'exploitation; la notion de *disponibilité asymptotique* des vannes appartenant aux alignements en question est considérée. Il s'agit alors d'une disponibilité à un instant donné de sollicitation de ressources.

La criticité permet de considérer des systèmes dont le fonctionnement dépend fortement de la disponibilité instantanée de ses ressources, et par définition de la qualité et des moyens engagés pour leur maintenance. La disponibilité peut résulter des conditions opérationnelles, par exemple dans le cadre d'une ‘production forcée’ où les activités de maintenance sont décalées du fait d'exigences de production fortes et dont la prise de décision d'ordonnancement dépend de la disponibilité des ressources. Dans ce contexte, l'indisponibilité d'un alignement affecté à une opération entraîne un coût en termes du retard occasionné et donc des pénalités pour le terminal. Cette politique de routage est inspirée des travaux [64], où les activités de maintenance sont priorisées par rapport à un indicateur basé sur l'indisponibilité de la ressource et des coûts impliqués. Notamment, dans ces travaux, un indicateur appelé le *CBC* ou indicateur de criticité

4. relation défaillance-conséquence financière.

basée sur le coût (en anglais : *Cost-Based Criticality*) est utilisé pour classer les opérations de maintenance. Le *CBC* est ainsi défini par ressource comme le produit en algèbre classique de l'indisponibilité et de ses conséquences financières. Parmi les possibles conséquences, on note sans exhaustivité : l'impact environnemental, les pertes de production et les pertes de qualité.

Un indicateur similaire est proposé, permettant de réaliser une affectation des variables de décision au modèle $(\max, +)$ non linéaire. Les dateurs restent déterminés alors par des manipulations simples du produit matriciel $(\max, +)$.

L'indisponibilité de chaque alignement est calculée en fonction de l'indisponibilité de ses vannes comme dans les travaux de [65] et [4]. La conséquence financière est considérée comme le *TPP* pour chaque client. L'indicateur intégrant les aspects de coût et d'indisponibilité proposé est le *PCI* (en anglais : *Penalty-based Criticality Index*), voir (2.27). $\overline{A}_i$ est l'indisponibilité de l'alignement affecté à l'opération de transfert i .

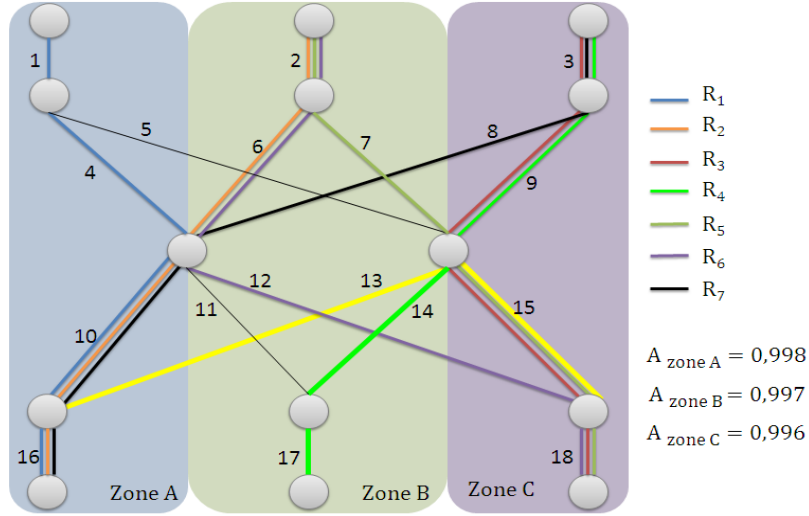
$$PCI_i = TPP_i \times \overline{A}_i \quad (2.27)$$

Indisponibilité d'un Alignement

Dans [4], une approche a été proposée pour déterminer le nombre maximum d'alignements disjoints (capacité opérative) tout en minimisant le risque de non service. Cette détermination repose sur un algorithme de flux maximum et de coût minimum. Le coût est lié à la disponibilité des ressources et le flux est considéré comme existant ou inexistant dans les segments de pipeline.

Pour qu'un alignement puisse être habilité pour une opération de transfert, l'ensemble des vannes d'écoulement doivent commuter vers l'état 'ouvert' et les vannes d'isolement, commuter vers l'état 'fermé'. Considérant que la disponibilité lors de chaque commutation de vanne est indépendante de toutes les autres et qu'elle est décrite par une variable aléatoire, une estimation de la disponibilité d'un alignement peut être décrite comme le produit, en algèbre conventionnelle, des disponibilités locales. Cet indicateur est défini par (2.28), où $v \in Algn_i$ dénote les vannes impliquées dans l'habilitation de l'alignement $Algn_i$.

Dans le scénario présenté par la Fig. 2.1a), en considérant l'alignement répondant à la requête R_1 , la disponibilité des ressources (A_i où $i = R_1$) est donnée par $A_{R_1} = A_1 \times A_4 \times A_{10} \times A_{16} \times A_5 \times A_6 \times A_8 \times A_{12} \times A_{11} \times A_{13}$. Ceci représente la disponibilité de l'alignement à partir de la disponibilité des vannes à ouvrir (i.e. 1, 4, 10 et 16) et des vannes à fermer (i.e. 5, 6, 8, 12, 11 et 13). Par conséquent, l'indisponibilité est définie de manière générique comme $\overline{A}_i = 1 - A_i$. Il est fondamental de comprendre que le rôle de cet indicateur est de différencier la qualité des alignements selon leur capacité à exécuter un ensemble donné de requêtes.

FIGURE 2.6: Exemple pour validation du modèle $(\max, +)$ linéaire basé sur le PCI .TABLE 2.3: Données de validation du modèle $(\max, +)$ linéaire basé sur le PCI .

Requête	p_i (heures)	c_i (\$/heure)	TPP_i	$\bar{A}_i$	PCI_i
R_1	20	3000	60000	0,0257	154,2
R_2	20	3000	60000	0,0277	166,2
R_3	20	3000	60000	0,0335	201,0
R_4	20	2000	40000	0,0315	126,0
R_5	20	2000	40000	0,0315	126,0
R_6	20	4000	80000	0,0315	252,0
R_7	20	4000	80000	0,0296	236,8

$$A_i = \prod_{(v \in \text{Algn}_i)} A_v \quad (2.28)$$

Dans la pratique, les valeurs de disponibilité peuvent être issues du monitoring si l'instrumentation le permet. En outre, d'autres niveaux de criticité établis par les requêtes pourraient également prioriser les opérations de transfert selon les stratégies de maintenance considérées.

Résultats

La Fig. 2.6 reprend l'exemple d'exécution de sept opérations de transfert. Dans cette figure, 3 zones sont identifiées : A, B et C, elles définissent 3 valeurs différentes de disponibilité de vannes. A priori, les vannes sont affectées d'un niveau de disponibilité asymptotique identique dans leur zone. Les vannes interzones sont caractérisées de la façon arbitraire suivante : vannes 5 et 13 $\in$ Zone A, et vannes 8 et 12 $\in$ Zone C. Les contraintes de conflits de partage de ressources pour cet exemple sont les mêmes que dans (2.13-2.19) et l'origine de l'horizon de planification est considéré à $t_0 = 0$.

Le Tableau 2.3 présente les données d'entrée pour l'exemple considéré où la dernière colonne présente l'indisponibilité résultante de chaque alignement. L'alignement pour l'opération R_3 présente une indisponibilité supérieure à celle de l'alignement pour R_1 en référence aux données de la Fig. 2.6, où il est évident que l'alignement pour R_3 se trouve dans la zone avec les dispositifs les moins disponibles.

Dans ce tableau, les temps de traitement sont supposés être égaux pour l'ensemble des opérations afin d'illustrer plus intuitivement les résultats. Avec ces données d'entrée, le vecteur d'indicateurs PCI obtenu correspond à : $PCI = [154, 2 \quad 166, 2 \quad 201, 0 \quad 126, 0 \quad 126, 0 \quad 252, 0 \quad 236, 8]^T$.

Supposant le scénario le plus serré, où tous les tankers arrivent en même temps et dans les fenêtres de temps autorisées, tous les conflits potentiels de partage de ressources deviennent réels. Le PCI permet de résoudre ces conflits sur la base d'un système d'équations $(\max, +)$ linéaire. Pour les valeurs de PCI obtenues, l'opération R_6 est définie comme étant la plus critique. Par conséquent, cette opération doit être lancée le plus tôt possible et ne doit pas dépendre de la date d'achèvement d'aucune autre opération. Par conséquent $V_{6,i} = \varepsilon$ pour toute opération en conflit i .

Ainsi, en sachant que les conflits de partage de ressources sont décrits par l'ensemble d'équations (2.13-2.19), l'affectation des variables de décision pour les conflits liés à R_6 correspond à : $V_{6,1} = V_{6,2} = V_{6,3} = V_{6,5} = V_{6,7} = \varepsilon$ dans (2.18), ce qui entraîne l'affectation des variables complémentaires dans toutes les autres équations, i.e. $V_{1,6} = V_{2,6} = V_{3,6} = V_{5,6} = V_{7,6} = e$. De manière analogue, toutes les variables de décision restantes prennent leurs valeurs par rapport au critère de PCI . Ainsi, dans (2.13) $V_{1,2} = e$ puisque l'opération R_1 dépend de la date d'achèvement R_2 , du fait que $PCI_2 > PCI_1$, conduisant à $V_{2,1} = \varepsilon$ dans (2.14), et ainsi de suite. Ainsi, le système $(\max, +)$ linéaire suivant est obtenu :

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} = \begin{pmatrix} . & p_2 & . & . & . & p_6 & p_7 \\ . & . & . & . & . & p_6 & p_7 \\ . & . & . & . & . & p_6 & p_7 \\ . & . & p_3 & . & . & . & p_7 \\ . & p_2 & p_3 & p_4 & . & p_6 & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & p_6 & . \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{pmatrix} \oplus \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{pmatrix}$$

Pour simplicité de compréhension, si les dates d'arrivée des tankers sont égales à zéro, i.e. arrivée des tankers dès la date au plus tôt, alors $u = [e, e, e, e, e, e, e]^T$ et la solution au système $(\max, +)$ linéaire déduit de la relation $X = A*b$ comme détaillé dans la section précédente. L'ordonnancement obtenu est présenté en Fig. 2.7.

L'affectation de ressources est réalisée par rapport aux conditions d'exploitation du réseau et aux coûts potentiels dérivant du dysfonctionnement des ressources. Il faut remarquer que pour des scénarios différents à celui présenté en Fig. 2.7 certaines opérations pourraient être exécutées simultanément, ce qui réduirait le temps global d'achèvement des opérations. Néanmoins, cela serait au détriment de la disponibilité des ressources et le risque pour certaines opérations prioritaires serait accru. Par exemple, dans d'autres scénarios non optimaux par rapport au *PCI*, R_4 pourrait être exécutée à $t_0 = 0$ simultanément avec R_6 , mais ceci réduirait la disponibilité pour l'exécution ultérieure de R_7 qui a une priorité plus élevée.

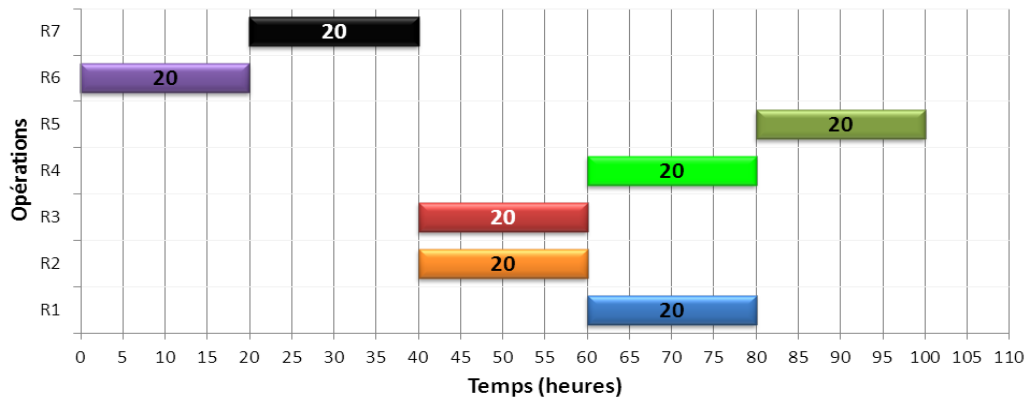


FIGURE 2.7: Ordonnancement optimum pour le modèle $(\max, +)$ linéaire basé sur le *PCI*

Ce scénario privilégiant le *PCI* vise à souligner l'importance des spécifications et des conditions opérationnelles de chaque opération.

L'exploitation d'une modélisation $(\max, +)$ reste suffisamment intuitive pour pouvoir synthétiser les contraintes fondamentales de fonctionnement en une relation unique. Elle permet d'exprimer la meilleure politique de priorisation entre toute paire d'opérations en conflit sur un compromis coût/disponibilité. Les résultats de cette approche ont été publiés dans [66].

Les compromis les plus adéquats devront être choisis selon les caractéristiques et conditions d'exploitation. L'approche $(\max, +)$ linéaire, par pré-affectation des variables de décision, permet ainsi de manipuler des pénalités potentielles, des notions de criticité, ou sur tout autre critère de priorisation entre deux opérations en conflit. Par ailleurs, si aucun critère ne peut être envisagé a priori on revient vers un modèle $(\max, +)$ non linéaire (2.3), pour lequel l'affectation des variables de décision repose sur des techniques approximatives. À ce titre, une proposition basée sur les algorithmes génétiques est présentée dans le chapitre suivant.

Des modèles $(\max, +)$ non linéaires, ainsi que linéaires, ont été proposés abordant le problème d'ordonnancement d'opérations par minimisation des pénalités sous l'hypothèse d'alignements prédéfinis. La section suivante aborde une extension du modèle $(\max, +)$ non linéaire obtenu pour l'intégration de la sélection d'alignements. Le modèle se complexifie, en incluant un aspect

supplémentaire de la prise de décision des opérateurs. Néanmoins, l'exploitation $(\max, +)$ rend toujours cette manipulation assez directe.

2.2.4 Modèle d'Optimisation de l'Ordonnancement d'Opérations et de Sélection d'Alignements

La modélisation synthétisant plusieurs types de contraintes d'algèbre conventionnelle dans une seule relation $(\max, +)$ a permis l'inclusion d'autres aspects de prise de décision.

Dans (2.3), la contrainte fondamentale du problème d'optimisation est considérée en supposant que l'alignement pour chaque opération de transfert est connu au préalable (hypothèse 2.1) et que les variables de décision sont celles liées au partage de ressources. Dans le cadre réel de la prise de décision dans les terminaux maritimes, l'opérateur de supervision doit également choisir, à partir d'un ensemble d'opérations de transfert, l'alignement adéquat dans le réseau pour chacune de ces opérations. La sélection d'un alignement pour une opération modifie la disponibilité du réseau pour les opérations restantes. L'ensemble de ces spécifications et contraintes temporelles différentes, fait que la sélection globale n'est pas évidente. Pratiquement, la sélection d'alignements résulte de l'expertise de l'opérateur. Pour une prise de décision qui tienne compte de ces nombreux aspects opérationnels et de leurs conséquences sur les délais de service des clients, un modèle $(\max, +)$ non linéaire est une solution convenable.

Ce modèle est issu d'une simple mais importante manipulation mathématique qui permet d'étendre la contrainte fondamentale du problème d'optimisation et de trouver des ordonnancements optima qui minimisent les pénalités globales et qui suggèrent les meilleurs alignements pour l'ensemble des opérations envisagées. Ainsi, un degré de flexibilité est ajouté au modèle initial, puisque chaque opération n'est plus restreinte à être exécutée sur un alignement déterminé. Une reconfiguration du réseau est alors réalisée pour minimiser les coûts de l'ensemble des opérations.

L'objectif demeure la minimisation du TCP pour un ensemble d'opérations de transfert J . Pour chaque requête, un alignement est sélectionné parmi un ensemble d'alignements candidats qui satisfont les spécifications de la requête. Un alignement est dit candidat pour une opération de transfert s'il permet de transférer le type et quantité de pétrole requis à partir des réservoirs pertinents vers les quais qui peuvent accueillir les gabarits des différents tankers.

$$x_{ij} = (t_0 \oplus u_i \oplus (\bigoplus_{i'} (x_{i'j'} \otimes p_{i'} \otimes V_{ij,i'j'}))) \otimes W_{ij} \quad \forall ij, i'j' \in \mathcal{OA} \quad (2.29)$$

L'équation (2.29) modélise la contrainte fondamentale augmentée, par l'inclusion de l'aspect de sélection d'alignements, et détermine le dateur pour une opération de transfert sur un alignement spécifique. La notation est la suivante :

$\mathcal{OA}$: ensemble de tous les appariements ij possibles tel que i représente une opération de transfert et j est un alignement candidat pour satisfaire la dite requête.

x_{ij} : dateur pour la requête i exécutable sur l'alignement j .

$V_{ij,i'j'}$: variable de décision binaire. $V_{ij,i'j'} \in \{\varepsilon, e\}$ détermine la précédence entre 2 alignements (j et j') en conflit satisfaisant 2 opérations de transfert (i et i'). Pour toute variable $V_{ij,i'j'}$, il existe une variable $V_{i'j',ij}$ complémentaire.

W_{ij} : variable de décision binaire. $W_{ij} \in \{\varepsilon, e\}$ permet de sélectionner l'alignement j pour la requête i . Cette variable détermine si l'alignement j est sélectionné ou pas pour traiter la requête i , et peut rendre effective ou négligeable la contrainte (2.29).

u_i : date d'arrivée du tanker pour la requête i .

t_0 , et $p_{i'}$, sont respectivement la date de début de l'horizon de planification, et la durée nominale de la requête i . Dans (2.29), le dateur pour la requête i avec l'alignement j est déterminé par la maximisation de 3 termes : t_0 , u_i , et la date d'achèvement pour les requêtes en conflit qui s'exécuteraient avant. En connaissant la topologie du réseau, pour toute paire d'alignements, les conflits peuvent être identifiés. Ces informations sont alors connues au préalable.

Dans (2.30), les valeurs possibles des variables de décision complémentaires sont restreintes, (pour ordonnancement et pour sélection d'alignement) à B (qui simule ε , i.e. $B \rightarrow -\infty$) et e . Dans (2.29), si $V_{ij,i'j'} = B$, l'ensemble du terme devient négligeable ce qui implique que l'opération i exécutée sur l'alignement j ne dépend pas de la date d'achèvement de l'opération de transfert i' exécutée sur l'alignement j' . Au contraire, si $V_{ij,i'j'} = e$, alors x_{ij} se positionne après $x_{i'j'} \otimes p_{i'}$ dans l'axe temporel. D'autre part, dans (2.29), si $W_{ij} = B$ alors la contrainte est négligeable pour l'appariement ij , i.e. l'alignement j n'est pas sélectionné pour la requête i . Si J_i est l'ensemble d'alignements candidats pour la requête i , pour tous les appariements possibles ij , il y aura un et un seul $j^{opt} \in J_i$, pour lequel $W_{ij^{opt}} = e$, et pour lequel $x_{ij} = x_{ij^{opt}} \neq \varepsilon$. L'équation (2.31) garantit cette unicité à l'aide d'un simple mapping $W_{ij} \rightarrow w_{ij}$, et naturellement $x_{ij^{opt}}$ correspond à $\bigoplus_j x_{ij}$ pour tous les appariements ij pour cette requête.

$$V_{ij,i'j'} \otimes V_{i'j',ij} = B; \quad V_{ij,i'j'} \oplus V_{i'j',ij} = e; \quad \bigotimes_j W_{ij} = B; \quad \bigoplus_j W_{ij} = e \quad (2.30)$$

$$\forall ij, i'j' \in \mathcal{OA}$$

$$w_{ij} = \begin{cases} 1 & \text{if } W_{ij} = e \\ e & \text{if } W_{ij} = \varepsilon \end{cases} \quad \bigotimes_j w_{ij} = 1 \quad \forall j \in J_i \wedge ij \in \mathcal{OA} \quad (2.31)$$

Les dates limites de service pour les requêtes sont définies comme dans le modèle non linéaire proposé par (2.9), i.e. avec un maximum de 36 heures de chargement. Finalement, (2.32) représente la fonction objectif qui évalue le *TCP* (*Total Cost due to Penalties*) pour toutes les requêtes

dans l'horizon de temps donné. Le terme *delay* est intuitivement le retard dans le service. Si des interruptions sont prévues par le terminal ou par le client, elles peuvent être intégrées comme dans (2.3).

$$Min\ TCP = \bigotimes_i^{nc} \left(\bigotimes_{n=1}^{delay_i} c_i \right) \quad \forall i \in \mathcal{I} \quad (2.32)$$

Pour illustrer les résultats, la Fig. 2.8a) présente une instance pour laquelle 3 opérations de transfert doivent être exécutées où (R_i, P_j) dénote une requête R_i pouvant être réalisée par l'alignement P_j . La topologie du réseau permet plusieurs alignements candidats. Notamment, le réseau a une capacité opérative (calculée dans les travaux de [1] par un problème de flux maximum) égale à 2 et, comme défini dans le chapitre 1, ceci se traduit par la possibilité d'effectuer au maximum 2 opérations de transfert simultanément.

Pour la requête R_1 , 2 réservoirs contiennent le produit requis et un seul quai est accessible au gabarit du tanker du client considéré. Ici, les 4 alignements possibles sont soulignés en bleu. Pour R_2 et R_3 , un seul réservoir et un seul quai peuvent habilitier le transfert des batch demandés ce qui génère 2 alignements candidats pour chacune de ces opérations.

Pour rappel, les seules vannes représentées en couleur sont les vannes d'ouverture habilitant l'alignement considéré; les vannes d'isolement sont omises dans la représentation. Supposant que tous les tankers arrivent en même temps et dans leurs fenêtres de temps autorisées à la date normalisée de $u_i = 95$ heures à partir du début de l'horizon de planification $t_0 = 0$, tous les clients peuvent alors entraîner des pénalités pour le terminal si les dates limites de service sont dépassées.

Avec le modèle proposé, l'ordonnancement optimum est obtenu tel que les alignements choisis sont ceux qui permettent de mieux intégrer les opérations en minimisant le *TCP*. Les Tableaux 2.4 et 2.5 présentent une simulation pour 2 instances différentes de configuration d'alignements, pour deux scénarios. L'exemple permet une validation intuitive des résultats.

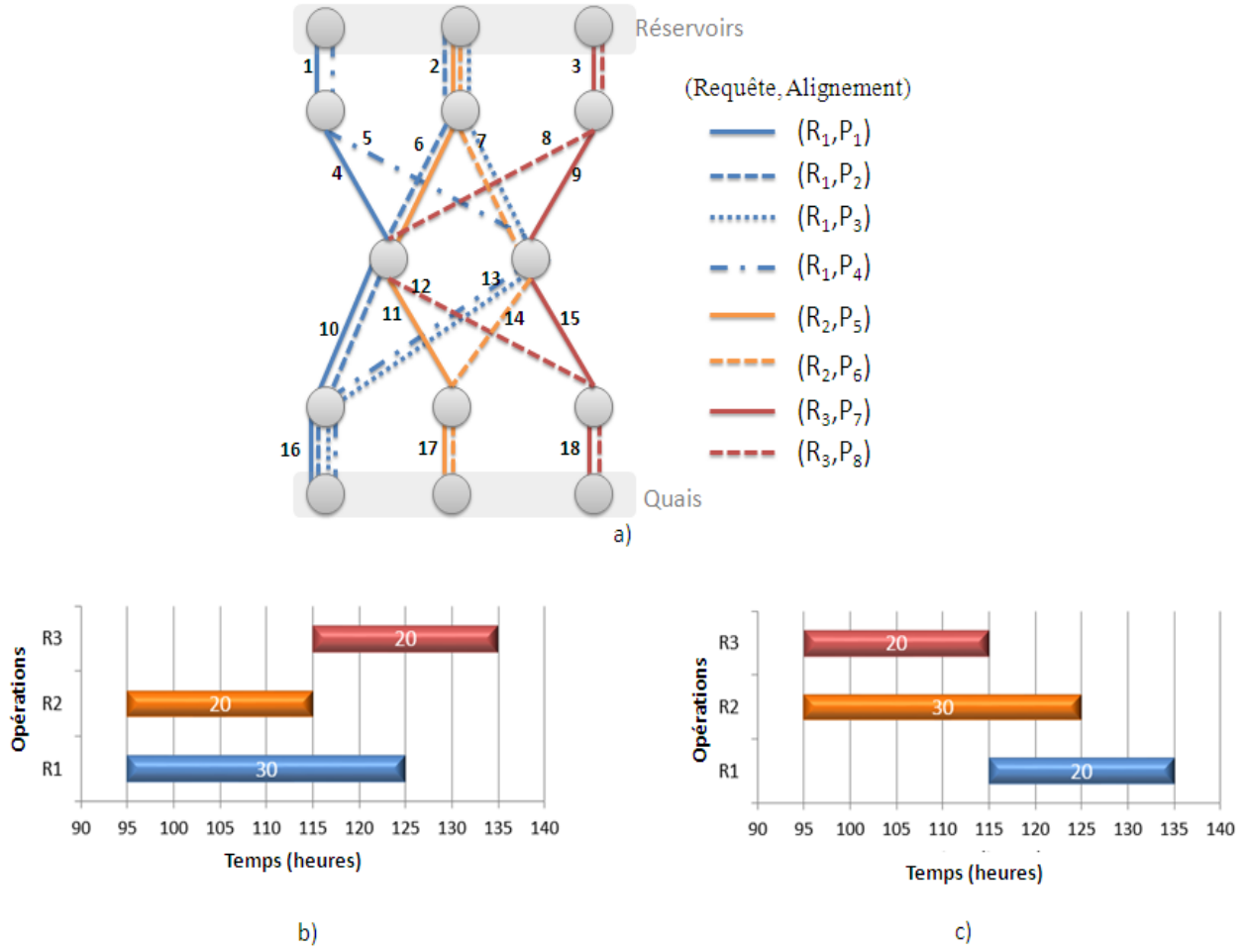


FIGURE 2.8: Ordonnements optima : a) Topologie de référence, b) ordonnancement optimum pour l'instance 1, c) Ordonnement optimum pour l'instance 2.

Requête	Input - Instance 1		Output - Instance 1		
	p_i (heures)	c_i (\$\backslash\$heures)	Alignement	Dateur	Retard (heures)
R_1	30	5000	P_1	95	0
R_2	20	4000	P_6	95	0
R_3	20	2000	P_7	115	4

TABLE 2.4: Instance 1 pour le problème de la Fig. 2.8.

Requête	Input - Instance 2		Output - Instance 2		
	p_i (heures)	c_i (\$\backslash\$heures)	Alignement	Dateur	Retard (heures)
R_1	20	2000	P_4	115	4
R_2	30	4000	P_5	95	0
R_3	20	5000	P_7	95	0

TABLE 2.5: Instance 2 pour le problème de la Fig. 2.8.

Dans la première instance, R_1 est identifiée de priorité forte et R_3 de priorité faible. Considérant qu'au plus 2 requêtes peuvent être traitées simultanément dans la topologie considérée, l'ordonnancement optimum apparaît dans la Fig. 2.8b). Dans cette figure, R_1 et R_2 sont exécutées simultanément avec les alignements disjoints P_1 , et P_6 (voir Tableau 2.4), respectivement. R_3 est traitée suite à l'achèvement de R_2 par l'alignement P_7 , ce qui permet une exécution en parallèle avec R_1 . Le retard généré par R_1 entraîne un TCP de \$8000.

Pour l'instance 2, l'opération R_3 est prioritaire en regard de R_1 . Dans les instance 2, une reconfiguration d'alignements est nécessaire en adaptation aux nouvelles informations opérationnelles telles que les pénalités potentielles pour chaque client. L'ordonnancement optimum est présenté sur la Fig. 2.8c), pour lequel le TCP est également de \$8000.

Les deux instances permettent d'identifier intuitivement les priorités entre les opérations et les possibles sélections d'alignements pour les satisfaire en minimisant les coûts. Pour des instances plus complexes, les interactions multiples qui surviennent du nombre de clients, d'alignements candidats, des dates d'arrivée des tankers et des pénalités, complexifient la prise de décision.

Les résultats du modèle proposé ont été publiés dans [67]. Ce modèle permet, à l'aide d'une manipulation mathématique simple, d'exploiter un modèle $(\max, +)$ concis et de résoudre, dans le cadre de l'optimisation de l'ordonnancement par une approche mono-critère, deux aspects différents de la prise de décision : la minimisation de pénalités et la sélection d'alignements. Ainsi, en cohérence avec les besoins réels opérationnels des terminaux maritimes, les impacts économiques de la sélection d'alignements sont prises en compte. La contrainte $(\max, +)$ (2.29) synthétise les conflits de partage de ressources, d'arrivée des clients, et de sélection d'alignements. Le modèle est non linéaire du fait du produit des dateurs avec des variables de décision associées aux conflits, et de celles associées à la sélection d'alignements. Une affectation de ces deux dernières rendent le modèle linéaire et permettent de déterminer les dateurs. Cette première phase d'étude motive l'intégration des aspects de modélisation non linéaire $(\max, +)$ avec des techniques heuristiques pour l'affectation des variables de décision tel que des sous-systèmes $(\max, +)$ soient obtenus à chaque itération et que la résolution matricielle $(\max, +)$ de ces systèmes soit exploitée. Ceci fait l'objet du chapitre suivant.

Conclusions Partielles

La modélisation mono-objectif répond au problème d'ordonnancement d'opérations de transfert en minimisant les pénalités subies par le terminal. Un modèle $(\max, +)$ non linéaire a représenté de manière concise et intuitive les contraintes du problème avec deux considérations : des alignements prédéfinis pour les opérations de transfert et des opérations de maintenance fixe. Ce point de départ a motivé l'exploration des modèles linéaires (basés sur le TPP ou le PCI) pour obtenir des ordonnancements optima à l'aide de produits matriciels $(\max, +)$ pour les opérations

de transfert. Ensuite, un modèle $(\max, +)$ non linéaire étendu a permis de gérer la sélection d'alignements sous la forme d'une contrainte pour la minimisation des pénalités. Cette dernière représentation motive des représentations plus complexes permettant de minimiser les retards des opérations de maintenance prévues, et d'explorer les compromis émergeant de l'optimisation de ces deux objectifs. À ce titre, un modèle basé sur des contraintes $(\max, +)$ et des techniques approximatives basées sur les algorithmes génétiques est proposé.

Chapitre 3

Modèle d'Optimisation Multi-objectif pour l'Ordonnancement d'Opérations

Préambule

Ce chapitre présente une extension des modèles $(\max, +)$ obtenus précédemment dans le cadre d'une exploitation à 2 objectifs : la minimisation des pénalités et la minimisation des retards des opérations de maintenance prédéfinies. L'ordonnancement d'opérations (de transfert et de maintenance), ainsi que la sélection d'alignements, reposent sur une proposition basée sur des algorithmes génétiques en intégration avec des modèles $(\max, +)$ linéaires. Les résultats visent à présenter à l'opérateur de supervision des solutions candidates répondant aux compromis optimaux par une approche de Pareto.

Sommaire

3.1 Cadre Général de la Problématique d'Optimisation Multi-Objectif	54
3.2 Optimisation Multi-Objectif et Algorithmes Génétiques - Notions Théoriques	56
3.3 Formulation du Problème d'Optimisation Multi-Objectif sous une Structure d'Algorithmes Génétiques	58
3.3.1 Structure Génomique	60
3.3.1.1 Vue d'Ensemble sur les Structures Génomiques	60
3.3.1.2 Structure Génomique Proposée	62
3.3.2 Opérateur de Mutation	67
3.3.3 Opérateur de Croisement	69
3.4 Stratégie Évolutionnaire et Intégration des Modèles Algébriques $(\max, +)$	72
3.5 Résultats	79

3.1 Cadre Général de la Problématique d'Optimisation Multi-Objectif

Ce chapitre porte sur la modélisation et la résolution d'un problème d'optimisation multi-objectif. La résolution du problème est abordée par une approche d'algorithmes génétiques exploitant la modélisation algébrique dans le cadre $(\max, +)$ développée dans le chapitre précédent. Dans cet esprit, les défis relèvent de la définition de la structure de codification de l'information pour chaque individu (i.e. génotype), et des opérations génétiques (i.e. sélection, mutation et croisement) sur cette structure.

Une implémentation d'une approche évolutionnaire basée sur les algorithmes génétiques est proposée, en intégrant l'exploitation des principes mathématiques des modèles $(\max, +)$ linéaires. Le résultat d'implémentation du modèle est un ensemble de solutions optimales, qui représentent un compromis entre deux objectifs visant à assister l'opérateur de supervision dans sa décision en fonction des résultats fournis et de son expertise. Ainsi, c'est l'opérateur qui finalement décide et applique une des alternatives d'affectation de ressources et de datation.

Les objectifs d'optimisation considérés sont les pénalités encourues par le terminal dû au service tardif et les retards des activités de maintenance. Ces deux objectifs sont antagonistes (en conflit, car l'un améliore en détriment de l'autre) ; par conséquent, des différents compromis entre eux peuvent être considérés lors de la gestion l'ordonnancement d'opérations et la sélection d'alignements. La pénalité globale à minimiser pour l'ordonnancement d'un ensemble d'opérations de transfert de pétrole a été représentée par l'indice TCP (*Total Cost due to Penalties*) introduit dans le chapitre précédent. D'autre part, il est admis qu'un programme de maintenance établi par une équipe spécialisée doit être respecté, dans les possibilités de fonctionnement du système, par l'opérateur de supervision lors de l'exécution des opérations de transfert.

Dans ce qui suit, l'indice TME (en anglais : *Total Maintenance Error*) représente la somme des retards des activités de maintenance. Les dates prévues d'intervention sont données au préalable, mais la date effective de maintenance dépend des activités de transfert. Ainsi, le lancement d'une activité de maintenance pourrait entraîner des pénalités du fait d'opérations de transfert en conflit. Par contre, le retard de l'opération de maintenance peut avoir des conséquences sur la fiabilité du dispositif en question et/ou sur les coûts d'intervention, qui doivent aussi être minimisés. En outre, le retard d'une ou de plusieurs activités de maintenance peut entraîner de nouveaux conflits avec d'autres opérations de transfert.

Les approches formelles sont d'usage difficile dans la résolution de problèmes d'optimisation multi-objectif du fait que l'unicité de la solution ne peut être assurée, que l'espace de recherche augmente dû aux plusieurs objectifs à gérer, et qu'il doit être parcouru en gérant des compromis de qualité entre les différents objectifs. Pour pallier ces difficultés, des approches approximatives

issues de l'intelligence artificielle parcourent l'espace de recherche dans un cadre distribué de collaboration entre plusieurs entités, chacune recherchant une solution. Ces sont des algorithmes basés sur des populations. Sans exhaustivité sur le sujet, il peut être signalé que d'autres approches se centrent sur l'amélioration d'une seule solution comme par exemple des approches de recherche locale.

L'espace de recherche est alors formé par l'ensemble des affectations possibles pour les variables de décision (pour l'ordonnancement d'opérations et pour la sélection d'alignements) qui conduisent à des différentes valeurs de TCP et de TME . En pratique, un $TME = 0$ (i.e. maintenance fixe) assure le déroulement des opérations de maintenance comme prévues mais pénalise la valeur du TCP . Si seul le TCP est minimisé, la planification d'opérations de maintenance peut être fortement impactée. Ainsi, il s'agit de déterminer des compromis entre les valeurs des deux objectifs, i.e. un ensemble de solutions optimales qui assistera à l'opérateur de supervision dans la prise de décision. Chacune de ces solutions est représentée par un élément de type (TCP, TME) issu d'une sélection d'alignements pour les opérations de transfert et d'un ordonnancement de l'ensemble d'opérations dans le système (de transfert et de maintenance).

Le choix de la méthode de résolution est motivée par rapport à un état de l'art faisant bien cas de l'intérêt des techniques évolutionnaires. Ces techniques émulant l'évolution naturelle des espèces, permettent de trouver un ensemble de solutions optimales en parcourant l'espace de recherche de manière distribuée, sur les bases de la collaboration entre différentes entités qui recherchent une solution. Les algorithmes génétiques ont été sélectionnés comme technique de résolution sachant que :

- 1) aucune approche n'est reconnue comme la meilleure pour le problème traité ;
- 2) au niveau scientifique un effort très important est encore dédié à l'amélioration de ces techniques ; et
- 3) aucun aspect supplémentaire n'impose la sélection d'une technique spécifique.

Cette approche de résolution des problèmes d'optimisation multi-objectif est bien connue dans la communauté scientifique de l'intelligence artificielle, et représente l'un des axes majeurs de recherche au laboratoire CEMISID, de l'Universidad de Los Andes (ULA) au Venezuela, ainsi qu'au laboratoire de Raisonnement et Analyse dans les Systèmes Complexes au sein de la TRT (en anglais : *Thales Research and Technology*), chez Thales Group.

3.2 Optimisation Multi-Objectif et Algorithmes Génétiques - Notions Théoriques

Les travaux dans [53] sont un rappel aux notions générales de l'optimisation multi-objectif. De façon générale, pour ces types de problème, une solution peut être dénotée comme un vecteur de décision $x_i = (x_{1i}, x_{2i}, \dots, x_{ni})$ dans l'espace de décision $\mathbf{X}$. Pour chaque vecteur de décision, une fonction $\mathbf{X} \rightarrow \mathbf{Y}$ permet de déterminer sa représentation dans l'espace de critères par un vecteur objectif $y_i = f(x_i) = (f_1(x_{1i}), f_2(x_{2i}), \dots, f_n(x_{ni}))$. Ainsi, formellement, un vecteur objectif y_1 domine y_2 , i.e. $y_1 \succ y_2$, si aucun composant de y_1 n'est inférieur au composant correspondant dans y_2 , et qu'au moins un composant est supérieur. Cette notion est nommée *dominance de Pareto*. Ainsi, une solution x_1 est meilleure que x_2 , i.e. x_1 domine x_2 ($x_1 \succ x_2$) si $f(x_1)$ domine $f(x_2)$.

Le problème de base repose sur le fait qu'il peut exister plusieurs vecteurs objectif optimaux représentant différents compromis de qualité entre les objectifs. L'ensemble des solutions optimales dans l'espace de décision $\mathbf{X}^* \subseteq \mathbf{X}$ est généralement appelé l'ensemble de Pareto, et son image dans l'espace d'objectifs est appelée la frontière de Pareto, i.e. $\mathbf{Y}^* = f(\mathbf{X})^* \subseteq \mathbf{Y}$.

La génération de la frontière de Pareto peut être coûteuse en temps d'exécution, raison pour laquelle les stratégies stochastiques, tel que les algorithmes évolutionnaires, permettent de trouver des solutions approximées (normalement dans la frontière de Pareto) dans des délais raisonnables.

Les algorithmes génétiques font partie des techniques évolutionnaires les plus développées et recherchées pour résoudre des problèmes d'optimisation complexes s'inspirant sur l'émulation de l'évolution naturelle des espèces. L'idée générale repose sur la génération d'une population d'individus, où chacun représente une solution au problème. Lors de l'exécution de l'algorithme, cette population évolue, ce qui représente le parcours de l'espace de recherche avec le but de trouver les solutions optimales. Ainsi, cet espace de recherche est parcouru de manière distribuée. Comme dans tout algorithme évolutionnaire, ce parcours est contrôlé par des opérateurs évolutionnaires qui visent à intensifier ou diversifier la recherche. L'intensification permet d'accélérer la convergence vers des points optimaux en privilégiant les meilleures solutions trouvées à chaque itération, et la diversification permet d'étendre l'exploration de l'espace de recherche pour éviter de tomber sur des optima locaux. Les deux opérateurs sont nécessaires, et leur correcte intégration permet d'obtenir les points optimaux visés dans des délais raisonnables. Une intensification trop forte réduit le temps d'exécution mais peut générer comme résultat des optima locaux, et une trop forte exploration évite des optima locaux mais pénalise fortement le temps d'exécution. Ainsi, un 'bon' paramétrage de ces opérateurs est crucial pour l'obtention des bonnes solutions dans des délais acceptables. Ceci fait actuellement objet de nombreux travaux de recherche. Souvent, d'importants tests de différentes combinaisons de paramètres sont nécessaires pour obtenir une configuration adaptée à l'instance traitée. Dans les algorithmes génétiques, l'opérateur

d'intensification est lié à la notion de fitness de chaque individu, i.e. son aptitude à transmettre ses gènes aux générations suivantes du fait de la qualité de solution représentée, et l'opérateur d'exploration vise à modifier, de manière aléatoire, certaines solutions pour les déplacer dans l'espace de recherche. La phase de *sélection* permet, comme dans l'évolution naturelle des espèces, d'assurer que les solutions avec plus de chance de survivre soient celles de fitness plus élevé. Ainsi, à partir d'une population initiale, une fonction de fitness doit être définie telle que les individus soient classés, et puis sélectionnés, pour construire la génération suivante. Des individus de fitness élevée ont plus de chance de se reproduire. Ceci est modélisé par l'opérateur de *croisement*, qui génère des individus nommés 'descendants' pour la prochaine génération à partir des individus nommés 'parents'. La charge génétique des individus descendants est ainsi formée par des gènes des deux parents. Finalement, l'opérateur de *mutation* permet d'appliquer des modifications aléatoires à certaines parties de la solution, en vue de trouver des solutions pas encore explorées.

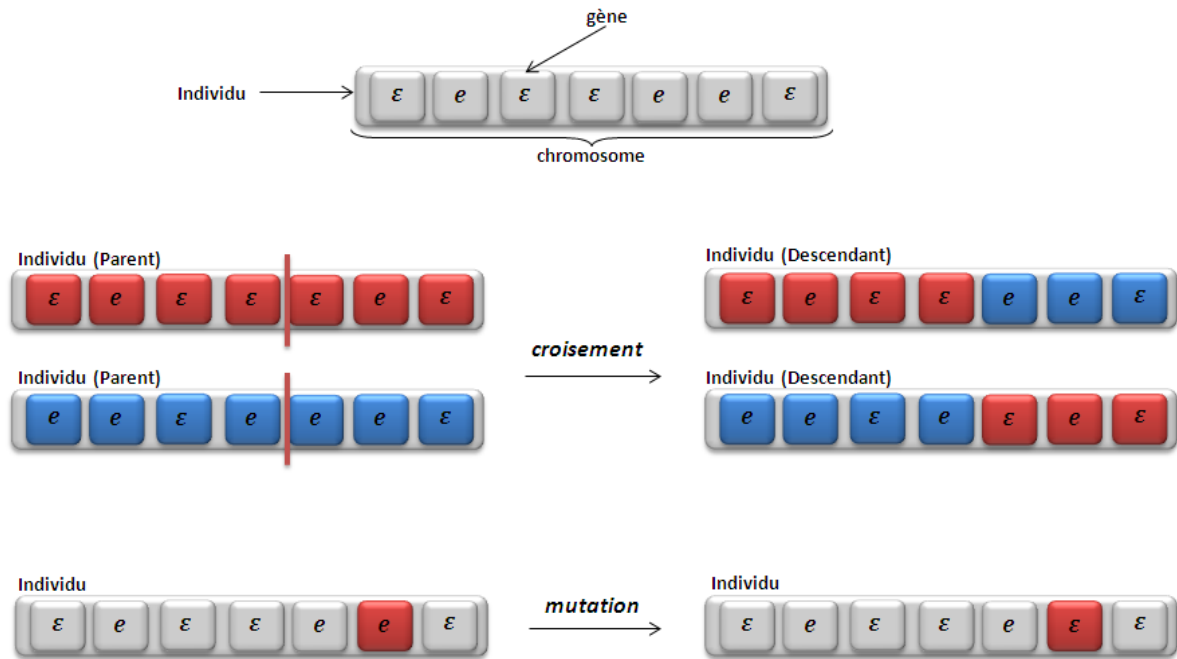


FIGURE 3.1: Opérateurs génétiques.

Les notions d'opérateurs et de structures génétiques sont illustrées dans la Fig. 3.1. La structure de données dénotant le matériel génétique de chaque individu correspond au 'génotype'. Le 'phénotype', par contre, est l'interprétation ou mapping de ces données vers l'information d'intérêt de la problématique. Finalement, un génome est défini comme une instance spécifique du génotype.

Dans la Fig. 3.1, un codage binaire du génotype est présenté comme exemple pour illustrer le fonctionnement des opérateurs génétiques. En rappelant que, pour le problème d'ordonnancement considéré, dans le contexte $(\max, +)$ les valeurs des variables de décision sont : éléments

neutre et nul¹ (i.e. $e = 0$ et $\varepsilon = -\infty$), les gènes dans le chromosome ne pourront prendre alors que ces valeurs. De manière générale, un individu sera caractérisé par un vecteur de variables de décision (d'affectation d'alignements et de précédence entre opérations en conflit) que détermine par conséquent un vecteur de dateurs x . Ce vecteur de dateurs est à la fois un ordonnancement caractérisé par des valeurs de TCP et TME .

Ainsi, l'opérateur de mutation, par exemple, peut modifier la valeur d'un gène de e à ε ou vice versa. L'opérateur de croisement, par contre, génère des individus comportant une combinaison des charges génétiques des parents.

3.3 Formulation du Problème d'Optimisation Multi-Objectif sous une Structure d'Algorithmes Génétiques

Il est nécessaire au préalable que la structure d'une solution soit encodée sous la forme d'un chromosome, i.e. le génotype doit être défini. Les principales contraintes du problème d'optimisation ont déjà été déterminées de manière concise par application de l'algèbre $(\max, +)$; voir (2.29-2.32). Considérant que les activités de maintenance sont intégrées au modèle dans l'approche multi-objectif, l'équation (2.29) est alors modifiée pour obtenir (3.1) et, par conséquent, (3.2) est rajoutée au modèle pour représenter les dateurs des opérations de maintenance.

$$x_{i,j} = (t_0 \oplus u_i \oplus (\bigoplus_{i',j'} (x_{i',j'} \otimes p_{i'} \otimes V_{i,j;i',j'})) \oplus (\bigoplus_k (x_{m_k} \otimes tm_k \otimes V_{i,j;k}))) \otimes W_{i,j} \quad \forall i, j \quad i', j' \in \mathcal{OA} \wedge \forall k \in \mathcal{M} \quad (3.1)$$

$$x_{m_k} = xf_k \oplus (\bigoplus_{i,j} (x_{i,j} \otimes p_i \otimes V_{k;i,j})) \quad \forall i, j \in \mathcal{OA} \wedge \forall k \in \mathcal{M} \quad (3.2)$$

La plupart des éléments dans les équations (3.1) et (3.2) ont déjà été définis dans la section 2.2.4 mais ils sont rappelés pour faciliter la compréhension de la suite de la section.

$x_{i,j}$: dateur de l'opération de transfert i si elle est exécutée sur l'alignement j .

$x_{i',j'}$: dateur de l'opération de transfert i' si elle est exécutée sur l'alignement j' .

x_{m_k} : dateur de l'activité de maintenance sur la vanne k .

xf_k : date prévue de l'activité de maintenance sur la vanne k , fixée dans le programme de maintenance obtenu au préalable.

1. déterminant la précédence entre les opérations en conflit ou l'affectation des alignements aux opérations de transfert, voir Section 2.2.4.

$\mathcal{OA}$: ensemble de paires requête-alignement i, j .

$\mathcal{M}$: ensemble des vannes avec une activité de maintenance prévue.

$V_{i,j;i',j'}$: variable de décision binaire (et complémentaire à la variable $V_{i',j';i,j}$ qui survient lors de la formulation de la contrainte analogue pour déterminer $x_{i',j'}$) déterminant la précédence entre l'opération de transfert i (si exécutée sur un certain alignement j) et l'opération en conflit i' (si exécutée sur un certain alignement j'). Les opérations i et i' sont en conflit à cause du partage de ressources et, par conséquent, ne peuvent pas être exécutées simultanément. $V_{i,j;i',j'}$ et $V_{i',j';i,j}$ prennent les valeurs $\varepsilon = -\infty$ ou $e = 0$, sachant que $V_{i,j;i',j'} \neq V_{i',j';i,j}$. Pour rappel, $V_{i,j;i',j'} = \varepsilon$ si l'opération de transfert i sur l'alignement j s'exécute avant l'opération de transfert i' sur l'alignement j' ; dans le cas contraire $V_{i,j;i',j'} = e$.

$V_{i,j;k}$: de manière analogue, détermine la précédence entre l'opération de transfert i (si exécutée sur un certain alignement j) et l'opération de maintenance sur la vanne k . De manière analogue, $V_{i,j;k}$ et $V_{k;i,j} \in \{\varepsilon, e\}$, sachant que $V_{i,j;k} \neq V_{k;i,j}$.

u_i : date d'arrivée du tanker pour l'opération de transfert i .

t_0 : date de début de l'horizon d'ordonnancement.

tm_k et $p_{i'}$: durées d'une opération de maintenance sur la vanne k et d'une opération de transfert i' , respectivement.

$W_{i,j}$: variable binaire de décision déterminant l'affectation (ou pas) de l'alignement j à l'opération de transfert i . La valeur de cette variable rend effective ou négligeable toute la contrainte (3.1) puisque les conflits modélisés pour obtenir le dateur (et par conséquent, l'ordonnancement) ne sont effectifs que comme conséquence du choix de cet alignement. $W_{i,j}$ prend la valeur de $e = 0$ si et seulement si l'alignement est choisi, et $\varepsilon = -\infty$ sinon.

Comme dans la section 2.2.4, naturellement, $x_{i,j,opt}$ correspond à $\bigoplus_j x_{i,j}$ pour tous les appariements i, j pour chaque requête i .

Considérant comme point de départ la modélisation à l'aide de l'algèbre $(\max, +)$, la formulation du problème d'optimisation de l'ordonnancement et sélection d'alignements sera abordée dans la section suivante sous forme d'une approche d'algorithme génétique, en cohérence avec le workflow de la Fig. 3.2. Tout d'abord, la structure d'un individu (solution) doit être définie pour le problème étudié. Ensuite, le comportement des opérateurs de variation (mutation et croisement) sont définis pour la structure proposée pour chaque individu. Ensuite, une stratégie doit être définie pour comparer les individus vis-à-vis d'un classement qui définira l'évolution de la population. Cela est issu de l'évaluation des individus par rapport à des indicateurs liés aux objectifs d'optimisation, tenant compte qu'il s'agit d'un problème d'optimisation multi-objectif.

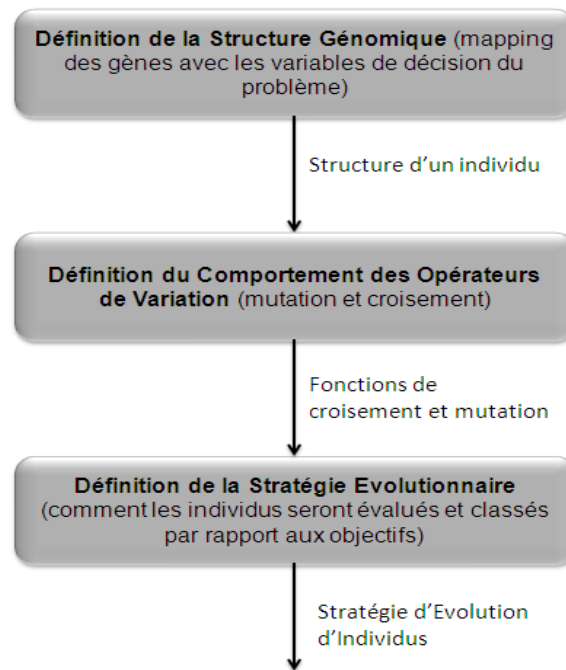


FIGURE 3.2: Workflow général pour la formulation du problème multi-objectif par une approche d'algorithmes génétiques.

3.3.1 Structure Génomique

Dans la formulation du problème sous forme d'approche génétique, le premier aspect à définir est la structure génomique, i.e. comment représenter les solutions sous une abstraction de chromosome, telles qu'elles puissent être manipulées dans le processus d'évolution.

Tout d'abord, diverses structures génomiques sont décrites et ensuite une structure génomique est proposée pour le problème adressé.

3.3.1.1 Vue d'Ensemble sur les Structures Génomiques

Dans les travaux de [68], différents types de codages génétiques sont présentés : le codage binaire (BGA : *Binary Genetic Algorithms* en anglais) où les gènes prennent des valeurs de 0 ou 1 et forment, dans un chromosome, une valeur associée à la solution (e.g. une distance ou un coût) ; le codage réel (RGA : *Real Genetic Algorithms* en anglais) où la représentation de chaque gène est en point flottant, ce qui permet d'accélérer le temps d'exécution ; et le codage structuré (SGA : *Structured Genetic Algorithms* en anglais) où les chromosomes sont des structures hiérarchiques avec codage binaire, et les gènes d'un haut niveau contrôlent les gènes d'un niveau inférieur.

Cette dernière structure génomique est d'un intérêt particulier puisque, même si elle peut être plus complexe à implémenter, elle donne un degré de flexibilité supplémentaire à l'affectation des valeurs des variables de décision, notamment pour traiter des inter-dépendances.

Dans cette approche structurée, un changement dans un gène de haut niveau entraîne plusieurs changements dans les gènes de niveau inférieur. Dans les algorithmes génétiques classiques, ce comportement ne pourrait être obtenu qu'avec une séquence de nombreux changements aléatoires. Finalement, dans [68], les RSGA ou algorithmes génétiques avec codage réel et structuré (en anglais : *Real-Coded Structured Genetic Algorithms*) sont décrits et appliqués à la conception des filtres numériques et des contrôleurs. Les RSGA combinent la représentation structurée et le codage réel, pour formuler une représentation génomique dans laquelle les gènes de haut niveau, aussi appelés «gènes de contrôle» modifient directement un des gènes de niveau inférieur, aussi appelés «gènes de paramétrage» ainsi chaque gène du chromosome prend une valeur réelle. La Fig. 3.3 schématise les structures SGA et RSGA. Les RSGA améliorent la performance de l'algorithme à partir de la représentation de chaque génome, tel que si dans les SGA les gènes de plus haute hiérarchie modifiaient plusieurs gènes de niveau inférieur pour coder une valeur réelle, dans les RSGA chaque gène de contrôle modifie la valeur de son gène de paramétrage affectant directement une valeur réelle.

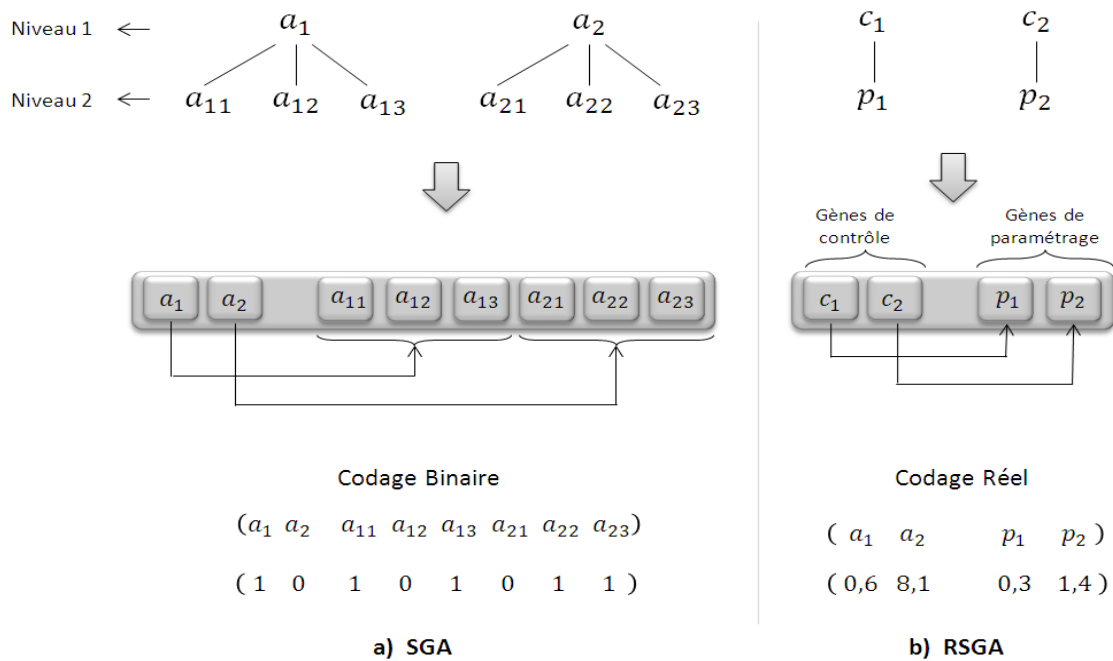


FIGURE 3.3: Schémas de représentation de l'information dans les algorithmes génétiques structurés.

La Fig. 3.3a) présente, selon [68], une structure hiérarchique de génotype où, par exemple, le gène a_1 contrôle les valeurs des gènes a_{11} , a_{12} et a_{13} . Puisqu'il s'agit d'une codification binaire, l'ensemble des gènes ne prennent que les valeurs 0 ou 1. Dans cette structure, chaque sous-ensemble de gènes de paramétrage représente normalement un nombre, e.g. l'affectation 1 0 1 représenterait le numéro 5 en base décimale. Ainsi, le codage binaire implique l'affectation d'une suite de valeurs aux gènes de paramétrage, en vue de l'obtention d'une valeur d'intérêt dans le problème d'optimisation. Par contre, le codage réel considère que chaque gène caractérise déjà

la valeur d'intérêt (Fig. 3.3b)). Les RSGA réduisent le temps de calcul, ainsi que la longueur des chaînes utilisées pour décrire chaque chromosome. En outre, un gène de paramétrage prend une certaine valeur dépendant de la valeur de son gène de contrôle, cette valeur peut correspondre à son activation, désactivation ou à une variation linéaire de sa valeur. Ainsi, lorsqu'un gène de contrôle change dans une génération spécifique, il peut entraîner un changement dans le gène de paramétrage correspondant. Son activation implique qu'il conserve la même valeur pour la génération suivante, la variation linéaire implique qu'il change de valeur par rapport à une fonction linéaire définie au préalable, et sa désactivation implique que sa valeur ne sera plus prise en compte pour les générations suivantes, n'ayant alors plus d'influence sur le processus d'évolution.

3.3.1.2 Structure Génomique Proposée

La structure génomique vise à définir les gènes nécessaires pour représenter un individu ou solution du problème d'optimisation. En outre, les interactions entre ces gènes sont définies par rapport à une structure hiérarchique.

L'idée fondamentale est de distinguer des variables de décision qui jouent un rôle dans l'affectation d'alignements aux opérations de transfert (interprétées comme des gènes de structure, i.e. $W_{i,j}$ dans (3.1)), et des variables de décision qui jouent un rôle dans l'ordonnancement d'opérations (interprétées comme des gènes de paramétrage, i.e. variables $V_{i,j;k}$, $V_{k;i,j}$, $V_{i,j;i',j'}$ et $V_{i',j';i,j}$ dans (3.1) et (3.2)). La structure mathématique de l'équation (3.1) met en évidence que la variable $W_{i,j}$ valide ou non la contrainte de partage de ressources dans le problème d'optimisation pour une paire i, j donnée. Ainsi, dans (3.1), si $W_{i,j} = \varepsilon$, alors l'ensemble de la contrainte tend vers $-\infty$ (i.e. $x_{ij} \rightarrow -\infty$), car l'alignement j n'est pas choisi pour l'exécution de la requête i . Comme la contrainte (3.1) modélise les conflits de partage de ressources qui surviennent lors de la sélection de l'alignement j pour l'opération i , ces conflits ne sont plus effectifs si l'alignement n'est pas choisi (et les variables de décision qui les modélisent ne sont plus pertinentes dans le modèle). Par contre, si l'alignement est choisi, la contrainte comportera bien les termes concernant les conflits de partage de ressources et les valeurs de ces variables seront considérées. Il faut remarquer ainsi que le caractère structurel des variables de sélection d'alignements a un impact sur les variables de décision pour l'ordonnancement. Plus spécifiquement, lorsqu'un alignement est sélectionné, il peut entraîner des conflits de partage de ressources avec d'autres alignements dans le réseau; par conséquent, l'ordonnancement d'opérations est déterminé en fonction de cette sélection.

L'exemple de la Fig. 3.4 vise à illustrer la structure hiérarchique proposée pour la définition du génotype du problème. Dans cette figure, deux requêtes doivent être exécutées, deux alignements sont possibles pour chacune d'entre elles et une maintenance est prévue sur l'une des ressources.

Supposant que tous les conflits potentiels deviennent effectifs, i.e. les deux tankers arrivent au début de l'horizon de temps et l'opération de maintenance sur la vanne 13 doit être effectuée à la même date, alors les contraintes fondamentales du système sont représentées de (3.3) à (3.7).

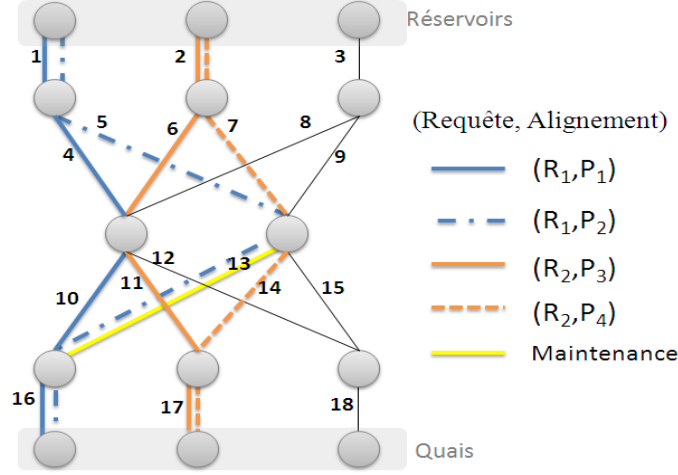


FIGURE 3.4: Instance simple pour la définition de la structure génomique de la problématique

$$x_{1,1} = (u_1 \oplus (x_{2,3} \otimes p_2 \otimes V_{1,1;2,3}) \oplus (x_{m_{13}} \otimes tm_{13} \otimes V_{1,1;13})) \otimes W_{1,1} \quad (3.3)$$

$$x_{1,2} = (u_1 \oplus (x_{2,4} \otimes p_2 \otimes V_{1,2;2,4}) \oplus (x_{m_{13}} \otimes tm_{13} \otimes V_{1,2;13})) \otimes W_{1,2} \quad (3.4)$$

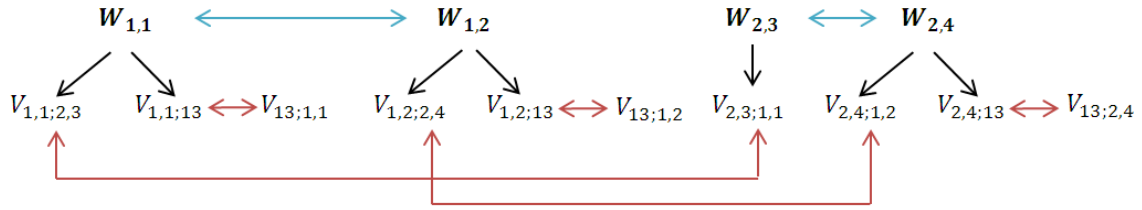
$$x_{2,3} = (u_2 \oplus (x_{1,1} \otimes p_1 \otimes V_{2,3;1,1})) \otimes W_{2,3} \quad (3.5)$$

$$x_{2,4} = (u_2 \oplus (x_{1,2} \otimes p_1 \otimes V_{2,4;1,2}) \oplus (x_{m_{13}} \otimes tm_{13} \otimes V_{2,4;13})) \otimes W_{2,4} \quad (3.6)$$

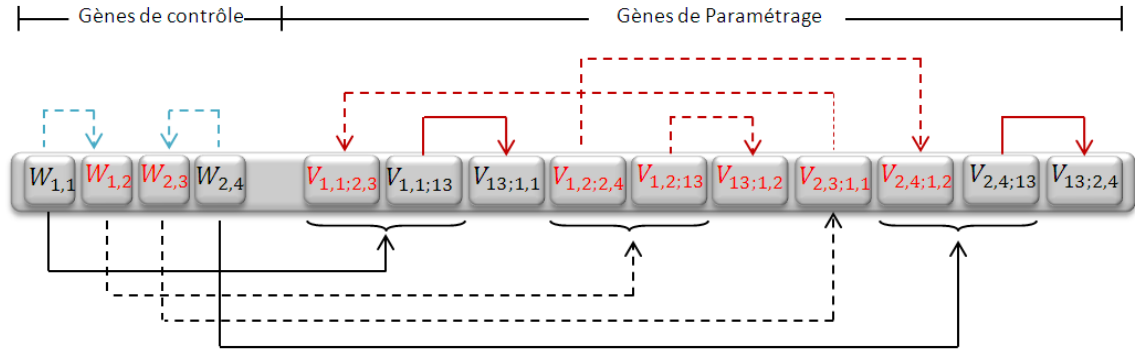
$$x_{m_{13}} = xf_{13} \oplus (x_{1,1} \otimes p_1 \otimes V_{13;1,1}) \oplus (x_{1,2} \otimes p_1 \otimes V_{13;1,2}) \oplus (x_{2,4} \otimes p_2 \otimes V_{13;2,4}) \quad (3.7)$$

Les variables de décision dans le système d'équations (3.3-3.7), pour l'exemple de la Fig. 3.4, sont transposées sur la structure génomique proposée dans la Fig. 3.5. La structure génomique est hiérarchique avec une caractérisation des variables de sélection d'alignements (i.e. $W_{i,j}$) comme des gènes de contrôle, et des variables de décision (i.e. $V_{i,j;i'}$, $V_{ij;k}$ ou ses complémentaires) associées au partage de ressources comme des gènes de paramétrage.

a) Structure Hiérarchique



b) Génome



c) Codage Binaire

(e ε ε e $\emptyset$ ε e $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$ $\emptyset$ ε e)

d) Solution Qualitative Encodée

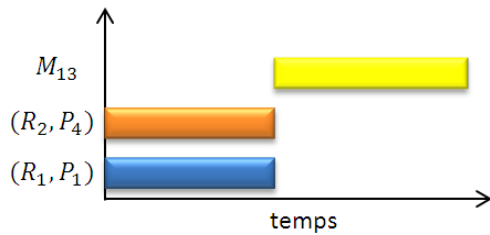


FIGURE 3.5: Structure génomique proposée.

Dans le cadre SGA (Fig. 3.3a)), le changement de valeur d'une variable de haut niveau peut générer plusieurs changements de valeurs des variables de niveau inférieur (flèches en noir). La Fig. 3.5a) présente la structure hiérarchique pour l'exemple de la Fig. 3.4 décrit par les équations (3.3-3.7); les flèches reliant les variables indiquent le sens du contrôle. Ainsi, les gènes de plus haut niveau contrôlent certains gènes de paramétrage. Néanmoins, à différence des approches SGA et RSGA, un contrôle est aussi établi entre certains gènes de paramétrage (flèches en rouge) et entre certains gènes de contrôle (flèches en bleu) concernant les variables complémentaires. Notamment, les approches consultées ne considèrent pas ce type de contrôle dans un même niveau; néanmoins, pour le problème adressé, cette manipulation permet d'exploiter les

informations connues au préalable, i.e. une affectation de variable conduit à l'affectation de sa variable complémentaire, ce qui augmente la convergence vers des solutions faisables.

Dans la Fig. 3.5, les flèches bidirectionnelles indiquent l'existence d'un contrôle dans les deux sens. La notion de base pour cette structure hiérarchique stipule que les valeurs des variables de sélection d'alignements déterminent la pertinence ou pas des variables de décision pour les opérations en conflit. Il est aussi convenu que chaque valeur de variable de décision de conflit détermine automatiquement la valeur de sa variable complémentaire (i.e. entre deux opérations «A» et «B» en conflit : si «A» est prioritaire, il est établi que «B» n'est pas prioritaire). Ces variables complémentaires seront nommées 'gènes complémentaires'. En outre, un gène de contrôle activé, peut désactiver les autres gènes de contrôle pour la même requête, ce qui traduit le fait que pour une requête, si un alignement est sélectionné, les autres sont rejetés.

Ainsi, dans la Fig. 3.5a), si $W_{1,1} = e$ alors les conflits de partage de ressources exprimés dans (3.3) deviennent effectifs pour la détermination de l'ordonnancement, i.e. la contrainte devient pertinente dans la résolution du problème, et implique que les variables $V_{1,1;2,3}$ et $V_{1,1;13}$ doivent être affectées puisque l'alignement 1 a été sélectionné. De même, si $V_{1,1;2,3} = e$ (la requête 2 sur l'alignement 3 s'exécute avant la requête 1 sur l'alignement 1), automatiquement la variable complémentaire $V_{2,3;1,1}$ acquiert la valeur ε (qui indique dans (3.5) que la requête 1 sur l'alignement 1 s'exécute après la requête 2 sur l'alignement 3). Par contre, si au niveau supérieur $W_{1,1} = \varepsilon$, cela implique que la contrainte (3.3) n'est plus pertinente dans la résolution du problème car comme l'alignement 1 n'a pas été sélectionné pour la requête 1, alors les conflits potentiels sous-jacents ne surviennent pas. Dans ce cas, toutes les variables de décision associées sont négligeables pour le traitement du problème.

La propagation d'affectation des valeurs du niveau supérieur génétique vers le niveau inférieur a été décrite, ainsi que la propagation dans le niveau inférieur grâce aux variables de décision complémentaires. Néanmoins, cette propagation au même niveau survient aussi pour les variables de sélection d'alignements. Pour l'exemple présenté, avec 2 alignements possibles par requête, si $W_{1,1} = \varepsilon$ alors $W_{1,2}$ prend la valeur e automatiquement, ou vice versa.

La Fig. 3.5b) représente le génome pour la structure considérée avec un contrôle génétique effectué au niveau hiérarchique. Le processus se déroule comme suit :

1. Pour chaque opération de transfert, un des gènes de contrôle est affecté de la valeur e représentant la sélection d'un alignement donné. Ainsi, aléatoirement pour l'opération de transfert 1 il est obtenu $W_{1,1} = e$, ce qui entraîne automatiquement la désactivation du gène complémentaire : $W_{1,2} = \varepsilon$ (i.e. si l'alignement 1 est sélectionné, l'alignement 2 est rejeté). De même pour l'opération de transfert 2, aléatoirement $W_{2,4} = e$ entraînant $W_{2,3} = \varepsilon$. Dans la Fig. 3.5b), les alignements rejetés sont illustrés par des gènes en rouge et sont dénotés comme 'désactivés'. La désactivation d'un gène de contrôle (par un autre

gène) est représentée par une flèche segmentée. Le facteur aléatoire de cette étape repose sur l'activation d'un seul gène de contrôle pour chaque requête i parmi J_i alignements possibles.

2. Les gènes de contrôle désactivés produisent une réaction en chaîne de désactivation des gènes de paramétrage (représentée par des lignes segmentées), et les gènes de paramétrage désactivés ont la faculté de désactiver simultanément ses gènes complémentaires. Tous les gènes désactivés sont illustrés en rouge dans la Fig. 3.5b).
3. Les gènes de contrôle qui restent actifs (i.e. de valeur égale à e) habilitent ses gènes de paramétrage à des valeurs aléatoires dans l'ensemble $\{e, \varepsilon\}$. Ces gènes actifs déterminent l'ordonnancement d'opérations. Dans l'exemple considéré, $V_{1,1;13}$ acquiert la valeur ε aléatoirement, affectant son gène complémentaire, i.e. $V_{13;1,1} = e$. De même, si $V_{2,4;13} = \varepsilon$ alors $V_{13;2,4} = e$.
4. Le codage binaire pour l'individu est exprimé dans la Fig. 3.5c). Gardant la notation de [68], les gènes de paramétrage rejetés de la solution sont représentés avec une valeur égale à $\emptyset$ pour indiquer une désactivation. Ceci indique simplement que la valeur du gène n'a aucun impact sur la solution.

La Fig. 3.5d) présente la solution qualitative caractérisée par le chromosome. Les gènes de structure correspondent au vecteur : ' $e \ \varepsilon \ \varepsilon \ e$ ', qui représente la sélection des alignements 1 et 4 pour les requêtes 1 et 2, respectivement. Cette sélection d'alignements est présentée dans la Fig. 3.5d) dans l'axe des ordonnées (i.e. appariement requête-alignement). Du fait de l'activation de gènes de structure, les gènes de paramétrage activés représentent, par rapport au génotype, l'affectation aléatoire des variables suivante : $V_{1,1;13} = \varepsilon$, $V_{13;1,1} = e$, $V_{2,4;13} = \varepsilon$, $V_{13;2,4} = e$. Cette affectation implique premièrement que par la sélection d'alignements considérée dans la structure, les conflits de partage de ressources surviennent entre chacun des alignements choisis et l'opération de maintenance sur la vanne 13. Les valeurs des gènes de paramétrage indiquent que la requête 1 est exécutée avant l'opération de maintenance (i.e. $V_{1,1;13} = \varepsilon$ et $V_{13;1,1} = e$) et que l'opération 2 est exécutée également avant l'opération de maintenance (i.e. $V_{2,4;13} = \varepsilon$ et $V_{13;2,4} = e$). Ainsi, aucun conflit n'émerge entre les 2 alignements choisis, ce qui implique que les opérations de transfert peuvent être exécutées simultanément. Les conflits pour les opérations peuvent être déduits directement de la Fig. 3.4, ainsi que les alignements choisis sont des alignements disjoints. Cette solution (Fig. 3.5d) montre alors un ordonnancement qualitatif où les opérations de transfert s'exécutent simultanément, suivies par l'opération de maintenance. Cet exemple vise à illustrer la transposition de l'information encodée par un chromosome et la solution du problème, même si l'ordonnancement reste qualitatif puisque les requêtes sont de durée identique et sans dateurs spécifiques.

Les points suivants synthétisent la proposition exposée :

- *Solution au problème* : une solution ou individu est encodé en termes de variables caractérisant la structure d'affectation des ressources (i.e. alignements), et de variables qui permettent d'ordonnancer les opérations en fonction de cette structure.
- *Réduction de l'espace de recherche* : chaque désactivation d'un gène de contrôle entraîne au moins une désactivation d'un gène de paramétrage, qui simultanément désactive son gène complémentaire. Sans connaissance de la structure algébrique du système et de ses propriétés, un algorithme pourrait tester un grand nombre de permutations de valeurs pour le chromosome entier tel que éventuellement les contraintes soient satisfaites. De nombreux algorithmes proposent une pénalisation a posteriori de la fitness des solutions qui sortent de l'espace de faisabilité, afin qu'elles soient éventuellement rejetées pour les générations suivantes ; [69] peut être consulté pour une description plus détaillée des techniques de gestion de contraintes. Pour l'exemple présenté, $2 \times 14 = 28$ possibilités pourraient définir un individu (i.e. 14 gènes affectés avec 2 valeurs possibles). Ici, du fait du contrôle hiérarchique et de la vérification simple et directe de faisabilité, 4 affectations sont possibles pour les gènes de contrôle, et 4 possibilités d'affectation pour les gènes de paramétrage, réduisant ainsi le temps de calcul.

3.3.2 Opérateur de Mutation

Comme proposé dans [68], il est retenu que dans la phase initiale d'évolution, une modification génétique de la structure des solutions (gènes de contrôle) soit appliquée. Par contre, vers les générations finales, la modification génétique vise les paramètres, qui améliorent la qualité des solutions correspondants aux structures déjà identifiées comme étant les plus intéressantes. L'opération de mutation est appliquée avec une *probabilité de mutation* p_m , paramètre de l'algorithme. Sachant que, si $|J_i|$ est le nombre d'alignements candidats pour la requête i , le génome aura autant de gènes de contrôle pour cette requête. L'implémentation de l'opérateur de mutation est proposée ensuite selon le comportement déclenché sur les gènes de structure et les gènes de paramétrage.

Gènes de Structure :

Un changement de valeur d'un gène de structure pour une requête entraîne le changement opposé sur un des gènes de contrôle pour la même requête, i.e. si l'alignement sélectionné initialement n'est plus choisi, alors un autre alignement doit être sélectionné (ou le cas contraire). La Fig. 3.6 illustre l'opération de mutation sur les gènes de contrôle considérant le codage issu de la Fig. 3.5. Tout d'abord, la solution courante montre une sélection des alignements 1 et 4 pour les requêtes 1 et 2, respectivement (i.e. ces gènes de structure prennent la valeur e). Ensuite, la mutation est appliquée, le premier gène est alors modifié par sa valeur opposée (i.e. ε) et le

génomique devient endommagé, signifiant qu'aucun alignement n'est choisi pour la requête 1. La réparation du génome se fait en propageant l'effet de la mutation sur le gène en question vers ses gènes de contrôle complémentaires. Comme dans l'exemple de la Fig. 3.6, chaque requête n'a que deux alignements possibles, la propagation se fait directement sur la seule variable restante pour la même requête (i.e. changement $\varepsilon \rightarrow e$ dans le deuxième gène de structure), et le génome est réparé au niveau de structure. Par contre, pour un changement initial du type $\varepsilon \rightarrow e$, il s'agit de changer la valeur du seul gène complémentaire affecté par e (i.e. changement $e \rightarrow \varepsilon$). Cette procédure rend le génome correct par la propagation des effets de mutation sur les gènes de structure. Néanmoins, pour plus de deux alignements par requête, si le changement initial est du type $e \rightarrow \varepsilon$, il faut choisir aléatoirement un des gènes de contrôle restants pour la même requête afin d'effectuer le changement $\varepsilon \rightarrow e$.

Gènes de paramétrage :

Considérant qu'une mutation correcte (i.e. avec réparation si nécessaire) est déjà appliquée sur les gènes de contrôle, la mutation doit être propagée vers les gènes de paramétrage. Cela implique la désactivation des gènes de paramétrage associés aux gènes de contrôle qui ont été désactivés (i.e. ces gènes de paramétrage ne sont plus considérés dans la solution). Dans l'exemple de la Fig. 3.6, le premier gène de contrôle a été désactivé, indiquant que l'alignement 1 n'est plus sélectionnée pour la requête 1. Par conséquent, ses gènes de paramétrage, qui sont les variables de décision pour les conflits entraînés par l'alignement 1, doivent être désactivés puisque ses conflits deviennent inexistantes. De manière analogue, comme le gène de contrôle 2 est activé suite à la réparation du génome, ceci signifie que l'alignement 2 est alors sélectionné pour la requête 1 et les variables de décision pour les conflits entraînés par l'alignement 2 doivent être affectées aléatoirement (i.e. activation des gènes de paramétrage). Un changement de valeur dans un gène de paramétrage entraîne forcément le changement opposé dans sa variable complémentaire. La Fig. 3.6 illustre cette propagation qui produit un nouvel individu : autre solution candidate. Un deuxième type de mutation sur les gènes de paramétrage peut survenir. Cette mutation (qui n'est pas issue de la propagation d'une mutation des gènes de contrôle) vise à optimiser les paramètres pour une structure qui a déjà vécu un certain nombre de générations (ce qui indique qu'il s'agit d'une structure optimale). Cette mutation est représentée par le même principe des gènes complémentaires : la structure reste identique, et lorsqu'un gène de paramétrage change de valeur, son gène complémentaire changera également de valeur, menant à une solution cohérente physiquement.

Une population initiale avec un haut niveau de diversité, et un bon paramétrage des opérateurs de mutation et croisement (définition des probabilités respectives) devra évoluer de telle sorte que les individus caractérisant des structures optimales obtiendront des valeurs de fitness plus

élevées, conduisant au passage de ces gènes aux générations suivantes. La réalisation d'un bon paramétrage n'est pas forcément évident et des essais successifs sont souvent nécessaires.

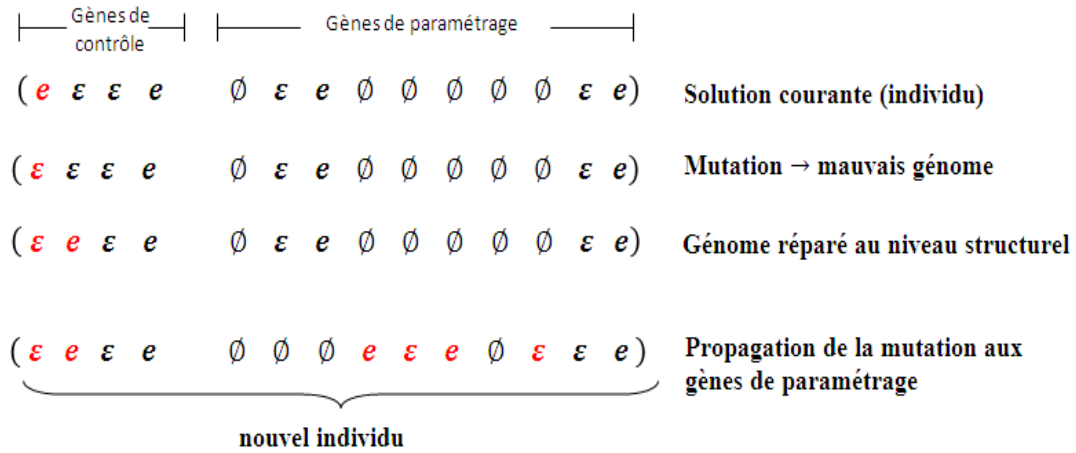


FIGURE 3.6: Mutation des gènes de contrôle et propagation vers les gènes de paramétrage - exemple.

3.3.3 Opérateur de Croisement

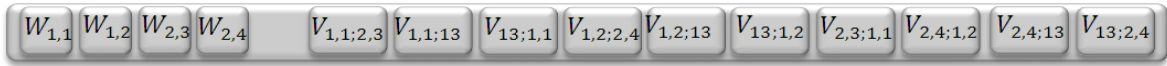
Comme pour l'opérateur de mutation, un croisement des gènes de structure entraîne des changements dans les gènes de paramétrage correspondants. L'idée sous-jacente est qu'à partir de deux chromosomes ou individus nommés parents, deux individus descendants sont générés avec une combinaison des caractéristiques génétiques de ses parents. Deux cas peuvent survenir, soit les parents ont une structure différente ou identique. Dans le premier cas, le croisement génère des individus qui diversifient la population en termes de structure ; et par conséquent de paramétrage. Dans le deuxième cas, le nouvel individu diversifie la population en termes de paramétrage tout en conservant la même structure. Considérant le facteur aléatoire associé à cet opérateur de *probabilité de croisement* p_c , si au bout d'un nombre de générations il surgit une tendance à ce que les parents choisis aléatoirement aient la même structure, il peut être déduit que cette caractéristique génétique a bien persisté lors de l'évolution et qu'elle représente une structure optimale. Par conséquent, dans les générations suivantes le croisement génère des individus avec structure optimale identique en variant les gènes de paramétrage pour trouver un paramétrage optimal, ce qui se traduit par une recherche de l'ordonnancement optimal d'opérations, sachant que la sélection d'alignements optimale a été trouvée. Les deux cas possibles de croisement sont décrits ensuite.

1. Si les parents possèdent de gènes de contrôle différents, les descendants héritent d'une structure 'mélangée' qui déclenche une série de changements dans les gènes de paramétrage. Comme cela a été indiqué pour l'opération de mutation, les gènes de structure désactivés déclenchent une désactivation de certains gènes de paramétrage. Ensuite, pour

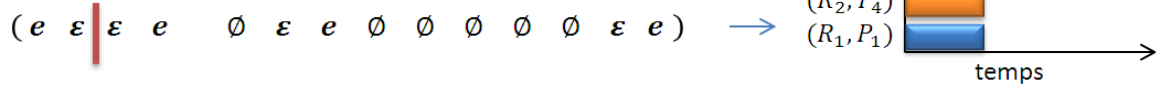
les gènes de paramétrage qui restent à définir, un choix aléatoire est réalisé, toujours respectant la notion de complémentarité des gènes. Un exemple illustre ce principe dans la Fig. 3.7, où la structure du génotype (Fig. 3.4) est rappelée. Chaque parent caractérise une solution qui est décrite par son chromosome, cette solution implique la sélection des alignements pour les opérations de transfert et un ordonnancement d'opérations spécifique. Une illustration qualitative de la solution décrite par chaque chromosome est présentée à droite de la Fig. 3.7. Les descendants dérivés sont des chromosomes issus d'une combinaison des caractéristiques structurelles des parents et les gènes de paramétrage à activer (dû au changement des gènes de structure) sont affectés de manière aléatoire.

2. Si les parents possèdent des structures identiques, de l'opération de croisement se dérivent des descendants de même caractéristique structurelle. Les gènes de paramétrage sont ainsi déterminés de manière aléatoire.

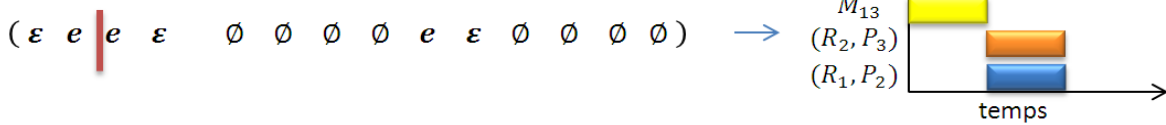
Génotype



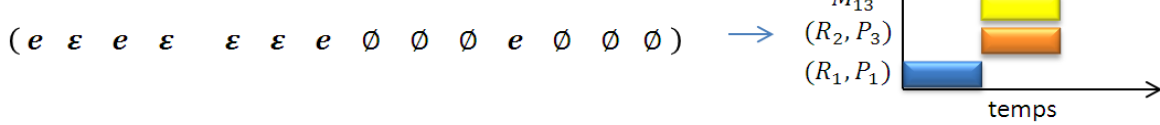
Parent P_1



Parent P_2



Descendant D_1



Descendant D_2

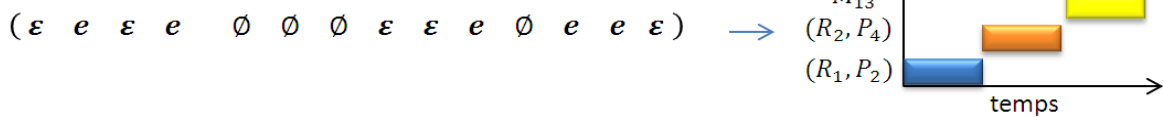


FIGURE 3.7: Opération de croisement avec des parents ayant des structures différentes appliquée à l'exemple de la Fig. 3.4.

Dans la Fig. 3.7, à chaque chromosome correspond une solution pour l'instance considérée. Chaque parent caractérise une solution différente et donne lieu à 2 autres solutions (i.e. les descendants) par l'opération de croisement. Le point de croisement est illustré par une ligne rouge dans les gènes de structure des parents. Pour cet exemple, cette position est le seul point cohérent possible puisque il n'y a que deux requêtes à traiter. Lorsque plus de 2 requêtes sont à traiter, le point de croisement est établi comme un point aléatoire localisé exclusivement dans les gènes de structure (i.e. gènes de contrôle), tel qu'il distingue les gènes associés à des requêtes différentes. Ceci évite des génomes endommagés dans les descendants tels que, par exemple, pour une requête spécifique, tous les gènes soient désactivés ou plusieurs gènes soient activés.

Un croisement de gènes de paramétrage n'est pas considéré car ces gènes dépendent directement des gènes de structure. Un croisement de gènes de paramétrage résulterait, dans la plupart des cas, en un chromosome endommagé représentant un ordonnancement incohérent et avec aucun lien avec les ressources déjà affectées (par les gènes de structure). Par exemple, si les alignements 1 et 3 sont affectées (gènes de structure) aux requêtes 1 et 2, respectivement, un croisement de gènes de paramétrage pourrait conduire à un ordonnancement d'opérations tel que l'opération 1 s'effectue sur l'alignement 2 avant que l'opération 2 ne s'effectue sur l'alignement 4. Ceci n'est pas cohérent avec les gènes de structure. En outre, d'autres incohérences pourraient survenir simultanément comme le non respect de la complémentarité des variables et l'affectation d'une requête à plus d'alignement.

Pour l'opération de croisement dans la Fig. 3.7, les gènes de structure du premier descendant D_1 combinent la première partie du parent P_1 et la deuxième partie du parent P_2 . Le descendant D_2 , combine la première partie du parent P_2 et la deuxième partie du parent P_1 . Pour chacun des descendants, à partir de ces gènes de structure, certains gènes de paramétrage sont désactivés (pour les gènes de contrôle inactifs) et les autres sont initialisés aléatoirement en cohérence avec la règle de complémentarité des variables.

À ce stade, le génome est défini, chaque individu représente une solution de sélection d'alignements ainsi que d'ordonnancement d'opérations de transfert et de maintenance. De plus, des règles de manipulation des gènes ont été établies dans l'application des opérations de mutation et de croisement. Le Tableau 3.1 synthétise le comportement proposé pour ces opérations. Un individu est ainsi complètement défini, ce qui permet la définition des règles d'évolution de chaque individu dans le cadre de sa population. Cette stratégie évolutionnaire est définie dans la section suivante.

TABLE 3.1: Synthèse du comportement proposé pour les opérateurs génétiques.

Opérateur de Mutation	Opérateur de Croisement
<i>Mutation d'un gène de contrôle :</i> Modifie l'affectation d'alignements et, par conséquent, l'ordonnancement d'opérations. Procédure : 1) 1 gène de contrôle est muté. 2) le génome endommagé est réparé au niveau de structure par affectation des gènes complémentaires (à celui qui vient de muter). 3) la réparation est propagée aux gènes de paramétrage. Les gènes à activer sont affectés par ε ou e de manière aléatoire, en cohérence avec les règles de complémentarité.	<i>Croisement par gènes de contrôle :</i> Modifie la sélection d'alignements et, par conséquent, l'ordonnancement d'opérations. Procédure : 1) un point de croisement est défini séparant 2 requêtes dans les gènes de structure. 2) le croisement est réalisé au niveau des gènes de structure. 3) Les gènes de paramétrage activés par rapport aux nouveaux gènes de structure sont affectés aléatoirement, par ε ou e, en cohérence avec les règles de complémentarité.
<i>Mutation d'un gène de paramétrage :</i> Modifie l'ordonnancement sans altérer les alignements affectés. Procédure : 1) 1 gène de paramétrage est muté. 2) le génome endommagé est réparé au niveau de paramétrage par affectation du gène complémentaire (à celui qui vient de muter).	

3.4 Stratégie Évolutionnaire et Intégration des Modèles Algébriques (Max,+)

Les algorithmes génétiques peuvent déterminer la meilleure affectation des variables de décision dans les modèles non linéaires (max,+). Le modèle algébrique (max,+) déjà obtenu doit être alors représenté sous forme de génotype.

Chaque chromosome caractérise un vecteur d'affectation de variables de décision. Considérant que les contraintes des dateurs dans le système correspondent à (3.1) et (3.2), et qu'elles deviennent linéaires lors de l'affectation de ces variables, alors chaque individu génère un système

d'équations linéaires $(\max, +)$ du type $X = AX \oplus b$ directement résolu par $X = A^*b$. De ce fait, émerge la notion d'intégration des modèles $(\max, +)$ avec l'approche d'algorithmes génétiques.

La stratégie évolutionnaire est la manière dont l'ordre partiel (i.e. $y_i \succ y_j$ ou relation de dominance entre solutions différentes) de la population peut être exploité pour trouver des solutions meilleures à chaque génération. Cet ordre partiel peut être obtenu à l'aide de la notion de *dominance de Pareto*. Dans le contexte mono-objectif, la notion de fitness est souvent équivalente à la valeur de la fonction objectif. Dans le contexte multi-objectif, cette notion de fitness d'un individu est associée à plusieurs objectifs, ainsi qu'à la relation de qualité entre l'individu en question et les autres individus de la population. Dans ce cadre, de nombreuses stratégies évolutionnaires permettent de déterminer la fitness d'un individu. Les travaux de [53] donnent une vue générale des diverses propositions. Parmi les plus connues, les stratégies basées sur le rang, la profondeur et le nombre de dominance peuvent être citées. Le *rang de dominance* détermine la valeur de fitness par rapport au nombre d'individus par lesquels un individu est dominé. La *profondeur de dominance* décompose la population sur plusieurs frontières et détermine la valeur de fitness par rapport à la profondeur, qui indique à quelle frontière chaque individu appartient. Le *nombre de dominance* détermine la fitness d'un individu par rapport au nombre d'individus qu'il domine.

Lors de l'évolution, pour maintenir la diversité de la population, de nombreuses stratégies utilisent la notion de densité, i.e. la probabilité de sélection pour un individu diminue lors que la densité d'individus dans son voisinage augmente. Parmi les techniques de gestion de densité se trouvent les histogrammes, le voisin le plus proche, et les techniques de kernel ; le lecteur peut se rapporter à [53] pour plus de détails.

Au même titre que la diversité et la fitness, l'élitisme est un des aspects déterminants dans le processus d'évolution. L'élitisme consiste à éviter que de bonnes solutions soient perdues lors du processus d'évolution. Cet aspect est souvent géré à travers d'une archive des meilleures solutions. Diverses techniques considèrent des archives de taille variable ou fixe.

Ainsi, en considérant l'ensemble des attributions marquant l'évolution de la population, tels que l'affectation de fitness, la gestion de la diversité, la politique de sélection (critère pour sélectionner les individus pour les opérations génétiques) et l'élitisme, de nombreuses stratégies évolutionnaires ont été développées et testées dans la littérature. Elles continuent à être un centre d'intérêt dans la recherche de l'intelligence artificielle. Sans être exhaustif, un rapide aperçu permet d'appréhender leur principe fondamental de fonctionnement. Parmi ces techniques il est retenu : le VEGA (*Vector-Evaluated Genetic Algorithm*), PAES (*Pareto Archived Evolution Strategy*), NSGA (*Non Dominated Sorted Genetic Algorithm*), NPGA (*Niched Pareto Genetic Algorithm*) et le SPEA (*Strength Pareto Evolutionary Algorithm*). Chacune d'entre elles gère de manière distincte la politique de classement d'individus, ainsi que les aspects d'élitisme et diversité. La stratégie SPEA a été sélectionnée du fait de son principe de fonctionnement assez

direct détaillé plus bas. Une recherche exhaustive sur le comportement des différentes stratégies évolutionnaires n'est pas le but de ce travail. Ainsi, d'autres choix pourraient être valables pour les objectifs de validation.

L'algorithme SPEA, ou Algorithme Évolutionnaire de Force de Pareto, est relativement récent comme le montre [57]. Il propose de considérer l'évolution de chaque individu par rapport à sa force (*'strength'* en anglais). Pour chaque individu, cet indicateur comptabilise le nombre d'individus dominés par l'individu en question. De manière générale, la force de chaque individu permet de calculer l'indicateur dénoté en anglais comme *«raw fitness»* (les détails sur sa détermination sont présentés plus bas après l'algorithme). Ici, pour faire référence à cet indicateur, le terme *«fitness pure»* sera employé. La fitness pure, liée à un indicateur de densité, permet de calculer la valeur de fitness de l'individu afin de déterminer s'il s'agit d'un individu dominé ou non dominé dans le sens de l'optimalité de Pareto.

Ainsi, la valeur de fitness, pour chaque individu de la population, détermine si les caractéristiques génétiques de cet individu seront retenues pour les générations qui suivent. L'algorithme SPEA2 est une version améliorée de l'algorithme SPEA. Les aspects améliorés ne sont pas objet d'analyse ici, mais incluent principalement l'intégration d'une estimation de densité pour mieux guider la recherche, [57]. Le principe de base de l'approche SPEA2 est qu'une population initiale est générée et celle-ci évolue au cours des générations suivantes. Pour chacune de ces générations, une archive est conservée en parallèle stockant les individus les plus forts (ou plus adaptés) trouvés lors de l'évolution, pour éviter que certaines solutions optimales soient perdues. A la fin de l'algorithme, l'archive contient les meilleures solutions. La structure générale de l'algorithme SPEA2 (proposée dans [57]) est décrite ensuite, suivie d'une explication détaillant chacune des phases et les calculs pour les indicateurs employés.

Entrées : N : taille de la population

$\bar{N}$: taille de l'archive

T : nombre maximum de générations

Sortie : A : ensemble non dominé

1. Initialisation : Générer une population initiale P_0 et créer l'archive vide (ensemble externe) $\bar{P}_0 = \emptyset$. Fixer $t = 0$ (i.e. première génération).
2. Affectation de Fitness : Calculer les valeurs de fitness pour les individus dans P_t et $\bar{P}_t$.
3. Sélection Environnementale : déterminer les individus non dominés contenus dans P_t et $\bar{P}_t$ et les copier dans $\bar{P}_{t+1}$. Si la taille de $\bar{P}_{t+1}$ dépasse $\bar{N}$, alors réduire $\bar{P}_{t+1}$ par un opérateur de troncation. Par contre, si la taille de $\bar{P}_{t+1}$ est inférieure à $\bar{N}$, alors terminer de remplir $\bar{P}_{t+1}$ avec des individus dominés dans P_t et $\bar{P}_t$.

4. Interruption : Si $t \geq T$ ou un autre critère d'arrêt est satisfait, alors A est l'ensemble des vecteurs de décision représentés par les individus non dominés dans $\overline{P_{t+1}}$. Arrêter.
 5. Sélection d'Accouplement (pour croisement) : application du tournoi binaire avec remplacement dans $\overline{P_{t+1}}$ pour remplir la piscine d'accouplement.
 6. Variation : Appliquer les opérateurs de croisement et mutation à la piscine d'accouplement, étant $\overline{P_{t+1}}$ la population résultante. Incrémenter le compteur de génération et aller au pas 2.
-

Dans la méthodologie présentée pour l'algorithme SPEA2, la phase d'initialisation est directe ; une population et une archive vide sont créées. Dans cette population, chaque individu a le type de structure proposée dans la Fig. 3.5b). De cette structure, les contraintes des valeurs possibles pour les variables de décision, i.e. e ou ε , sont déjà prises en compte lorsque le génotype approprié est défini. Ces contraintes couvrent les variables de décision de résolution de conflits et de sélection d'alignements, comme indiquées dans (2.30) où $B \rightarrow \varepsilon$. Les variables de décision pour les conflits entre les opérations de maintenance et de transfert, sont définies de manière analogue dans (3.8) :

$$V_{i,j;k} \otimes V_{k;i,j} = B \quad ; \quad V_{i,j;k} \oplus V_{k;i,j} = e \quad \forall i, j \in \mathcal{OA} \wedge k \in \mathcal{M} \quad (3.8)$$

Conservant la notation de [68], un individu quelconque est dénoté par i . Cette notation n'introduit pas d'ambiguïté avec les requêtes de transfert (aussi dénotées i dans ce document) puisqu'elle n'est utilisée que pour définir la méthodologie générale de l'algorithme dans cette section.

Dans l'étape 2 de l'algorithme, la valeur de fitness pour chaque individu est déterminée. Pour cela, premièrement la force de chaque individu i doit être déterminée par (3.9) où P_t représente la population à la génération t , $\overline{P_t}$ est l'archive correspondante à cette même génération, $i \succ j$ dénote la dominance d'un individu i sur un individu j (dans le sens de Pareto, i.e. $i \succ j$ si et seulement si i améliore certains objectifs de j sans détériorer d'autres), l'opérateur $'+''$ dénote l'union des ensembles et $|\cdot|$ dénote la cardinalité de l'ensemble. Ainsi, l'évaluation de la force d'un individu est effectuée sur l'ensemble des individus de la population et de l'archive actuelle.

$$S(i) = |\{ j \mid j \in (P_t + \overline{P_t}) \wedge i \succ j \}| \quad (3.9)$$

La fitness pure d'un individu i est déterminée comme la somme des forces des individus j qui dominant l'individu i (voir (3.10)). Il faut noter que dans ce cadre mathématique, le but est de minimiser la fitness. Ainsi, pour tout individu i non dominé $R(i) = 0$.

$$R(i) = \sum_{\substack{j \in (P_t + \overline{P}_t) \\ j \succ i}} S(j) \quad (3.10)$$

Plusieurs individus peuvent avoir la même valeur de fitness pure. Pour les discriminer, la notion de densité est appliquée. La densité est calculée dans (3.11), où σ_i^k représente la distance euclidienne dans l'espace d'objectifs jusqu'au voisin k pour chaque individu i . Le voisin k est souvent estimé avec $k = \sqrt{N + \overline{N}}$, [57]. Ainsi, pour chaque individu i , la distance de chacun de ses voisins j (dans la population, ainsi que dans l'archive) est déterminée et stockée dans une liste. Cette liste est mise en ordre de manière croissante, et la position k de la liste est prise en compte pour obtenir σ_i^k . Le dénominateur dans (3.11) est défini de façon à éviter la division par zéro et à imposer que $D(i) < 1$.

$$D(i) = \frac{1}{\sigma_i^k + 2} \quad (3.11)$$

Finalement, la valeur de fitness pour un individu i est obtenue par (3.12). D'après cette expression, il est clair que la stratégie évolutionnaire vise alors à minimiser $F(i)$.

$$F(i) = R(i) + D(i) \quad (3.12)$$

Une fois les valeurs de fitness affectées à tous les individus de la population et de l'archive, le processus de sélection peut être appliqué. Notamment, la sélection environnementale est définie par (3.13). Elle consiste à extraire les meilleurs individus de la génération pour appliquer ultérieurement des opérateurs génétiques et créer une nouvelle génération, sachant que l'archive stocke les meilleurs individus obtenus depuis la première génération.

$$\overline{P}_{t+1} = \{ i \mid i \in (P_t + \overline{P}_t) \wedge F(i) < 1 \} \quad (3.13)$$

Si le nombre d'individus non dominés (i.e. individus pour lesquels $F(i) < 1$) est égal à la taille de l'archive, cette sous-procédure s'achève. Sinon, deux situations peuvent survenir : selon que la taille de l'ensemble soit supérieure ou non à la taille de l'archive. Dans le premier cas, les meilleurs $\overline{N} - |\overline{P}_{t+1}|$ individus dominés sont copiés dans l'archive. Par contre, si le nombre d'individus non dominés dépasse la taille exigée pour l'archive, une procédure de troncature est appliquée pour éliminer des individus itérativement jusqu'à ce que $|\overline{P}_{t+1}| = \overline{N}$.

Une fois effectuée la sélection environnementale, la condition d'arrêt de l'algorithme est évaluée et si elle est vraie, alors l'archive $\overline{P}_{t+1}$ contient les meilleurs individus. Par contre, si l'exécution de l'algorithme continue, la sélection d'accouplement est effectuée. Cette étape implique la sélection aléatoire de deux paires d'individus pour déterminer pour chaque couple l'individu le plus fort, en vue de faire passer ses gènes à la génération suivante. Pour ces deux individus choisis, l'opération de croisement est effectuée avec une probabilité p_c , i.e. probabilité de croisement, définie comme

paramètre de l'algorithme. Le croisement génère 2 descendants qui remplaceront les parents dans la prochaine génération. Ensuite, l'opérateur de mutation est appliqué, aussi avec une probabilité de mutation p_m définie aussi en tant que paramètre de l'algorithme. Cette opération permet de diversifier l'ensemble de solutions.

Chaque individu représente une solution caractérisée par 2 valeurs pour les fonctions objectif du TCP et du TME. Ces valeurs sont déterminées par les dateurs du système. Ces dateurs sont à la fois calculés en fonction d'une affectation de variables de décision par (3.1) et (3.2). Ainsi, sachant que chaque individu représente une solution d'affectation des variables de décision générant un système $(\max, +)$ linéaire du type $X = AX \oplus b$, la Fig. 3.8 montre l'intégration de la modélisation mathématique en algèbre $(\max, +)$ et l'approche SPEA2, pour la résolution du problème d'optimisation multi-objectif considéré. Le principe de base est de partir du modèle algébrique $(\max, +)$ non linéaire, générer des individus caractérisant des valeurs pour les variables de décision et, pour chaque individu, résoudre le système d'équations $(\max, +)$ linéaire résultant pour obtenir les dateurs. Les dateurs déterminent les valeurs des fonctions objectif, ainsi que les valeurs de fitness qui permettent d'appliquer la variation génétique, obtenir une nouvelle population et répéter la procédure de manière itérative. Cette procédure est décrite en détail ensuite.

Le point de départ est la connaissance du fonctionnement du système et de sa modélisation algébrique en $(\max, +)$ par (3.1) et (3.2); voir bloc 1 : 'Modèle (Max,+) Non Linéaire' sur la Fig. 3.8. Comme indiqué, cette modélisation est concise, intuitive, et couvre des aspects de différentes natures tels que la gestion de partage de ressources, les dépendances d'arrivée du client et le début d'horizon de temps.

Ensuite, la génération de la population initiale (bloc 2) est réalisée par affectation aléatoire des variables de décision, en respectant l'ensemble de contraintes restreignant les valeurs acceptables (voir (2.30) et (3.8)), ce qui assure d'éviter de s'éloigner des solutions de l'espace de faisabilité.

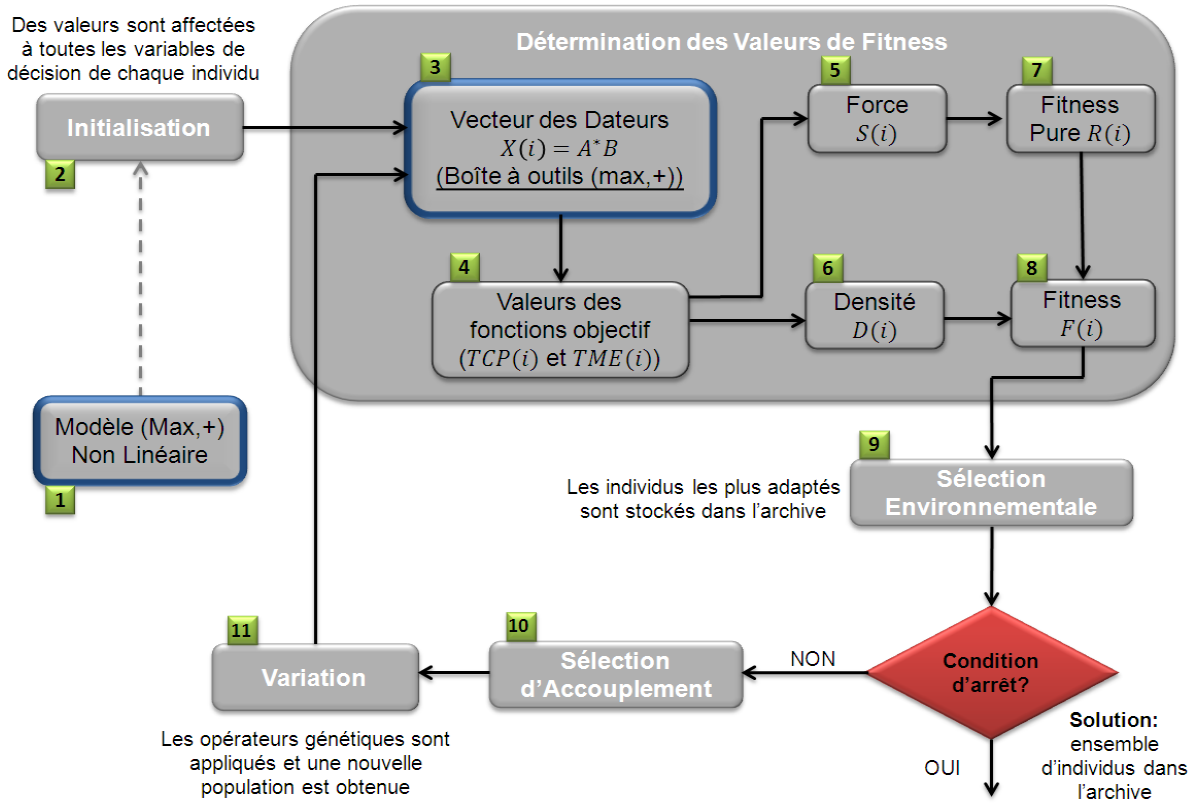


FIGURE 3.8: Flux d'exécution de l'algorithme SPEA2 avec des opérations en algèbre (max, +).

Une fois les individus générés, comme conséquence de l'affectation des variables de décision, un modèle (max,+) linéaire se déduit des équations non linéaires (3.1) et (3.2). Le système d'équations obtenu est du type $X = AX \oplus B$ qui peut être résolu par l'opération $X = A*B$ (voir [61]) où $A^* \stackrel{def}{=} \bigoplus_{n \in N} A^n$. Ce calcul est effectué par utilisation de la boîte à outils (max,+) dans Scicoslab. Le résultat est un vecteur de dateurs X (dates de démarrage des opérations) pour chacun des individus (bloc 3). Ce vecteur permet de calculer les valeurs des fonctions objectif : TCP et TME (bloc 4). Chacun de ces vecteurs représente une solution candidate au problème.

Une fois que chaque individu a été affecté de son vecteur de dateurs X , les relations de dominance de Pareto peuvent être déterminées pour tous les individus de la population. Ainsi, la force de chacun est déterminée selon (3.9), voir bloc 5. Ensuite, les valeurs de fitness pure (bloc 7) et densité (bloc 6) sont calculées selon (3.10) et (3.11), respectivement. Finalement, la valeur de fitness (bloc 8) est déterminée selon (3.12).

L'algorithme se poursuit en réalisant la sélection des meilleurs individus dans l'archive (bloc 9) et si la condition d'arrêt n'est pas atteinte, la sélection d'accouplement est effectuée (bloc 10) et les opérateurs génétiques sont appliqués (bloc 11). Une nouvelle génération est alors obtenue pour laquelle le vecteur de dateurs pour chaque individu est nouvellement calculé.

3.5 Résultats

L'instance de la Fig. 3.9 permet d'interpréter intuitivement les meilleures solutions et de les comparer avec les solutions apportées par l'approche proposée dans la Fig. 3.8. L'instance comporte 3 opérations de transfert, huit alignements et 1 opération de maintenance. Il est supposé que tous les tankers pour les requêtes arrivent dans leurs fenêtres de temps autorisées (ce qui implique que tout retard dans le service peut entraîner une pénalité pour le terminal maritime) à la date $u_i = 95 \quad \forall i = 1, 2, 3$. Cette instance vise à entraîner forcément des pénalités, étant donné les contraintes serrées des dates limites de service rappelées dans le Tableau 3.2. L'opération de maintenance est prévue à la date $t = 100$ avec une durée de 10 heures. L'algorithme SPEA2 a été implémenté directement dans Scicoslab afin d'exploiter la boîte à outils (max,+) dans la résolution des systèmes d'équations linéaires (max,+).

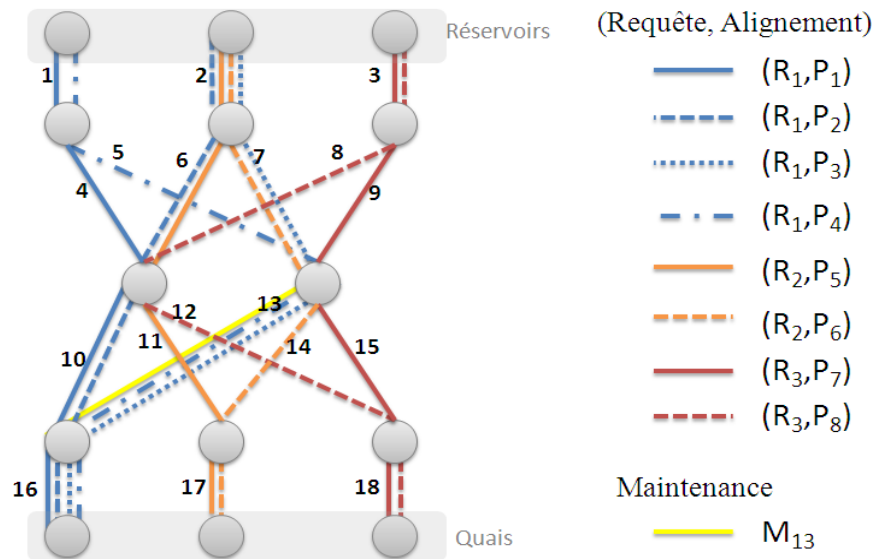


FIGURE 3.9: Exemple pour validation de l'approche intégrée SPEA2 + modélisation (max,+).

TABLE 3.2: Données d'entrée pour les opérations de transfert dans l'instance de la Fig. 3.9.

Opération de Transfert	Temps de Traitement (heures)	Pénalité (\$/heure)	Date Limite
R_1	30	5000	131
R_2	20	4000	131
R_3	20	2000	131

Les paramètres sélectionnés sont : taille de population (N) : 30, nombre de générations (ng) : 10, taille de l'archive ($\bar{N}$) : 5, probabilité de mutation (p_m) : 0.6 et probabilité de croisement (p_c) : 0.6.

Les résultats obtenus sont présentés par la Fig. 3.10, où les individus sont définis dans l'espace de critères TME vs TCP. Il est rappelé que chaque individu est représenté par un chromosome décrivant l'affectation des variables de décision gérant une sélection d'alignements et un vecteur de dateurs pour les opérations de transfert et de maintenance. Ainsi, chaque individu représente une solution au problème répondant aux objectifs du TCP et du TME ².

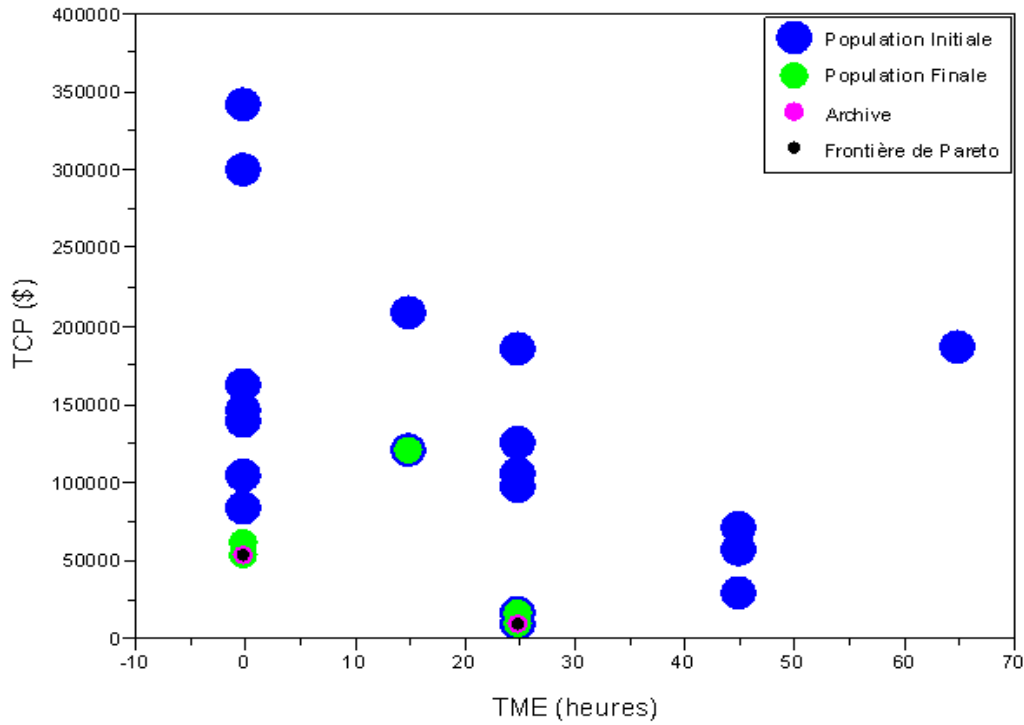


FIGURE 3.10: Solutions dans l'espace de critères pour l'instance de la Fig. 3.9.

Comme espéré, lors de l'évolution, les individus convergent vers les meilleures solutions possibles. Pour les deux solutions inscrites dans la frontière de Pareto, il peut être constaté qu'un des objectifs est amélioré au détriment de l'autre. En outre, aucun autre individu n'apporte d'amélioration des deux objectifs simultanément. Ces deux individus sont dénotés comme l'ensemble d'individus non dominés. Dans la pratique, ces individus non dominés caractérisent les solutions figurant en Fig. 3.11. Les deux individus non dominés caractérisent la sélection des alignements P_4 , P_5 et P_8 , et la différence entre les deux est l'affectation des variables pour la précedence entre les opérations, i.e. le compromis entre décaler ou pas l'opération de maintenance.

Prenant la Fig. 3.11b) pour analyse, l'affectation des alignements P_4 , P_5 et P_8 aux requêtes R_1 , R_2 et R_3 , respectivement, permet une exécution parallèle de R_1 et R_2 , puis une exécution parallèle de R_1 et R_3 , et ensuite une exécution parallèle de R_3 et M_{13} (une fois que R_1 est achevée).

2. Les mêmes résultats peuvent être obtenus avec un effort de calcul moins important ; e.g. population de 10 individus et 3 générations. Néanmoins, les paramètres choisis visent à illustrer une différence plus importante entre les populations initiale et finale, ainsi qu'avec la frontière de Pareto pour une meilleure appréhension.

Toute autre affectation d'alignements génère des solutions moins désirables. Par exemple, la sélection P_1 , P_6 et P_7 , en comparaison avec la Fig. 3.11b), garderait le même ordonnancement entre les opérations de transfert (i.e. même TCP), mais entraînerait un décalage plus important de l'opération de maintenance puisque il faudrait attendre la fin d'exécution de R_3 (voir conflit entre P_7 et M_{13} en Fig. 3.10), augmentant ainsi le TME.

La démarche présentée est ainsi capable de proposer à l'opérateur de supervision des solutions optimales. La décision resterait basée sur les conditions d'exploitation en temps réel. Dans un système automatisé, pour chacune des solutions présentées dans la Fig. 3.10, l'interface de supervision devrait permettre à l'opérateur de cliquer sur chacune d'entre elles, générant le détail de chaque solution (Fig. 3.11 par exemple).

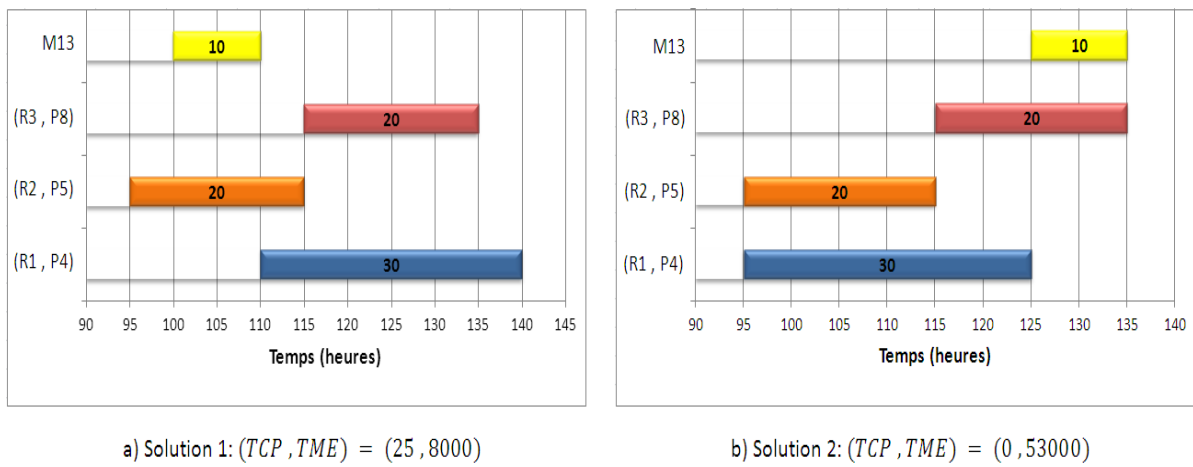


FIGURE 3.11: Solutions représentées par les individus non dominés dans la Fig. 3.10.

Le problème d'optimisation multi-objectif est résolu à travers une approche basée sur des algorithmes génétiques en combinaison avec une modélisation $(\max, +)$. Les résultats déjà obtenus sur la modélisation algébrique $(\max, +)$, qui avaient été validés à l'aide du logiciel d'optimisation (LINGO) dans la section 2.2.4, ont été exploités dans la technique de résolution proposée. Il est à remarquer que les résultats sur la modélisation algébrique avaient déjà permis de réduire les efforts de modélisation et l'intégration de nouveaux aspects, tels que la sélection d'alignements, grâce au caractère intuitif et concis de l'algèbre. Ainsi, ces propriétés ont été exploitées dans un contexte heuristique et ont permis de définir une structure génomique évitant de modéliser et de traiter des fonctions de contraintes dans l'algorithme génétique, même si des approches déjà existantes pourraient les intégrer dans le cadre de la fonction de fitness.

L'ensemble des solutions optimales peut être fourni à l'opérateur de supervision pour soutenir sa prise de décision. Chaque solution optimale correspond à un ensemble d'alignements sélectionnés pour les requêtes et un ordonnancement d'opérations (de transfert et de maintenance) sur un

compromis de minimisation du TCP et du TME.

Conclusions Partielles

Des approches classiques d'algorithmes génétiques pourraient considérer une affectation aléatoire et indépendante des gènes qui devrait aboutir aux mêmes résultats avec un paramétrage adéquat. Une modélisation classique pour les contraintes opérationnelles supposerait des contraintes de dépendance d'arrivée du tanker, du début de l'horizon de temps, et des contraintes de conflit de partage de ressources. Ainsi, lors d'une affectation aléatoire des gènes, les individus caractérisant des solutions infaisables seraient pénalisés dans les valeurs de fitness. Pour chaque individu, les dateurs devraient être calculés par évaluation des différents types de contraintes.

L'avantage de l'approche proposée ici est que l'algèbre $(\max, +)$ permet de formuler intuitivement, et dans une équation unique les différents types de contraintes, de telle manière que l'affectation des variables de décision conduit à un système d'équations $(\max, +)$ linéaire pour lequel les dateurs peuvent être déterminés par produit matriciel. Sachant que les conflits impactent l'ordonnancement d'opérations et qu'ils dépendent de l'affectation d'alignements, une structure génomique hiérarchique permet de gérer l'affectation des variables de manière plus efficace. Cette efficacité dérive de l'adaptation des opérations de mutation et croisement en fonction de cette hiérarchie, et des contraintes de complémentarité de variables de décision.

L'intérêt dans les techniques formelles de résolution, en contraste avec les techniques approximatives (telles que les algorithmes génétiques), a conduit à l'exploration des représentations basées sur les automates tropicaux (intégration de la théorie d'automates et les algèbres tropicales). Ces investigations ont conduit à proposer de modéliser l'optimisation de l'ordonnancement à l'aide d'automates tropicaux.

Chapitre 4

Modélisation à Base d'Automates Tropicaux pour l'Optimisation de l'Ordonnancement d'Opérations

Préambule

Les approches formelles pour modéliser la dynamique des systèmes à événements discrets composés et conflictuels sont exploitées à l'aide d'automates tropicaux. Une modélisation intuitive des conflits de partage de ressources est proposée au travers d'automates locaux, et un nouveau produit synchrone d'automates est envisagé pour déterminer un comportement global au plus tôt (avec un temps d'exécution minimum de toutes les opérations). Les principes de modélisation et de détermination de l'alphabet du nouveau produit sont présentés. Un cadre théorique basé sur les dateurs est alors proposé et illustré par de nombreux exemples. La problématique abordée avec cette technique reste une problématique d'engagement de ressources, pour un comportement au plus tôt.

Sommaire

4.1 Automates $(\text{Min}, +)$ et $(\text{Max}, +)$: Notions Théoriques	85
4.2 Modélisation des Conflits	89
4.3 Bases Théoriques pour l'Alphabet Proposé	91
4.4 Alphabet du Nouveau Produit d'Automates Tropicaux	97
4.4.1 Alphabet pour une Paire d'Automates Synchronisés dans un Système à 3 Automates ($n = 3$)	99
4.4.2 Alphabet pour Trois Paires d'Automates en Synchronisation dans un Système à 3 Automates ($n = 3$)	103

4.4.3 Alphabet pour un Système à 3 Automates ($n = 3$) avec Tierce Synchronisée	106
4.4.4 Exemple pour un Système à 3 Automates ($n = 3$)	106
4.4.5 Extensions Envisagées pour $n > 3$	110

4.1 Automates $(\min, +)$ et $(\max, +)$: Notions Théoriques

La théorie d'automates $(\max, +)$ et $(\min, +)$ sera présentée succinctement avant son exploitation pour un ordonnancement d'opérations au plus tôt.

Il est rappelé que le terme 'opération' fait référence à une opération de transfert de pétrole ou à une opération de maintenance de manière indistincte, et que les ressources correspondent aux vannes dans le réseau. Aussi, la terminologie employée inclut les notions de parallélisme, de simultanéité, et de synchronisation, en cohérence avec la théorie classique des systèmes à événements discrets. Néanmoins, afin d'éviter des ambiguïtés, ici 2 automates sont dits en *synchronisation* lorsqu'ils exécutent la même opération, i.e. de manière synchrone début et fin. Par contre, 2 opérations à durées quelconques s'exécutent *en parallèle* ou *simultanément* si elles se chevauchent dans l'axe du temps ; pourtant, elles ne démarrent, ni ne se terminent forcément à la même date.

Un automate $(\min, +)$ est un automate dans lequel chaque transition a un poids associé qui peut être interprété comme un coût. Chaque chemin dans l'automate peut être décrit avec un mot w dont le coût est la somme, en algèbre conventionnelle, des coûts des transitions. Le coût associé au mot w est le coût *minimal* parmi les coûts de tous les chemins qui peuvent être décrits avec le même mot w dans l'automate. Un automate $(\min, +)$ peut aussi être dénoté dans la littérature comme un *automate de distance* (voir [70] et [71]).

Réciproquement, un automate $(\max, +)$ a une représentation duale dont les poids des transitions peuvent être associées à des gains. Ainsi, un chemin décrit par le mot w est le chemin de poids maximal parmi tous les chemins décrits par le même mot w , i.e. le chemin de gain maximal.

Ces notions de maximisation et minimisation sont représentées par le semi-anneau dans lequel l'automate est défini. Dans ce contexte, les structures des algèbres $(\max, +)$ et $(\min, +)$ inspirent la définition des automates $(\max, +)$ et $(\min, +)$, respectivement. Ici, le terme 'automate tropical' fera référence à un automate qui peut être défini dans le cadre des algèbres tropicales ; notamment, soit l'algèbre $(\min, +)$ ou $(\max, +)$.

L'algèbre $(\max, +)$ a déjà été définie dans le chapitre 2 et l'algèbre $(\min, +)$ décrit un comportement dual tel que les deux opérations mathématiques de base $\oplus$ et $\otimes$ correspondent à la minimisation et l'addition, respectivement. Les propriétés de commutativité, associativité, distributivité et idempotence restent valables. Les éléments nul et neutre correspondent à $\varepsilon = \infty$ et $e = 0$, respectivement.

Un automate $(\min, +)$ est défini formellement comme $G = (Q, A, \alpha, t, \beta)$ où A est l'alphabet (ou ensemble d'événements), Q est l'ensemble fini d'états, $t : Q \times A \times Q \rightarrow \mathbb{N}$ est la fonction de transition qui exprime une durée, et $\alpha : Q \rightarrow \mathbb{N}$ et $\beta : Q \rightarrow \mathbb{N}$ sont les délais initial et final, respectivement. Dans ce contexte, les poids des transitions appartiennent à l'ensemble des

nombre naturels. Deux états, q et q' , sont liés par une transition étiquetée par un événement a de durée $t(q, a, q')$. Si aucune transition ne relie ces deux états, alors $t(q, a, q') = \varepsilon = \infty$. Dans un automate $(\min, +)$ (respectivement $(\max, +)$), la durée d'un événement a peut être représentée par une variable $y(a)$ nommée *dateur*. Un dateur peut aussi modéliser la date d'achèvement d'une séquence d'événements consécutifs, e.g. $y(ab)$ est le dateur pour la séquence 'ab'. La définition syntaxique d'un automate $(\max, +)$ ou $(\min, +)$ est donnée par son alphabet, la fonction de transition et l'ensemble d'états. Dans les travaux de [71], la définition syntaxique est dénotée : $G = (A, Q, I, F, t)$ où I représente l'ensemble d'états initiaux et F l'ensemble d'états finaux.

Réciproquement, un automate $(\max, +)$ est défini dans un semi-anneau avec une opération intrinsèque de *maximisation*. Dans un automate $(\max, +)$, lorsque aucune transition a ne relie deux états q et q' alors $t(q, a, q') = \varepsilon = -\infty$. Les automates $(\max, +)$ ont été largement étudiés dans la littérature et quelques résultats sont transposables aux automates $(\min, +)$ par dualité entre les deux techniques de modélisation. Les travaux de [40] définissent les bases pour l'application de ces techniques à la modélisation des systèmes à événements discrets et [42] et [43] abordent des applications de modélisation avec des automates $(\max, +)$.

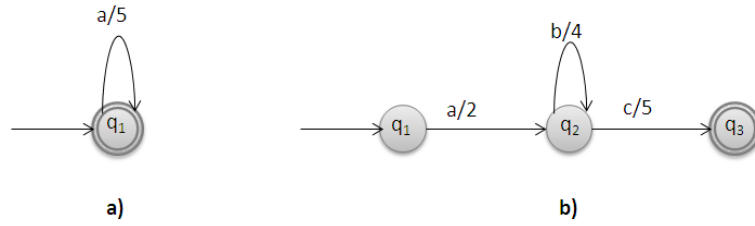


FIGURE 4.1: Exemples d'automates tropicaux : a) automate à 1 état. b) automate à plusieurs états.

La Fig. 4.1 présente 2 exemples d'automates tropicaux avec 1 ou plusieurs états. Les états initiaux sont représentés avec des flèches entrantes et les états finaux avec un double contour. Ces automates $(\max, +)$ ou $(\min, +)$ ne peuvent pas être différenciés par sa représentation syntaxique ni graphique, d'où le terme 'tropicaux'¹. Pour l'automate de la Fig. 4.1a), la syntaxe est la suivante : $A = \{a\}$, $Q = \{q_1\}$, $t(q_1, a, q_1) = 5$. Cet exemple comporte un seul état (état initial et final identique) et une seule transition représentant l'exécution de l'événement 'a'. Dans la pratique cet événement ferait référence à l'exécution d'une tâche, opération, procédure ou autre, d'une durée de 5 unités de temps. Ainsi, pour l'événement 'a' le dateur correspond à $y(a) = 5$, et le dateur de la séquence 'aa', par exemple, correspondrait à $y(aa) = 10$, i.e. l'opération 'a' est exécutée 2 fois en 10 unités de temps. De manière analogue, au niveau syntaxique, l'automate de la Fig. 4.1b) est représenté par $A = \{a, b, c\}$, $Q = \{q_1, q_2, q_3\}$, $t(q_1, a, q_2) = 2$, $t(q_2, b, q_2) = 4$ et $t(q_2, c, q_3) = 5$. Ainsi, les dateurs des événements sont : $y(a) = 2$, $y(b) = 4$ et $y(c) = 5$ et les

1. Dans la littérature le terme 'automate à multiplicités' est aussi employé. Néanmoins, il fait référence à des automates avec des transitions pondérées, une classe plus générale d'automates où les semi-anneaux de définition ne sont pas forcément ceux des algèbres $(\min, +)$ et $(\max, +)$.

dateurs des séquences ‘ab’, ‘abc’ et ‘abbc’ seraient : $y(ab) = 6$, $y(abc) = 11$ et $y(abbc) = 15$. Il est à remarquer que les automates de la Fig. 4.1 peuvent représenter soit des automates $(\max, +)$ ou soit des automates $(\min, +)$. La définition sémantique de ces automates détermine alors s’il s’agit d’un automate $(\max, +)$ ou $(\min, +)$ vis-à-vis de la recherche des chemins étiquetés par un mot. Dans le cas d’un automate $(\max, +)$ il s’agit de trouver le chemin à poids maximal parmi les chemins décrits par le même mot, et dans le cas d’un automate $(\min, +)$ il s’agit de trouver le chemin à poids minimal. Dans ce qui suit, seule la syntaxe des automates est étudiée. Par conséquent, le terme ‘automate tropical’ sera conservé².

Les développements dans ce chapitre ont été inspirés par les travaux de [44] et [72]. Dans [44] un produit synchrone d’automates $(\max, +)$ est proposé pour des systèmes de 2 et 3 automates locaux. Dans [72], sur les bases d’une modélisation avec des réseaux de Petri, le comportement est analysé par une décomposition d’états en micro-états dénotant le passage du temps, où les macro-états (regroupant plusieurs micro-états) dénotent le changement de marquage dans le réseau.

Dans [44] un produit synchrone pour automates $(\max, +)$ permettant l’exécution simultanée d’opérations a été proposé. Dans ce cas, il est supposé que les opérations simultanées doivent démarrer à la même date, et que la date d’achèvement de l’ensemble d’opérations est unique et correspond à la date d’achèvement de l’opération la plus longue. Par exemple, si les événements a et b sont exécutés simultanément et que $y(a) = 4$ et $y(b) = 10$, alors un nouvel événement (a, b) est généré dans l’alphabet du produit dénotant l’exécution simultanée de a et b , et sa durée est définie comme $y(a, b) = y(a) \oplus y(b) = 4 \oplus 10 = 10$. Ce comportement force les ressources associées à l’opération a à rester oisives jusqu’à ce que b soit achevée, dû à la synchronisation forcée par l’opérateur de maximisation $(\oplus)$. Par conséquent, les ressources de l’opération plus courte ne peuvent pas être exploitées par une autre opération dès qu’elles sont libérées par la première. Le même comportement décrit des situations avec plus de deux opérations en exécution simultanée. Néanmoins, ceci est bien l’objectif de ce produit, i.e. la détermination des chemins à poids maximal.

Afin d’illustrer le principe du produit proposé dans [44] sans rentrer dans sa formalisation (abordée dans la section 4.4), la Fig. 4.2 présente le produit synchrone de 2 automates par l’approche de [44]. Dans l’automate produit (automate c)) des événements composés dénotent l’exécution simultanée de 2 opérations avec la durée de l’opération la plus longue, i.e. $(b, c)/10$, et l’événement commun ‘a’ synchronise les automates. Dans la Fig. 4.3, le produit est appliqué à 3 automates résultant (automate (d)). La composition d’événements mène toujours à une maximisation des durées ce qui génère 2 trajectoires vers des états marqués de 25 et 17 unités de temps. La transition ‘1,1,1’ $\rightarrow$ ‘2,2,2’ de l’automate (d) indique que l’exécution de a et b s’achève après 15

2. puisqu’un automate $(\max, +)$ ne peut être différencié d’un automate $(\min, +)$ à partir d’une représentation syntaxique.

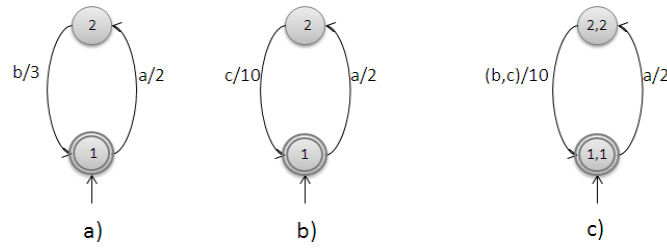


FIGURE 4.2: Produit synchrone 2 automates : l'automate c) c'est le produit des automates a) et b).

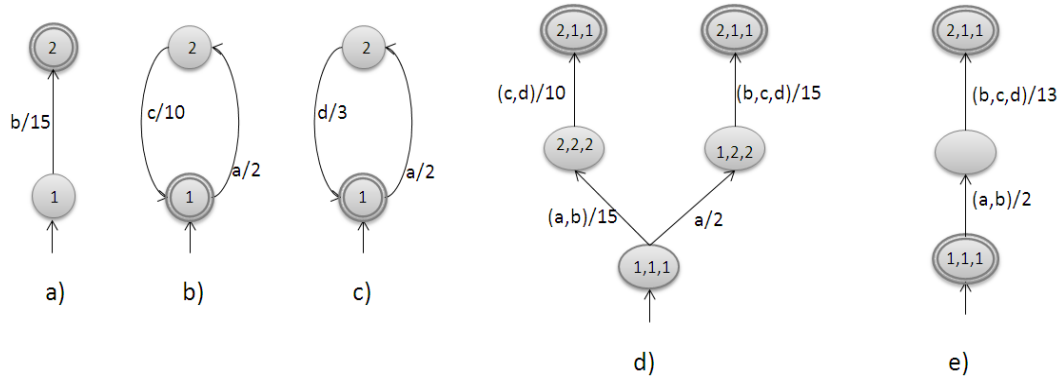


FIGURE 4.3: Produit synchrone de 3 automates : l'automate d) c'est le produit des automates a), b) et c). L'automate e) représente le produit souhaité pour le cas d'étude.

unités de temps, quand dans la pratique les ressources associées à l'opération a se libèrent après 2 unités de temps. Ce comportement est naturel dans un cadre de maximisation des poids des chemins comme celui abordé par les automates $(\max, +)$. Néanmoins, pour une problématique de fonctionnement au plus tôt i.e. élimination des périodes dites oisives des ressources, l'intérêt se focalise sur la minimisation du temps d'exécution de toutes les opérations possibles dans le système. Il paraît alors nécessaire de définir un produit synchrone plus adapté en conservant une représentation similaire à celle proposée en Fig. 4.3e). Dans cette représentation, l'opération b est décomposée de telle manière qu'une partie est exécutée en parallèle avec l'opération a (jusqu'à un état auxiliaire) et le reste en parallèle avec les opérations c et d (jusqu'à l'état final), résultant en une trajectoire de 15 unités de temps³. Ceci illustre la formulation d'un nouveau produit afin d'optimiser l'exploitation des ressources pour une minimisation du temps d'exécution total.

L'approche décrite dans [44] est pratique et concise. Cependant, elle vise à maximiser les poids des chemins et ne représente pas une approche suffisamment adéquate pour minimiser le temps d'exécution de l'ensemble des opérations dans un horizon de temps. En outre, l'application directe de la fonction de minimisation dans le cadre d'un produit dual avec des automates $(\min, +)$ n'est pas envisageable, puisque le minimum de 2 dateurs (pour deux opérations en exécution simultanée) ne modélise pas le temps d'achèvement de l'opération la plus longue et,

3. les autres opérations pourraient être aussi décomposées. Pourtant, le but est de représenter la durée de la trajectoire qui serait toujours de 15 unités de temps.

par conséquent, la libération des ressources associées. En outre, la résolution des conflits avec optimisation des engagements d'opérations simultanées, doit être extrapolable pour des sous-systèmes de plus grande taille.

4.2 Modélisation des Conflits

Cette section introduit le cadre de définition d'un nouveau produit synchrone modélisant le comportement global d'un système à conflit d'opérations dû au partage de ressources. Il est proposé que ces conflits peuvent être modélisés par des automates tropicaux. Seul les aspects de syntaxe sont abordés dans ces travaux, notamment la caractérisation de l'alphabet.

Le comportement logique fondamental modélisé par des 'automates tropicaux', ou tout simplement 'automates' dans ce qui suit, concerne l'exclusion mutuelle entre les opérations pour un ordonnancement global. Ainsi, n événements (i.e. opérations) mutuellement exclusifs sont modélisés par un automate ayant n arcs différents sortant de l'état initial.

Les opérations sans conflit mutuel sont modélisées comme des opérations exécutables dans des automates appelés 'locaux' différents pouvant s'engager simultanément. Ainsi, les automates modélisent à la fois les conflits et les opérations pouvant s'exécuter en parallèle.

La Fig. 4.4 montre un exemple de 4 opérations de transfert (r_i) devant être exécutées par les alignements signalés dans le réseau (seules les vannes ouvertes sont représentées en couleur) lorsque 2 activités de maintenance, m_2 , et m_6 , doivent également être lancées sur les vannes 2 et 6. Dans cette figure, et celles qui suivent, les indices de labellisation des vannes restent les mêmes dans le réseau de pipelines et ne doivent pas être confondus avec les durées des opérations qui sont illustrées dans les représentations par automates (G_i), e.g. l'opération r_1 nécessite l'ouverture des vannes : 1, 4, 10 et 16 (réseau de pipelines de Fig.4.4) et sa durée est de 6 unités de temps (G_2 dans Fig. 4.4).

Dans la Fig. 4.4, les opérations m_2 , r_2 et m_6 sont en conflit, ce qui est représenté par l'automate G_1 . Ces opérations sont conflictuelles car :

- 1) pour exécuter r_2 , les vannes 7, 4, 8, 11, 12, et 13 doivent être fermées, et les vannes 2, 6, 10, et 16 doivent être ouvertes ;
- 2) pour exécuter m_2 , les vannes 7 et 6 doivent être fermées pour que la vanne 2 soit isolée ;
- 3) pour exécuter m_6 , les vannes 2, 7, 4, 8, 10, 11, et 12 doivent être fermées pour isoler la vanne 6.

Localement sur G_1 , une seule de ces opérations ne peut être exécutée à la fois (i.e. représentant l'engagement des ressources pour exécuter une des opérations qui les sollicitent). Chaque transition a une étiquette avec l'événement déclencheur de l'opération et sa durée. L'état initial est

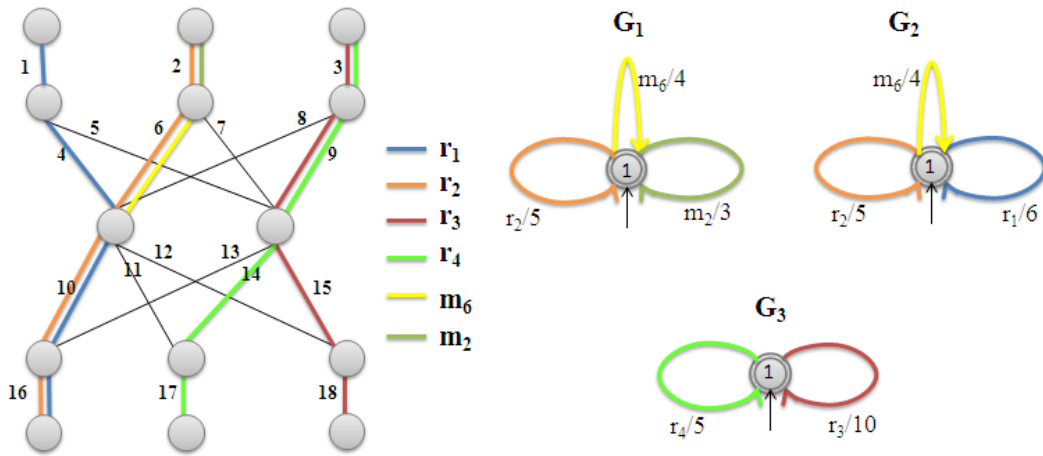


FIGURE 4.4: Exemple de modélisation des conflits par automates tropicaux mono-état : les opérations mutuellement exclusives sont modélisées par des transitions sortant du même état.

représenté avec une flèche entrante, et l'état final est représenté avec un double cercle ; i.e. état identique pour un automate mono-état.

La représentation par un automate mono-état (Fig. 4.4) illustre de manière concise le principe d'exclusion mutuelle pour les opérations en conflit du fait du partage de ressources. Formellement, chaque automate serait défini de manière générique comme $G_i = (Q_i, A_i, \alpha_i, t_i, \beta_i)$. Chaque automate possède un seul état, i.e. $Q_i = \{1\} \forall i \in \{1, 2, 3\}$, et les délais initial et final (qui retardent le début d'exécution à l'état initial et l'achèvement d'exécution à l'état final) sont négligeables ici, i.e. $\alpha_i = \beta_i = e \quad \forall i \in \{1, 2, 3\}$, et seule les durées d'exécution des opérations importe. Les alphabets sont $A_1 = \{r_2, m_6, m_2\}$, $A_2 = \{r_2, m_6, r_1\}$, et $A_3 = \{r_4, r_3\}$, et finalement les transitions peuvent être dénotées de manière générique comme $t_i(q_i, a, q'_i)$; e.g. $t_1(1, m_2, 1) = 3$ pour l'automate G_1 , $t_2(1, r_1, 1) = 6$ pour l'automate G_2 , et $t_3(1, r_3, 1) = 10$ pour l'automate G_3 .

Le modèle est interprété comme suit : les opérations r_2 , m_6 , et m_2 sont toutes en conflit puisqu'elles ne peuvent pas être exécutées simultanément. Ce conflit est modélisé par l'automate G_1 . Les transitions sortantes de l'état unique modélisent le fait qu'une seule de ces opérations peut s'exécuter à la fois. De manière analogue, r_1 , r_2 , et m_6 sont des opérations aussi en exclusion mutuelle (i.e. sont aussi en conflit), et par conséquent génèrent l'automate G_2 . Finalement, G_3 modélise le conflit entre r_4 et r_3 . Le modèle décrit aussi des opérations indépendantes (i.e. sans conflit) qui peuvent être exécutées simultanément (comme b et c pour la Fig. 4.2) telles que m_2 , r_1 et r_3 , par exemple. Ainsi, l'intérêt d'un modèle à un seul état par automate est qu'il modélise l'exclusion mutuelle de manière concise.

Hypothèse 4.1. Toutes les opérations doivent être exécutées, et chaque opération qu'une seule fois et de manière ininterrompue.

L'hypothèse 4.1 restreint le comportement du modèle (i.e. ensemble d'opérations à exécuter dans un système non périodique). De manière rigoureuse, la représentation dans la Fig. 4.4 permet que chaque opération puisse être exécutée plus d'une fois. Pour représenter de manière précise les contraintes considérées, i.e. exécution unique, un modèle étendu est défini avec l'ensemble des permutations permises, ce qui restreint le comportement logique pour que les cycles ne soient pas autorisés. Cette représentation multi-état est présentée dans la Fig. 4.5 pour le même exemple.

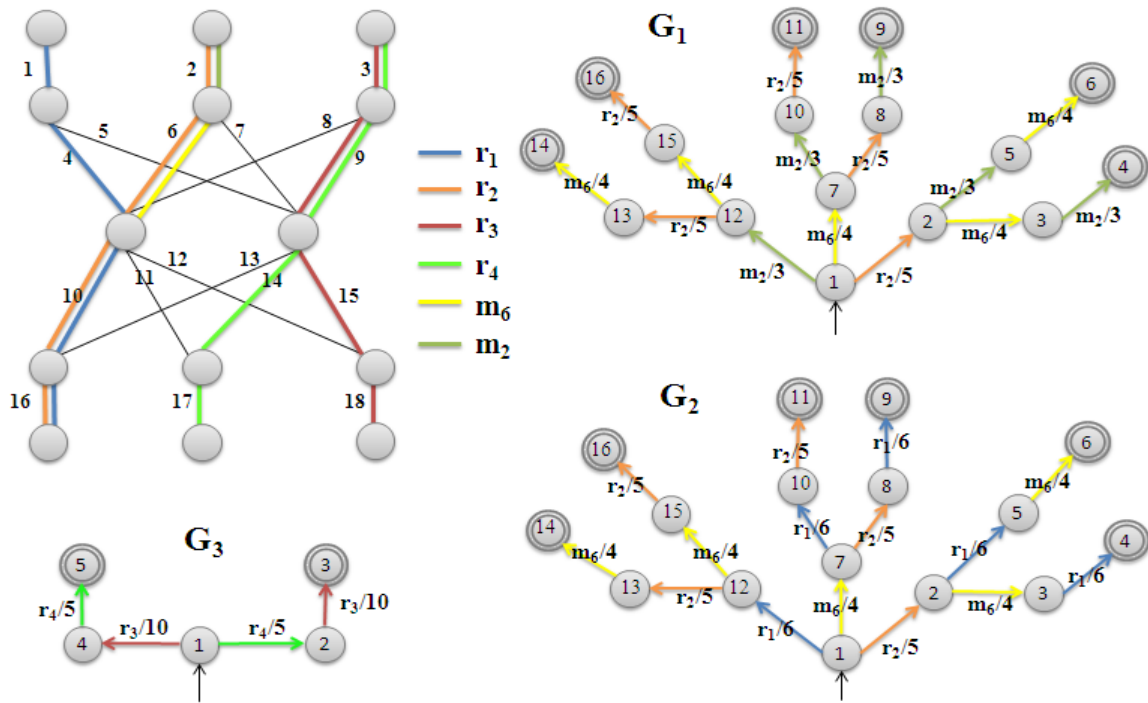


FIGURE 4.5: Exemple de modélisation de conflit par automates tropicaux : représentation multi-état pour restreindre l'exécution de chaque opération à une seule occurrence.

Naturellement, l'espace d'états du modèle augmente ; néanmoins, un comportement plus spécifique est représenté⁴. Pour d'autres applications où chaque opération serait exécutée plusieurs fois, le modèle mono-état peut contenir l'explosion combinatoire. Les propos de cette recherche se focalisent sur les relations d'exclusion mutuelle ; ainsi, le modèle mono-état donnera les mêmes résultats et il sera conservé du fait de sa simplicité.

4.3 Bases Théoriques pour l'Alphabet Proposé

Le développement théorique qui suit caractérise à partir des dateurs une sous-classe d'automates pour laquelle chaque événement possède une durée unique.

4. i.e. pour tous les automates locaux, à partir de l'état initial unique, une seule trajectoire sera possible telle qu'elle conduit vers un seul état final et toutes les opérations seront exécutées 1 fois.

Hypothèse 4.2. n opérations en conflit sont modélisées comme n événements mutuellement exclusifs dans un automate (i.e. n arcs sortants de l'état initial).

Définition 4.1. r_i^1 dénote un *micro-événement* dérivé d'un événement original r_i si la durée de r_i^1 est d'une unité de temps ; i.e. $y(r_i^1) = 1$.

La définition 4.1 vise à décomposer chaque opération originale de durée t_i en plusieurs opérations atomiques modélisant le passage du temps, chacune avec une durée d'une unité de temps, i.e. chaque micro-événement r_i^1 représente un tic d'horloge lié à l'opération r_i .

Définition 4.2. Une fois qu'une opération est lancée, les micro-événements subséquents pour la même opération sont exécutés de manière consécutive. Cette séquence de micro-événements s'achève quand la durée t_i de l'opération est terminée.

La définition 4.2 définit le comportement des séquences de micro-événements en cohérence avec l'hypothèse 4.1.

La Fig. 4.6 illustre un exemple simple relatif au cas d'étude, et la Fig. 4.7 présente sa décomposition en micro-événements. Sur la Fig. 4.6, les alignements pour 3 opérations de transfert sont représentées en couleur (spécifiant seulement les vannes à ouvrir). Dans cette configuration, une opération de maintenance doit aussi être effectuée sur la vanne 2. Dans cet exemple, deux conflits surgissent : le premier entre les opérations r_2 et m_2 , et le deuxième entre les opérations r_1 et r_2 . L'automate G_2 modélise le premier conflit, lorsque l'automate G_3 modélise le deuxième. D'autre part, l'opération r_3 est complètement indépendante et peut être exécutée simultanément avec toute autre opération ; ce qui est modélisé par l'automate G_1 .

Globalement, G_1 , G_2 , et G_3 modélisent un système dans lequel r_3 peut être exécutée simultanément avec r_1 ou r_2 ou m_2 . Cependant, dû aux conflits, r_1 et r_2 ne peuvent pas être exécutés en parallèle, ainsi que r_2 et m_2 . La Fig. 4.7 précise la décomposition en micro-événements ; notamment en une séquence de tics d'horloge décrivant le temps passé dans le cadre d'un comportement logique d'exclusion mutuelle. Naturellement, le nombre de micro-événements pour chaque opération est égal à la durée originale t_i de l'opération dans la Fig. 4.6, comme dans la définition 4.2. La décomposition montre une certaine flexibilité de l'exécution d'opérations simultanées, i.e. une opération a peut s'exécuter en parallèle avec une opération b pour une durée déterminée, et puis peut continuer à s'exécuter en parallèle avec une opération c sollicitant les ressources libérées par b .

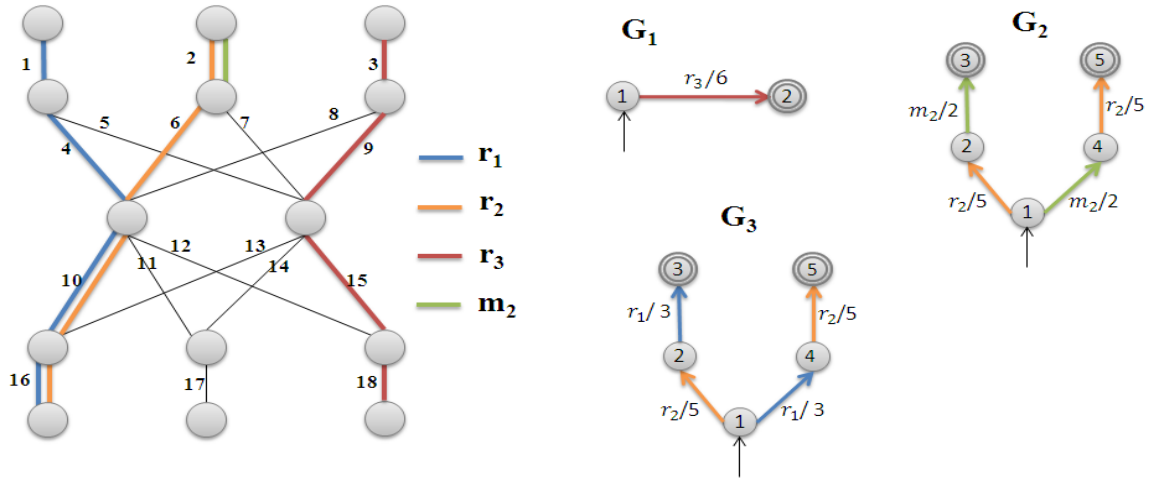


FIGURE 4.6: Exemple pour 2 conflits de partage de ressources.

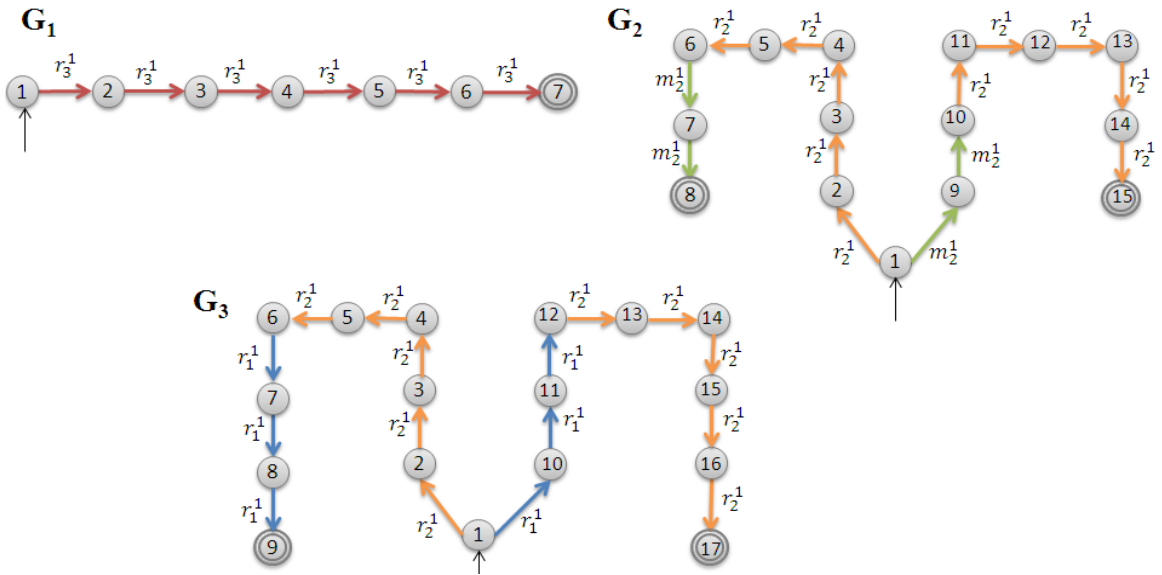


FIGURE 4.7: Décomposition en micro-événements pour l'exemple dans la Fig. 4.6.

Définition 4.3. Un *macro-événement* r_i^j représente une séquence de j micro-événements, i.e. $r_i^j = r_i^1 \dots r_i^j$, pour un événement originalement dénoté comme r_i . Le dateur d'un macro-événement est $y(r_i^j) = j$, et décrit l'exécution de r_i pendant j unités de temps, i.e. $y(r_i) \geq y(r_i^j)$.

Pour l'exemple de la Fig. 4.6, r_3 a une durée de 6 unités de temps ($y(r_3) = 6$), ce qui est équivalent à un macro-événement dénoté par r_3^6 , ou à des séquences de macro-événements telles que $r_3^2 r_3^4$ ou $r_3^5 r_3^1$ (où chaque macro-événement est une séquence de micro-événements). Cette flexibilité majeur pour la représentation des phénomènes de parallélisme entre des tâches de durée différente sera d'une importance capitale pour la proposition du nouvel alphabet.

Hypothèse 4.3. Le nouveau produit synchrone vise à exploiter les phénomènes de parallélisme au plus tôt ; i.e. parmi les possibilités d'exécution des opérations, seront retenues celles avec un nombre maximum d'opérations en parallèle.

L'hypothèse 4.3 vise à réduire le spectre des solutions d'ordonnancements.

Définition 4.4. Un *micro-événement composé* dénoté $(r_1^1, r_2^1, \dots, r_{(n-1)}^1, r_n^1)$, représente l'exécution simultanée de n micro-événements dans n automates ; son dateur est d'une unité de temps.

Définition 4.5. Un *macro-événement composé* dénoté $(r_1^j, r_2^j, \dots, r_{(n-1)}^j, r_n^j)$, ou de manière équivalente $(r_1^1, r_2^1, \dots, r_{(n-1)}^1, r_n^1)^j$, représente l'exécution simultanée de n macro-événements sur n automates, tous de durée de j unités de temps.

Définition 4.6. La cardinalité d'un *macro-événement composé* est le nombre de composantes, elle est définie par le nombre d'automates locaux (sous-systèmes) dans le modèle.

Définition 4.7. e^1 dénote le *micro-événement vide*, i.e. $y(e^1) = 1$, et e^n le *macro-événement vide*, i.e. $y(e^n) = n$.

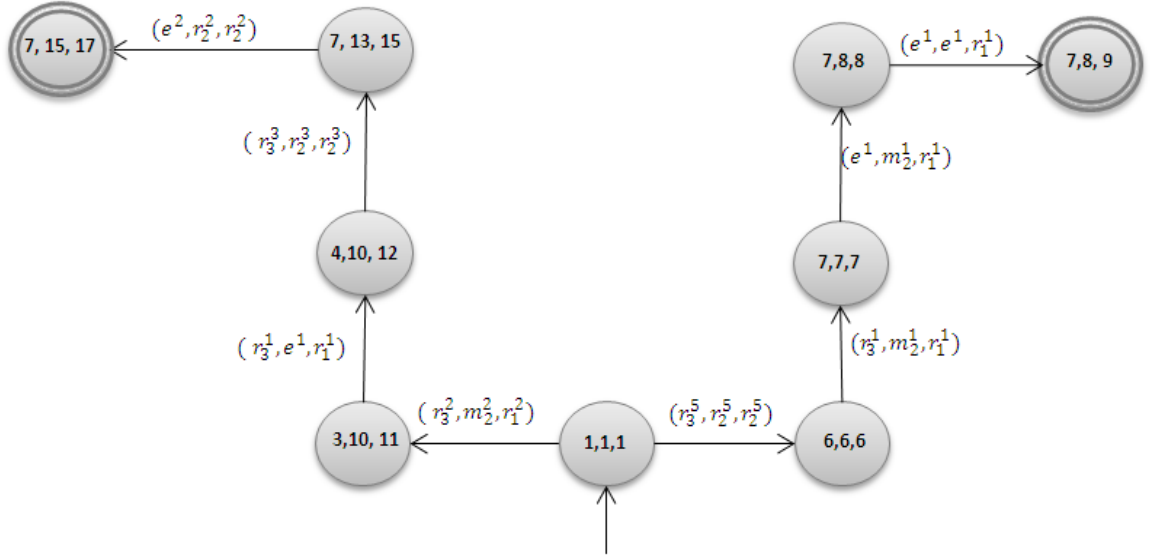
La définition 4.6 implique que pour un système à 3 automates, par exemple, chaque macro-événement doit avoir 3 composantes, e.g. (r_1^4, r_5^4, r_8^4) représenterait l'exécution de r_1 dans l'automate G_1 , r_5 dans G_2 et r_8 dans G_3 , toutes de durée de 4 unités de temps. La définition 4.7 permet de conserver cette notation même si aucun événement ne s'exécute sur un ou plusieurs automates, e.g. (e^4, r_5^4, r_8^4) ou (e^4, e^4, r_8^4) .

Dans la Fig. 4.8, les 2 trajectoires possibles pour exécuter au plus tôt les 4 opérations de l'exemple de la Fig. 4.7 sont représentées avec une exploitation maximale du parallélisme dès l'état initial. Chaque état étiqueté par ' i, j, k ' représente la phase d'exécution⁵ de l'état ' i ' dans G_1 , l'état ' j ' dans G_2 , et l'état ' k ' dans G_3 . La première trajectoire de l'état ' $1, 1, 1$ ' vers l'état ' $7, 8, 9$ ' commence par l'exécution de r_2 , de manière synchrone dans G_2 et dans G_3 , lorsque r_3 s'exécute en parallèle dans G_1 .

La deuxième trajectoire, de l'état ' $1, 1, 1$ ' vers ' $7, 15, 17$ ', exécute r_1 avant r_2 . Les deux trajectoires modélisent la possibilité d'exécuter le maximum d'opérations simultanément et au plus tôt. Par exemple, la transition de l'état ' $1, 1, 1$ ' vers l'état ' $6, 6, 6$ ' est étiquetée par le macro-événement composé (r_3^5, r_2^5, r_1^5) qui modélise l'exécution simultanée d'opérations r_3 , et r_2 pendant 5 unités de temps. (r_3^1, m_2^1, r_1^1) représente l'exécution de r_3 lors de sa dernière unité de temps, en même temps que m_2 est exécuté dans G_2 et r_1 dans G_3 .

Le produit dans la Fig. 4.7 est déterministe ; i.e. l'automate possède un seul état initial et de chaque état un même événement ne peut générer des transitions vers des états différents.

5. positionnement sur les états des automates locaux pour définir un comportement global.


 FIGURE 4.8: Représentation du produit synchrone $G_1||G_2||G_3$ pour l'exemple de la Fig. 4.7.

Pour des propos d'illustration, avec l'approche proposée dans [44] pour un produit d'automates $(\max, +)$, et en conservant leur terminologie et notation, l'alphabet obtenu pour l'exemple de la Fig. 4.6 inclurait l'événement composé (r_2, r_3) modélisant l'exécution simultanée de r_2 et r_3 avec une date d'achèvement de $y(r_2) \oplus y(r_3) = 5 \oplus 6 = 6$. Dans la Fig. 4.8, et selon le nouveau produit synchrone, r_2 et r_3 sont exécutées simultanément et dès que les ressources associées à r_2 sont libérées (i.e. à $t = 5$), m_2 et r_1 sont exécutées à $t = 5$ en parallèle avec la fin d'exécution de r_3 .

Définition 4.8. Le *dateur d'un macro-événement composé* est défini comme la somme des composants des dateurs, i.e. $y(r_u^j, r_v^j) = y(r_u^j) \oplus y(r_v^j) = j \oplus j = j$.

Ceci est équivalent à la définition de dateur des nouveaux événements générés pour le produit d'automates $(\max, +)$ dans [44]. Néanmoins ici, étant donné que par définition les composants seront de durée identique, la propriété d'idempotence de l'opérateur $\oplus$ (i.e. $a \oplus a = a$) conduit à ce que le dateur d'un macro-événement composé soit égal aux dateurs de ses composants ; par exemple : $y(r_1^3, r_3^3) = y(r_1^3) \oplus y(r_3^3) = 3 \oplus 3 = 3$. Ainsi, la théorie proposée est applicable également pour les automates $(\min, +)$ et $(\max, +)$ puisqu'ils sont basés sur des algèbres avec l'opérateur $\oplus$ idempotent.

Définition 4.9. La *concaténation de macro-événements composés*, tous de cardinalité n , est effectuée par composants. Le résultat est un macro-événement de même cardinalité n .

Afin illustrer le principe de la définition 4.9, reprenant l'exemple de la Fig. 4.7, le résultat de la concaténation de $a = (r_3^5, r_2^5, r_2^5)$, $b = (r_3^1, m_2^1, r_1^1)$, et $c = (e^1, m_2^1, r_1^1)$ serait $abc = (r_3^5 r_3^1 e^1,$

$r_2^5 m_2^1 m_2^1, r_2^5 r_1^1 r_1^1$) qui est synthétisé comme : $abc = (r_3^6 e^1, r_2^5 m_2^2, r_2^5 r_1^2)$. Évidemment, l'alphabet est associatif puisque $a(bc) = (ab)c$.

Dans ce qui suit, l'alphabet de chaque automate local est décrit selon 2 types d'événements : événements *privés*, et événements de *synchronisation* ou communs. Les événements privés pour un automate G_i sont ceux qui ne surviennent que dans cet automate. Alors que les événements de synchronisation sont communs à plusieurs automates locaux. Ces notions sont formalisées par des définitions illustrées par des exemples.

Définition 4.10. Un *dateur privé* pour un automate G_i est défini comme le produit de tous les dateurs concernant les événements privés dans cet automate ; i.e. $y_{G_{ip}} = \bigotimes_{a \in A_{ip}} y(a)$. Le dateur privé est dénoté par $y_{G_{ip}}$ et A_{ip} dénote le sous-ensemble de l'alphabet de G_i qui ne contient que des événements privés.

Pour le système de la Fig. 4.6 (respectivement Fig. 4.7), les dateurs privés identifiés sont : $y_{G_{1p}} = y(r_3) = 6$, $y_{G_{2p}} = y(m_2) = 2$, et $y_{G_{3p}} = y(r_1) = 3$.

Définition 4.11. Le *dateur de synchronisation* pour un automate G_i est défini comme le produit des dateurs des événements communs de l'automate G_i avec tout autre automate.

Pour un système à 3 automates locaux, les dateurs de synchronisation possibles sont les combinaisons : $y_{s(G_i, G_j)} = \bigotimes_{a \in A_i \cap A_j} y(a)$, $y_{s(G_i, G_k)} = \bigotimes_{b \in A_i \cap A_k} y(b)$, $y_{s(G_j, G_k)} = \bigotimes_{c \in A_j \cap A_k} y(c)$, $y_{s(G_i, G_j, G_k)} = \bigotimes_{d \in A_i \cap A_j \cap A_k} y(d)$. En outre, $s(G_u, G_v)$ dénote l'ensemble des événements synchronisant 2 automates : G_u et G_v , et $s(G_u, G_v, G_w)$ l'ensemble d'événements synchronisant 3 automates : G_u , G_v , et G_w .

Pour le système représenté dans la Fig. 4.4, le seul dateur synchronisant non nul est : $y_{s(G_1, G_2)} = y(r_2) \otimes y(m_6) = 4 \otimes 5 = 9$, alors que pour le système de la Fig. 4.6 le dateur synchronisant correspond à $y_{s(G_2, G_3)} = y(r_2) = 5$.

Hypothèse 4.4. Étant donné qu'à part de l'exclusion mutuelle aucune autre contrainte n'est prise en compte pour l'ordonnancement d'opérations, il est considéré que lorsqu'un automate comporte plusieurs événements privés, ils sont tous exécutés de manière consécutive et sans ordre d'exécution imposé, i.e. toutes les permutations sont acceptables. Le même principe est valable pour le cas d'un système avec plusieurs événements de synchronisation. Ceci réduit de manière significative le spectre de solutions pour l'alphabet.

Définition 4.12. Un *dateur global* pour un automate G_i est défini comme le produit de l'ensemble des dateurs pour des événements appartenant à l'alphabet ; i.e. $y_{G_i} = \bigotimes_{a \in A_i} y(a)$. Le dateur global regroupe dateurs privées et de synchronisation et formellement s'exprime par :

$y_{G_i} = y_{G_{ip}} \otimes y_{s(G_i, G_j)} \otimes y_{s(G_j, G_k)} \otimes y_{s(G_i, G_k)} \otimes y_{s(G_i, G_j, G_k)}$, pour le cas de 3 automates, par exemple.

Pour l'exemple de la Fig. 4.4, les dateurs globaux sont : $y_{G_1} = 12$, $y_{G_2} = 15$ et $y_{G_3} = 15$, lorsque pour la Fig. 4.6, $y_{G_1} = 6$, $y_{G_2} = 7$, et $y_{G_3} = 8$.

Définition 4.13. Dans un système modélisé par n automates, chaque automate est indexé et dénoté par G_i pour $i = 1, \dots, n$, en rapport aux *dateurs globaux* selon la règle de classement décrite par (4.1).

Cette règle de classement a été appliquée à l'indexation des exemples présentés. Le but est d'envisager les meilleures ordonnancements qualitatifs tels que toutes les opérations soient exécutées au plus tôt.

$$y_{G_n} \geq y_{G_{n-1}} \geq \dots \geq y_{G_2} \geq y_{G_1} \quad (4.1)$$

Définition 4.14. A_i^1 est le *micro-alphabet* d'un alphabet originalement dénoté comme A_i pour un automate G_i . Il est formé de tous les micro-événements associés aux événements de l'alphabet original. Par exemple, dans la Fig. 4.6 pour l'automate G_2 , $A_2 = \{r_2, m_2\}$ et $A_2^1 = \{r_2^1, m_2^1\}$.

4.4 Alphabet du Nouveau Produit d'Automates Tropicaux

Dans ce qui suit, le terme 'makespan' est employé, en analogie avec les systèmes de production, pour faire référence au temps d'exécution d'un ensemble d'opérations.

Dans cette section, l'alphabet pour le nouveau produit synchrone d'automates de 3 ou plus automates tropicaux ($n \geq 3$) est défini⁶.

Par l'approche de [44], pour le cas d'un système à 2 automates locaux, notamment G_1 et G_2 de la Fig. 4.4 (et selon leur notation) l'alphabet résultant du produit $G_1 || G_2$ serait : $A = \{r_2, m_6, (m_2, r_1)\}$, selon (4.2). Ainsi, lorsqu'un événement commun aux deux automates (comme r_2 ou m_6) est exécuté, il est exécuté dans les deux automates, i.e. les automates sont synchronisés. Sinon, les événements privés sont exécutés simultanément (i.e. m_2 avec r_1) et la date d'achèvement de cette exécution simultanée est le maximum des durées des événements privés, dans ce cas 6 unités de temps. Dans tout scénario d'exécution, e.g. $r_2 m_6 (m_2, r_1)$ ou $(m_2, r_1) r_2 m_6$, la date d'achèvement de la séquence est identique, i.e. 15 unités de temps, si chaque événement est exécuté une seule fois. Comme l'automate G_2 nécessite au moins 15 unités de temps pour exécuter toutes les opérations au moins une fois, ce résultat ne pourrait jamais

6. Les résultats pour $n = 2$ sont considérés triviaux et sont en fait équivalents aux travaux de [44] puisque le parallélisme peut être exploité en générant un makespan optimal ; alors que pour $n = 3$ l'optimum souhaité ne peut être atteint avec l'approche [44].

être amélioré, i.e. aucun intervalle oisif n'apparaît dans l'ordonnancement. Au delà de deux automates, des conflits d'ordre supérieur sont à considérer (conflits entre les différentes paires, tierces, plusieurs tierces et autres).

$$A = (A_1 \cap A_2) \cup [(A_1 \setminus A_2)^* \times (A_2 \setminus A_1)^*] \quad (4.2)$$

Les résultats dans [44] (pour le produit d'automates $(\max, +)$) ont inspiré l'approche présentée ici où les dates d'achèvement de tous les événements seront minimisées. Par conséquent, [44] est une référence dans les développements qui suivent.

Dans [44], l'alphabet est proposé pour un produit avec $n = 3$ dans lequel les opérations en parallèle possèdent la date d'achèvement de l'opération la plus longue. Cet alphabet est décrit par (4.3). La notation est telle que $A_i \setminus (A_j \cup A_k)$ dénote les éléments de A_i sauf ceux qui sont communs à A_j et/ou A_k . Par conséquent, $A_i \setminus (A_j \cup A_k)$ dénote les événements privés pour l'automate G_i (i.e. événements qui ne surviennent que dans cet automate). Le produit cartésien ' $\times$ ' est utilisé pour créer de nouveaux événements composés, basés sur les alphabets originaux ; par exemple, si $A_i = \{a, b\}$, et $A_j = \{c\}$, alors $A_i \times A_j = \{(a, c), (b, c)\}$.

$$A = (A_1 \cap A_2 \cap A_3) \cup \bigcup_{i \neq j \neq k} [(A_i \cap A_j) \times (A_k \setminus (A_i \cup A_j))^*] \cup (A_1 \setminus (A_2 \cup A_3))^* \times (A_2 \setminus (A_1 \cup A_3))^* \times (A_3 \setminus (A_1 \cup A_2))^* \quad (4.3)$$

Dans (4.3), le premier terme de l'alphabet dénote les événements communs aux 3 automates. Le deuxième terme dénote un événement composé, formé par des événements communs aux 2 automates G_i et G_j et un événement (ou séquence) privé du troisième automate (G_k) s'exécutant en parallèle. Finalement, le dernier terme représente un événement composé formé par 3 éléments, chacun dénotant des événements (ou séquences) privés, s'exécutant dans chaque automate simultanément. Avec la formulation (4.3), les événements communs aux 3 automates s'exécutent en même temps car il s'agit d'événements de synchronisation et leurs dateurs sont égaux. Les événements communs aux 2 automates sont exécutés de manière synchronisée et, en même temps, ils peuvent s'exécuter en parallèle avec des événements (ou séquences) privés de l'automate restant. Dans ce cas, le dateur du nouvel événement composé est le maximum des 2 dateurs (de synchronisation et privé). Le même principe est appliqué lorsque des événements (ou séquences) privés s'exécutent simultanément dans les 3 automates.

Le nouvel alphabet proposé ici, décrit par (4.4) vise à garantir un makespan minimal. Le premier terme : $(A_1^1 \cap A_2^1 \cap A_3^1)$ dénote tous les micro-événements composés synchronisant les 3 automates. $A_{i,j,k}^1$ représente tous les micro-événements composés synchronisant G_i et G_j et s'exécutant simultanément avec des micro-événements privés dans G_k . Finalement, $A_{1,2,3}^1$ dénote tous les micro-événements composés issues de l'exécution simultanée de micro-événements privés dans les 3 automates. La distinction fondamentale entre l'équation (4.4) par rapport à l'équation (4.3)

est de disposer de différents scénarios de parallélisme entre les opérations par une décomposition en tics d'horloge dans les différents automates. L'exposant 1 rappelle la durée d'une unité de temps affectée à chacun des micro-événements.

Le terme $(A_1^1 \cap A_2^1 \cap A_3^1)$ est défini comme l'intersection de tous les micro-alphabets. Alors que, les termes $A_{ij,k}^1$ et $A_{1,2,3}^1$ sont déterminés par une procédure plus complexe impliquant l'analyse des exécutions possibles de tous les types d'événements (i.e. de synchronisation ou privés) pour que la meilleure configuration soit choisie, telle que le temps d'exécution soit minimisé. Les résultats varient ainsi selon les scénarios de synchronisation retenus ; par exemple : absence de synchronisation, synchronisation entre une paire d'automates, 2 paires, 3 paires, une tierce, et des combinaisons de synchronisation de la tierce avec les scénarios précédents.

$$A^1 = (A_1^1 \cap A_2^1 \cap A_3^1) \cup A_{12,3}^1 \cup A_{13,2}^1 \cup A_{23,1}^1 \cup A_{1,2,3}^1 \quad (4.4)$$

Dans ce qui suit, le nouvel alphabet est proposé pour les différents scénarios de synchronisation (e.g. $s_1, s_2, s_3, s_4, s_5, s_i$) pour un système à 3 automates. Pour l'obtention de résultats optimaux (en makespan), l'exécution d'événements de synchronisation en parallèle avec des événements privés est priorisée.

4.4.1 Alphabet pour une Paire d'Automates Synchronisés dans un Système à 3 Automates ($n = 3$)

Tout d'abord, la synchronisation de la paire (G_2, G_3) est étudiée en détail et ensuite le comportement pour la synchronisation des autres paires est représenté de manière analogue. Sachant que l'analyse reste la même pour les 3 paires possibles, la paire (G_2, G_3) est prise aléatoirement pour une explication détaillée.

Selon la définition 4.13, les indices des modèles d'automate sont choisis tels que $y_{G_3} \geq y_{G_2} \geq y_{G_1}$, et ainsi les résultats (4.5) définissent la relation des dateurs lorsque les automates G_2 et G_3 sont synchronisés.

$$y_{G_{3p}} \otimes y_{s(G_2, G_3)} \geq y_{G_{2p}} \otimes y_{s(G_2, G_3)} \geq y_{G_{1p}} \quad (4.5)$$

Cette relation entre dateurs, apporte des informations significatives pour un ordonnancement d'opérations optimisant le makespan dans le produit synchrone et, par conséquent, sera la base de la définition des éléments de l'alphabet correspondant.

La première inégalité de (4.5) fournit (4.6) (le dateur privé pour G_3 sera égal ou supérieur au dateur privé pour G_2). La deuxième inégalité fournit (4.7) et indique que $y_{G_{2p}} \otimes y_{s(G_2, G_3)} \geq y_{G_{1p}}$, conduisant à 2 possibilités : $y_{s(G_2, G_3)} \geq y_{G_{1p}}$ ou $y_{s(G_2, G_3)} < y_{G_{1p}}$. L'intérêt dans cette relation réside sur le fait que $s(G_2, G_3)$ et G_{1p} sont des ensembles d'opérations qui peuvent être

exécutés de manière simultanée ; par conséquent, ces relations de dateurs définissent la base de la détermination du nouvel alphabet. Finalement, de (4.5) se déduit (4.8) par transitivité (i.e. si $y_{G_3} \geq y_{G_2}$ et $y_{G_2} \geq y_{G_1}$, alors $y_{G_3} \geq y_{G_1}$).

Il peut être aussi déduit à partir de l'équation (4.5) que tous les dateurs privés seront non nuls (ceci est vrai pour G_1 puisque sinon cet automate n'existerait pas car il ne comporte que d'événements privés, et aussi pour G_2 et G_3 puisque les événements privés sont ceux qui permettent de les différencier⁷).

$$y_{G_{3p}} \geq y_{G_{2p}} \quad (4.6)$$

$$y_{G_{2p}} \otimes y_{s(G_2, G_3)} \geq y_{G_{1p}} \cdot \cdot y_{s(G_2, G_3)} \geq y_{G_{1p}} \vee y_{s(G_2, G_3)} < y_{G_{1p}} \quad (4.7)$$

$$y_{G_{3p}} \otimes y_{s(G_2, G_3)} \geq y_{G_{1p}} \quad (4.8)$$

Les résultats présentés en Fig. 4.9 reprennent (4.5) et l'hypothèse 4.3. L'ordonnancement de la Fig. 4.9a) peut être considéré comme l'ordonnancement qualitatif optimal pour l'exécution de toutes les opérations au plus tôt. Il est appelé qualitatif car la date d'exécution de chaque opération n'est pas spécifiée mais plutôt les dates d'exécution des blocs d'opérations (e.g. bloc des événements privés à G_1 en parallèle avec le bloc des événements communs à G_2 et G_3), et toute permutation d'opérations dans chaque bloc est possible. Cet ordonnancement est intuitif du fait de la simplicité du scénario de synchronisation considéré. Il est construit sur les bases des relations de dateurs, permettant d'identifier les événements composés possibles survenant dans le système pour que le makespan soit minimisé.

Afin d'interpréter les différents ordonnancements optimaux présentés ici, une logique d'intégration de blocs d'opérations doit être considérée, la Fig. 4.9 en présente un exemple intuitif. Dans la figure, chaque bloc grisé indique l'intervalle des dates d'achèvement possibles pour le bloc précédent.

Tout d'abord (Fig. 4.9a)), $s(G_2, G_3)$ est exécuté en parallèle avec G_{1p} et la date d'achèvement pour G_{1p} est cohérente avec (4.7), indiquant que G_{1p} peut s'achever avant, concomitamment, ou après $s(G_2, G_3)$ (d'où le bloc grisé supérieur). Ensuite, (4.6) révèle que G_{3p} s'achève concomitamment ou après G_{2p} . Finalement, (4.8) indique que la date d'achèvement des opérations dans G_3 doit être égale ou supérieure à G_{1p} .

Qualitativement l'objectif est de restreindre l'espace de solutions aux ordonnancements optimaux, en laissant la liberté de prendre en compte toute permutation possible d'opérations dans chaque bloc ou ensemble. L'analyse de la structure optimale pour l'ordonnancement permet de déterminer les possibles micro-événements composés dans le nouveau produit synchrone pour une exécution optimale de l'ensemble des opérations. Notamment, d'après la Fig. 4.9a), s'il est

7. Si un seul de ces automates possédait des événements privés, l'autre automate ne comporterait aucune information supplémentaire, et si aucun des deux n'avait d'événement privé, ils représenteraient le même automate.

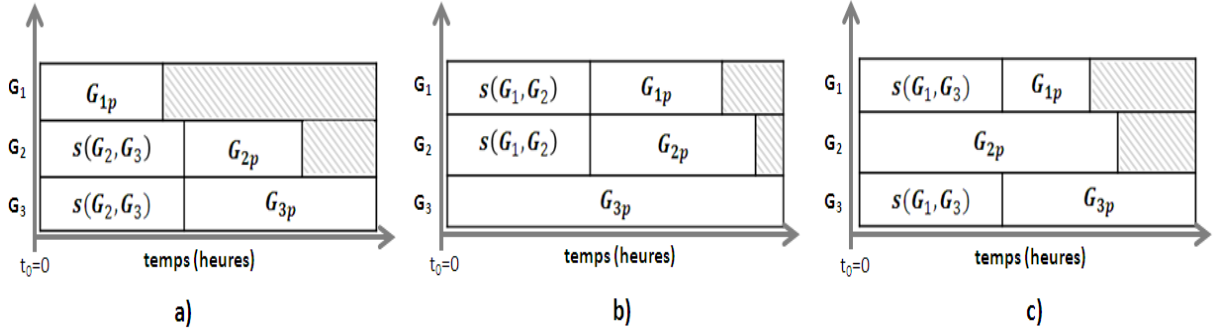


FIGURE 4.9: Ordonnancements qualitativement optimaux pour des systèmes avec une paire d'automates synchronisés : 3 cas possibles.

convenu dans le système que $y_{G_{1p}} < y_{s(G_2, G_3)}$ alors les micro-événements composés comportant des éléments de $s(G_2, G_3)$ seront de deux types : (a^1, b^1, b^1) et (e^1, b^1, b^1) , avec $a \in G_{1p}$ et $b \in s(G_2, G_3)$, puisqu'une fois le bloc G_{1p} achevé, $s(G_2, G_3)$ continuera son exécution. Pour ce scénario de synchronisation, une configuration non optimale serait d'exécuter au début tous les événements privés de tous les automates en parallèle. Le bloc privé de G_3 devrait s'achever avant que $s(G_2, G_3)$ ne soit exécuté, introduisant alors un intervalle d'oisiveté pour G_2 , alors que l'objectif est de libérer les ressources au plus tôt pour qu'elles puissent être exploitées ultérieurement.

L'avantage de la configuration proposée en Fig. 4.9a) est de prioriser $s(G_2, G_3)$ puisqu'il offre la possibilité d'inclure des événements privés en parallèle d'un troisième automate au plus tôt.

Dans le cas où la seule paire synchronisée est (G_1, G_2) , la relation fondamentale des dateurs correspond à (4.9), et si par contre (G_1, G_3) sont synchronisés, cela répond à la relation (4.10). Sur les bases de (4.9) et (4.10), et appliquant la même analyse que pour la paire (G_2, G_3) , les ordonnancements de référence apparaissent en Fig. 4.9b) et 4.9c). Ils permettent de déterminer les événements dans l'alphabet du nouveau produit synchrone en termes de conditions de dateurs.

L'analyse des 3 scénarios pour 1 paire d'automates synchronisés conduisent à la proposition de l'alphabet défini dans (4.4) où les éléments $A_{ij,k}^1$ et $A_{12,3}^1$ suivent les expressions (4.11-4.15).

$$y_{G_{3p}} \geq y_{G_{2p}} \otimes y_{s(G_1, G_2)} \geq y_{G_{1p}} \otimes y_{s(G_1, G_2)} \quad (4.9)$$

$$y_{G_{3p}} \otimes y_{s(G_1, G_3)} \geq y_{G_{2p}} \geq y_{G_{1p}} \otimes y_{s(G_1, G_3)} \quad (4.10)$$

$$A_{13,2}^1 = \begin{cases} (A_1^1 \cap A_3^1) \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_1^1 \cap A_3^1) & \text{si } y_{s(G_1, G_3)} \neq 0 \\ \emptyset & \text{sinon} \end{cases} \quad (4.11)$$

$$A_{12,3}^1 = \begin{cases} (A_1^1 \cap A_2^1) \times (A_1^1 \cap A_2^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1) & \text{si } y_{s(G_1, G_2)} \neq 0 \\ \emptyset & \text{sinon} \end{cases} \quad (4.12)$$

$$A_{23,1}^1 = \begin{cases} (A_1^1 \setminus A_2^1 \cup A_3^1) \times (A_2^1 \cap A_3^1) \times (A_2^1 \cap A_3^1) & \text{si } y_{s(G_2, G_3)} \neq 0 \wedge (y_{G_{1p}} \geq y_{s(G_2, G_3)}) \\ [(A_1^1 \setminus A_2^1 \cup A_3^1) \cup e^1] \times (A_2^1 \cap A_3^1) \times (A_2^1 \cap A_3^1) & \text{si } y_{s(G_2, G_3)} \neq 0 \wedge (y_{G_{1p}} < y_{s(G_2, G_3)}) \\ \emptyset & \text{si } y_{s(G_2, G_3)} = 0 \end{cases} \quad (4.13)$$

$$A_{1,2,3}^1 = \begin{cases} [((A_1^1 \setminus A_2^1 \cup A_3^1) \cup e^1) \times ((A_2^1 \setminus A_1^1 \cup A_3^1) \cup e^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1)] \setminus A_{e(1,2)} & \\ \quad \text{si } (y_{G_3} > y_{G_2}) \wedge (y_{G_{1p}} > y_{s(G_2, G_3)}) & \\ e^1 \times ((A_2^1 \setminus A_1^1 \cup A_3^1) \cup e^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1) & \text{si } (y_{G_3} > y_{G_2}) \wedge (y_{G_{1p}} \leq y_{s(G_2, G_3)}) \\ ((A_1^1 \setminus A_2^1 \cup A_3^1) \cup e^1) \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1) & \\ \quad \text{si } [(y_{G_3} = y_{G_2}) > y_{G_1}] \wedge (y_{G_{1p}} > y_{s(G_2, G_3)}) & \\ e^1 \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1) & \text{si } [(y_{G_3} = y_{G_2}) > y_{G_1}] \wedge (y_{G_{1p}} \leq y_{s(G_2, G_3)}) \\ (A_1^1 \setminus A_2^1 \cup A_3^1) \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1) & \text{si } y_{G_3} = y_{G_2} = y_{G_1} \end{cases} \quad (4.14)$$

$$A_{e(1,2)}^1 = \begin{cases} [e^1 \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1)] \cup [(A_1^1 \setminus A_2^1 \cup A_3^1) \times e^1 \times (A_3^1 \setminus A_1^1 \cup A_2^1)] & \\ \quad \text{si } y_{G_2} = y_{G_1} & \\ [(A_1^1 \setminus A_2^1 \cup A_3^1) \times e^1 \times (A_3^1 \setminus A_1^1 \cup A_2^1)] & \text{si } y_{G_2} > y_{G_1} \end{cases} \quad (4.15)$$

Le terme $A_{23,1}^1$ existe si et seulement si G_2 et G_3 possèdent des événements communs, ce qui répond à (4.13). Dans ce cas, alors 2 scénarios sont possibles selon la valeur du dateur privé de G_1 : soit $y_{G_{1p}} \geq y_{s(G_2, G_3)}$ ou $y_{G_{1p}} < y_{s(G_2, G_3)}$. Ce résultat dérive de l'analyse des dateurs (4.5-4.8) et les impacts temporels sont identifiées dans la Fig. 4.9a). Dans le premier cas ($y_{G_{1p}} \geq y_{s(G_2, G_3)}$), les micro-événements de $s(G_2, G_3)$ s'exécuteront forcément avec des événements privés de G_1 (première partie de l'équation (4.13)). Dans le deuxième cas ($y_{G_{1p}} < y_{s(G_2, G_3)}$ et deuxième partie de (4.13)), un micro-événement vide est ajouté au premier composant puisque G_{1p} s'achèvera avant la terminaison de $s(G_2, G_3)$. La même analyse est valable pour (4.11) et (4.12) dans les scénarios où la paire d'automates synchronisés n'est pas (G_2, G_3) .

Finalement, (4.14-4.15) développent le terme $A_{1,2,3}^1$ pour les 3 possibilités de synchronisation d'une paire d'automates. Les formulations sont plus denses pour $A_{1,2,3}^1$ puisque plusieurs possibilités sont envisageables pour composer les micro-événements privés. L'objectif est toujours de restreindre l'alphabet pour qu'aucune trajectoire n'introduise des intervalles oisifs dans l'ordonnancement. L'analyse effectuée est alors la même que dans (4.11-4.13).

Suivant cette procédure, l'alphabet peut être construit pour 2 ou 3 paires d'automates synchronisés. Dans ce qui suit, les formulations pour le cas le plus complexe (i.e. 3 paires d'automates synchronisés) sont présentées.

4.4.2 Alphabet pour Trois Paires d'Automates en Synchronisation dans un Système à 3 Automates ($n = 3$)

La règle principale de classement des dateurs globaux répond à l'expression (4.16) pour le scénario de synchronisation de 3 paires d'automates. Elle correspond à la substitution des dateurs dans l'expression générique présentée dans la définition 4.13.

$$y_{G_{3p}} \otimes y_{s(G_2, G_3)} \otimes y_{s(G_1, G_3)} \geq y_{G_{2p}} \otimes y_{s(G_2, G_3)} \otimes y_{s(G_1, G_2)} \geq y_{G_{1p}} \otimes y_{s(G_1, G_2)} \otimes y_{s(G_1, G_3)} \quad (4.16)$$

La démarche correspond encore à se référer un ordonnancement potentiel pour en extraire l'alphabet. Dans la Fig. 4.10, les 2 ordonnancements qualitatifs optimaux sont issus de l'analyse des scénarios possibles d'exécution des blocs d'opérations. La proposition est de débiter par l'exécution de G_2 et G_3 , en découlent plusieurs scénarios selon le dateur privé de G_1 : soit G_{1p} s'achève avant, concomitamment ou après $s(G_2, G_3)$.

Scénario 1 : L'achèvement de G_{1p} concomitamment ou après⁸ $s(G_2, G_3)$, voir Fig. 4.10a), i.e. $y_{G_{1p}} \geq y_{s(G_2, G_3)}$.

Scénario 2 : L'achèvement G_{1p} avant $s(G_2, G_3)$, voir Fig. 4.10b), i.e. $y_{G_{1p}} < y_{s(G_2, G_3)}$.

Pour le scénario 1, G_{1p} s'achève concomitamment ou après $s(G_2, G_3)$, alors de la deuxième inégalité de (4.16) il est déduit (en algèbre conventionnelle pour faciliter la compréhension) : $y_{G_{2p}} + y_{s(G_2, G_3)} \geq y_{G_{1p}} + y_{s(G_1, G_3)}$. Par conséquent, si $y_{G_{1p}} \geq y_{s(G_2, G_3)}$ alors $y_{G_{2p}} \geq y_{s(G_1, G_3)}$. De la première inégalité de (4.16) il est obtenu : $y_{G_{3p}} + y_{s(G_1, G_3)} \geq y_{G_{2p}} + y_{s(G_1, G_2)}$; par conséquent, si $y_{G_{2p}} \geq y_{s(G_1, G_3)}$ alors $y_{G_{3p}} \geq y_{s(G_1, G_2)}$. Cela indique que l'ensemble d'événements privés G_{ip} dans chaque automate s'achève en même temps ou après l'ensemble d'événements de synchronisation

8. Les deux scénarios $y_{G_{1p}} = y_{s(G_2, G_3)}$ et $y_{G_{1p}} > y_{s(G_2, G_3)}$ génèrent les mêmes éléments dans l'alphabet dans le produit lors de l'exécution de $s(G_2, G_3)$. Notamment, des éléments du type (a^1, b^1, b^1) où $a^1 \in G_{1p}$, et $b^1 \in s(G_2, G_3)$.

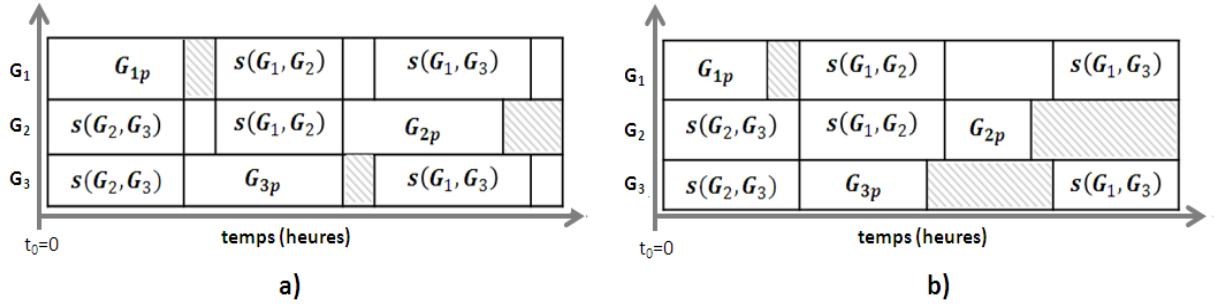


FIGURE 4.10: Ordonnements Qualitatifs Optimaux pour des Systèmes à 3 Automates et avec 3 Paires d'Automates en Synchronisation.

des paires $s(G_j, G_k)$; l'intérêt est qu'il s'agit des blocs d'opérations qui peuvent être exécutés en parallèle.

La première inégalité est équivalente à $y_{G_{3p}} - y_{s(G_1, G_2)} \geq y_{G_{2p}} - y_{s(G_1, G_3)}$, indiquant que G_{3p} dépasse $y_{s(G_1, G_2)}$ dans une mesure égale ou supérieure à celle dans laquelle $y_{G_{2p}}$ dépasse $y_{s(G_1, G_3)}$. Cela suggère intuitivement que G_{3p} doit être exécuté au plus tôt, afin de minimiser le temps d'exécution global. Par conséquent, dans la Fig. 4.10a), il est proposé d'exécuter G_{3p} aussi tôt que possible (i.e. lorsque $s(G_2, G_3)$ s'achève) pour que le parallélisme puisse être exploité tout d'abord avec G_{1p} , suivi par $s(G_1, G_2)$, et finalement de G_{2p} . Dans la Fig. 4.10a), certains intervalles d'oisiveté sont inévitables telles que le bloc suivant l'exécution de $s(G_2, G_3)$, introduit par l'achèvement de G_{1p} (si achèvement après $s(G_2, G_3)$) qui conditionne le début d'exécution de $s(G_1, G_2)$ dans les deux automates G_1 et G_2 . Le même principe est appliqué pour le bloc suivant l'exécution de $s(G_1, G_2)$ dans G_1 .

Pour le scénario 2, comme les événements de G_1 s'achèvent avant $s(G_2, G_3)$ (voir Fig. 4.10b)), la séquence optimale proposée est d'exécuter immédiatement $s(G_1, G_2)$ (avec G_{3p} simultanément) et ensuite exécuter $s(G_1, G_3)$ une fois que G_{3p} s'achève. G_{3p} conditionne le début d'exécution de $s(G_1, G_3)$ dans les automates G_1 et G_3 , ce qui peut introduire l'intervalle inévitable d'oisiveté dans G_1 suivant l'exécution de $s(G_1, G_2)$.

De manière analogue à la section précédente, le comportement des dateurs est examiné et formulé dans les expressions (4.17-4.21). Dans (4.17), la première partie de la fonction établit que si le scénario 1 est considéré, les seuls micro-événements générés dans $A_{23,1}^1$ (à être substitués dans (4.4)) seront de la forme (a^1, b^1, b^1) . Par contre, si le scénario 2 se déroule, alors les micro-événements générés peuvent être des types (a^1, b^1, b^1) et (e^1, b^1, b^1) , puisque (Fig. 4.10b)) G_{1p} s'achève avant $s(G_2, G_3)$, introduisant un laps de temps inévitable.

Les conditions C_i , pour $i = 1, \dots, 8$, visent à simplifier la notation dans les équations (4.18) et (4.19), et décrivent des relations générales des dateurs (e.g. le dateur privé de G_i est inférieur au dateur privé de G_j et en même temps les dateurs globaux sont égaux). Les conditions C_{i-jk} dénotent la valeur dans laquelle le dateur privé d'un automate dépasse le dateur de synchronisation

de l'automate G_j et G_k sachant que les événements respectifs (privés de G_i et de synchronisation de G_j et G_k) sont exécutés simultanément.

L'analyse de (4.17) est identique pour l'ensemble des équations (4.18-4.21) où les éléments $A_{12,3}^1$, $A_{13,2}^1$, et $A_{1,2,3}^1$ sont déterminés pour être substitués dans (4.4).

$$A_{23,1}^1 = \begin{cases} (A_1^1 \setminus A_2^1 \cup A_3^1) \times (A_2^1 \cap A_3^1) \times (A_2^1 \cap A_3^1) & \text{si } (y_{G_{1p}} \geq y_{s(G_2, G_3)}) \\ [(A_1^1 \setminus A_2^1 \cup A_3^1) \cup e^1] \times (A_2^1 \cap A_3^1) \times (A_2^1 \cap A_3^1) & \text{si } (y_{G_{1p}} < y_{s(G_2, G_3)}) \end{cases} \quad (4.17)$$

$$C_{i-jk} : y_{G_{ip}} - y_{s(G_j, G_k)}$$

$$C_1 : (y_{G_{3p}} \geq y_{s(G_1, G_2)}) \wedge (y_{G_{1p}} < y_{s(G_2, G_3)})$$

$$C_2 : (y_{G_2} \otimes C_{1-23} \geq y_{G_3}) \wedge (y_{G_{1p}} \geq y_{s(G_2, G_3)})$$

$$C_3 : (y_{G_3} = y_{G_2}) \wedge (y_{G_{1p}} < y_{s(G_2, G_3)})$$

$$C_4 : (y_{G_2} \otimes C_{1-23} < y_{G_3}) \wedge [(y_{G_3} - y_{G_1}) < y_{G_{2p}}] \wedge (y_{G_{1p}} \geq y_{s(G_2, G_3)})$$

$$C_5 : (y_{G_{3p}} > y_{s(G_1, G_2)}) \wedge (y_{G_3} > y_{G_2}) \wedge (y_{G_{2p}} > C_{3-12}) \wedge (y_{G_{1p}} < y_{s(G_2, G_3)})$$

$$C_6 : (y_{G_{3p}} \leq y_{s(G_1, G_2)}) \wedge (y_{G_{2p}} < y_{s(G_1, G_3)}) \wedge (y_{G_{1p}} < y_{s(G_2, G_3)})$$

$$C_7 : [(y_{G_3} - y_{G_1}) \geq y_{G_{2p}}] \wedge (y_{G_{1p}} \geq y_{s(G_2, G_3)})$$

$$C_8 : (y_{G_{2p}} \leq C_{3-12}) \wedge (y_{G_{1p}} \leq y_{s(G_2, G_3)})$$

$$A_{12,3}^1 = \begin{cases} (A_1^1 \cap A_2^1) \times (A_1^1 \cap A_2^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1) & \text{si } C_1 \vee (y_{G_{1p}} \geq y_{s(G_2, G_3)}) \\ (A_1^1 \cap A_2^1) \times (A_1^1 \cap A_2^1) \times [(A_3^1 \setminus A_1^1 \cup A_2^1) \cup e^1] & \\ & \text{si } (y_{G_{3p}} < y_{s(G_1, G_2)}) \wedge (y_{G_{1p}} \leq y_{s(G_2, G_3)}) \end{cases} \quad (4.18)$$

$$A_{13,2}^1 = \begin{cases} (A_1^1 \cap A_3^1) \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_1^1 \cap A_3^1) & \text{si } C_2 \vee C_3 \\ (A_1^1 \cap A_3^1) \times [(A_2^1 \setminus A_1^1 \cup A_3^1) \cup e^1] \times (A_1^1 \cap A_3^1) & \text{si } C_4 \vee C_5 \vee C_6 \\ (A_1^1 \cap A_3^1) \times e^1 \times (A_1^1 \cap A_3^1) & \text{si } C_7 \vee C_8 \end{cases} \quad (4.19)$$

$$A_{1,2,3}^1 = \begin{cases} [(A_1^1 \setminus A_2^1 \cup A_3^1) \times e^1 \times (A_3^1 \setminus A_1^1 \cup A_2^1)] \cup A_{13} & \text{si } (y_{G_{1p}} > y_{s(G_2, G_3)}) \\ e^1 \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1) & \\ & \text{si } (y_{G_{3p}} > y_{s(G_1, G_2)}) \wedge (C_{3-12} \leq y_{G_{2p}}) \wedge (y_{G_{1p}} \leq y_{s(G_2, G_3)}) \\ e^1 \times [(A_2^1 \setminus A_1^1 \cup A_3^1) \cup e^1] \times (A_3^1 \setminus A_1^1 \cup A_2^1) & \\ & \text{si } (y_{G_{3p}} > y_{s(G_1, G_2)}) \wedge (C_{3-12} > y_{G_{2p}}) \wedge (y_{G_{1p}} \leq y_{s(G_2, G_3)}) \\ \emptyset & \text{si } (y_{G_{3p}} \leq y_{s(G_1, G_2)}) \wedge (y_{G_{1p}} \leq y_{s(G_2, G_3)}) \end{cases} \quad (4.20)$$

$$A_{13}^1 = \begin{cases} [e^1 \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1)] \cup [e^1 \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times e^1] & si \quad (y_{G_2} \otimes C_{1-23} > y_{G_3}) \wedge (y_{G_3} > y_{G_1}) \\ e^1 \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times (A_3^1 \setminus A_1^1 \cup A_2^1) & si \quad (y_{G_2} \otimes C_{1-23} \leq y_{G_3}) \wedge (y_{G_3} > y_{G_1}) \\ e^1 \times (A_2^1 \setminus A_1^1 \cup A_3^1) \times e^1 & si \quad (y_{G_2} \otimes C_{1-23} > y_{G_3}) \wedge (y_{G_3} = y_{G_1}) \end{cases} \quad (4.21)$$

Ainsi, l'ensemble des équations (4.17-4.21) définit l'alphabet pour un système à 3 automates, où 3 paires sont en synchronisation, respectant un ordonnancement optimal forcé minimisant le makespan.

4.4.3 Alphabet pour un Système à 3 Automates ($n = 3$) avec Tierce Synchronisée

La synchronisation de 3 automates peut être interprétée comme un décalage des ordonnancements proposés précédemment. Le micro-alphabet déjà défini pour le produit précédent est modifié par l'insertion des micro-événements synchronisant la tierce (G_1, G_2, G_3). Le bloc $s(G_1, G_2, G_3)$ peut être considéré dès l'instant initial ou à n'importe quel point d'exécution pour lequel les 3 automates permettent la synchronisation. Cette inclusion des micro-événements liés à l'exécution de $s(G_1, G_2, G_3)$ n'a aucun effet sur les éléments déjà déterminés pour les autres configurations d'exécution, i.e. $A_{23,1}^1$, $A_{12,3}^1$ et $A_{13,2}^1$ et $A_{1,2,3}^1$.

4.4.4 Exemple pour un Système à 3 Automates ($n = 3$)

La Fig. 4.11, présente un exemple avec l'ensemble des phénomènes de synchronisation à $n = 3$. Comme précédemment, les modèles d'automates mono-état illustrent les conflits entre les opérations sans aucune perte de généralité, sachant qu'une représentation multi-états (des permutations) sous-jacente assure que chaque opération ne s'exécute qu'une seule fois.

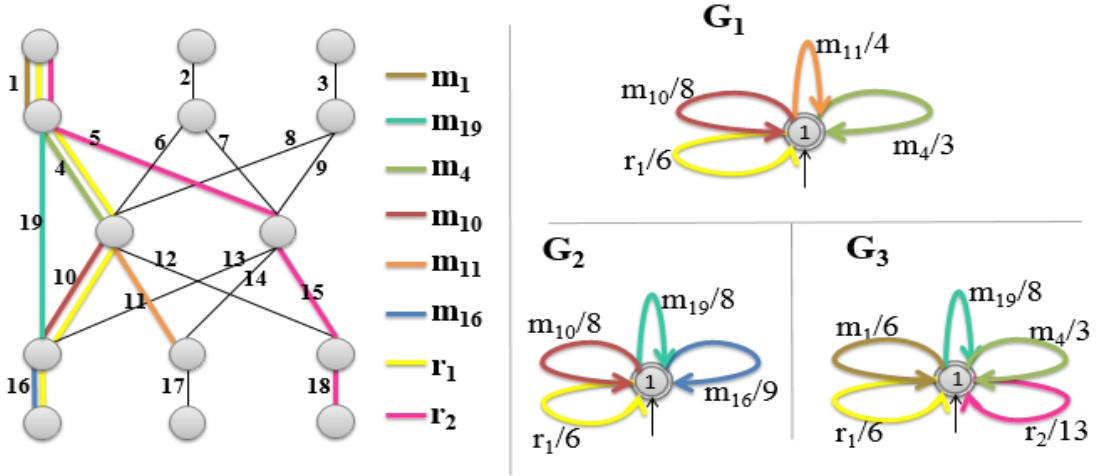


FIGURE 4.11: Système à 3 automates ($n = 3$) présentant tous les possibles types de synchronisation (représentation simplifiée mono-état pour faciliter la compréhension).

- Les alphabets originaux pour les automates locaux sont :

$$A_1 = \{r_1, m_{10}, m_{11}, m_4\},$$

$$A_2 = \{r_1, m_{10}, m_{19}, m_{16}\},$$

$$A_3 = \{r_1, m_1, m_{19}, m_4, r_2\}.$$

- Les dateurs globaux : $y_{G_1} = 21$, $y_{G_2} = 31$, et $y_{G_3} = 36$.

- Les dateurs privés : $y_{G_{1p}} = y(m_{11}) = 4$, $y_{G_{2p}} = y(m_{16}) = 9$, et $y_{G_{3p}} = y(m_1) \otimes y(r_2) = 19$.

- Les dateurs de synchronisation (pour les trois paires et la tierce) :

$$y_{s(G_1, G_2)} = y(m_{10}) = 8,$$

$$y_{s(G_1, G_3)} = y(m_4) = 3,$$

$$y_{s(G_2, G_3)} = y(m_{19}) = 8,$$

$$y_{s(G_1, G_2, G_3)} = y(r_1) = 6.$$

Les expressions (4.17-4.21), avec les valeurs des dateurs correspondants, définissent les éléments ci-dessous :

$$A_{23,1}^1 = \{(m_{11}^1, m_{19}^1, m_{19}^1), (e^1, m_{19}^1, m_{19}^1)\};$$

$$A_{12,3}^1 = \{(m_{10}^1, m_{10}^1, m_1^1), (m_{10}^1, m_{10}^1, r_2^1)\};$$

$$A_{13,2}^1 = \{(m_4^1, e^1, m_4^1)\};$$

$$A_{1,2,3}^1 = \{(e^1, m_{16}^1, m_1^1), (e^1, m_{16}^1, r_2^1), (e^1, e^1, m_1^1), (e^1, e^1, r_2^1)\}.$$

Par conséquent, le micro-alphabet (par substitution de termes selon (4.4)) correspond à :

$$A^1 = \{(r_1^1, r_1^1, r_1^1), (m_{11}^1, m_{19}^1, m_{19}^1), (e^1, m_{19}^1, m_{19}^1), (m_{10}^1, m_{10}^1, m_1^1), (m_{10}^1, m_{10}^1, r_2^1), (m_4^1, e^1, m_4^1), (e^1, m_{16}^1, m_1^1), (e^1, m_{16}^1, r_2^1), (e^1, e^1, m_1^1), (e^1, e^1, r_2^1)\}.$$

Chaque élément dans ce micro-alphabet représente l'exécution simultanée de plusieurs opérations pendant une unité de temps, e.g. $(m_{11}^1, m_{19}^1, m_{19}^1)$ dénote l'exécution de m_{11} dans G_1 , et m_{19} dans G_2 , et G_3 . En outre, A^1 garantit que ces éléments génèrent au moins une trajectoire telle que le makespan soit minimisé. Les trajectoires générées représenteraient alors des mots dans le langage du nouveau produit.

Trajectoires Générées

Par concaténation des éléments dans le micro-alphabet A^1 , des trajectoires (ou suites d'exécution) peuvent être obtenues. Les trajectoires T_1 et T_2 sont générées en respectant l'hypothèse 4.1 :

$$T_1 : (r_1^1, r_1^1, r_1^1)^6 (m_{11}^1, m_{19}^1, m_{19}^1)^4 (e^1, m_{19}^1, m_{19}^1)^4 (m_{10}^1, m_{10}^1, m_1^1)^6 (m_{10}^1, m_{10}^1, r_2^1)^2 (e^1, m_{16}^1, r_2^1)^9 (e^1, e^1, r_2^1)^2 (m_4^1, e^1, m_4^1)^3,$$

$$T_2 : (r_1^1, r_1^1, r_1^1)^6 (m_{11}^1, m_{19}^1, m_{19}^1)^4 (e^1, m_{19}^1, m_{19}^1)^4 (m_{10}^1, m_{10}^1, r_2^1)^8 (e^1, m_{16}^1, r_2^1)^5 (e^1, m_{16}^1, m_1^1)^4 (e^1, e^1, m_1^1)^2 (m_4^1, e^1, m_4^1)^3.$$

Pour T_1 , rappelant que la notation dans la définition 4.5 indique que $(a^1, b^1, c^1)^j \Leftrightarrow (a^j, b^j, c^j)$, l'opération r_1 est active pendant 6 unités de temps, autorisée communément par les 3 automates, jusqu'à son achèvement. Ensuite m_{11} est activée (dans G_1) simultanément avec m_{19} (dans G_2 et G_3) pendant 4 unités de temps jusqu'à l'achèvement de m_{11} . Puis m_{19} se poursuit pendant 4 unités de temps encore, et ainsi de suite. Les dateurs pour T_1 et T_2 prennent la même valeur, qui est le produit des dateurs des macro-événements composés. Par exemple, pour T_1 :

$$y(T_1) = y((r_1^1, r_1^1, r_1^1)^6) \otimes y((m_{11}^1, m_{19}^1, m_{19}^1)^4) \otimes \dots \otimes y((m_4^1, e^1, m_4^1)^3)$$

$$y(T_1) = [y(r_1^6) \oplus y(r_1^6) \oplus y(r_1^6)] \otimes [y(m_{11}^4) \oplus y(m_{19}^4) \oplus y(m_{19}^4)] \otimes \dots \otimes [y(m_4^3) \oplus y(e^3) \oplus y(m_4^3)]$$

$$y(T_1) = [6 \oplus 6 \oplus 6] \otimes [4 \oplus 4 \oplus 4] \otimes \dots \otimes [3 \oplus 3 \oplus 3]$$

$$y(T_1) = 6 \otimes 4 \otimes \dots \otimes 3 = 36.$$

Le dateur de la trajectoire $y(T_1)$ représente la date d'achèvement pour toutes les opérations dans cette séquence. En outre, dans la Fig. 4.11, l'automate du plus grand dateur est G_3 avec $y_{G_3} = 36$; confirmant que T_1 modélise l'exécution de toutes les opérations au plus tôt puisqu'il s'agit du temps minimum pour exécuter toutes les opérations dans G_3 . Remarque identique pour T_2 qui a le même dateur mais modélise une séquence d'exécution différente.

Les trajectoires, telles que T_1 et T_2 , sont issues de l'alphabet proposé et modélisent des exécutions optimales de toutes les opérations en exploitant le parallélisme entre les automates

locaux. Ainsi, l'approche optimise l'exécution d'opérations dans la phase de détermination du comportement du système global, i.e. de l'intégration des composants. Ce comportement est optimisé par analyse des ordonnancements des blocs d'événements résultant d'un classement d'automates par une caractérisation temporelle. Le comportement optimisé du système est alors un modèle décrivant, par le micro-alphabet proposé, les possibles exécutions de toutes les opérations au plus tôt.

Comparaison avec le Produit Synchrone d'Automates (Max, +)

L'approche proposée dans [44] relève de la détermination d'un produit synchrone pour automates (max, +). Les bases formelles de cette approche, comme l'extension des alphabets originaux, a inspiré les développements proposés ici. Afin de souligner la contribution de l'approche présentée, d'autres trajectoires obtenues par [44] (4.3) pour le même exemple seraient :

$$T_3 : r_1(m_{19}, m_{11})(m_{10}, m_1 r_2)(m_4, m_{16}).$$

$$T_4 : r_1(m_{10}, m_1 r_2)(m_{19}, m_{11})(m_4, m_{16}).$$

Les trajectoires T_3 et T_4 sont décrites sous la notation de [44]. Les dateurs pour T_3 , et T_4 , sont calculés comme proposé dans [44] (et de manière analogue au calcul ici) :

$$y(T_3) = y(r_1) \otimes y(m_{19}, m_{11}) \otimes y(m_{10}, m_1 r_2) \otimes y(m_4, m_{16})$$

$$y(T_3) = y(r_1) \otimes [y(m_{19}) \oplus y(m_{11})] \otimes [y(m_{10}) \oplus y(m_1 r_2)] \otimes [y(m_4) \oplus y(m_{16})]$$

$$y(T_3) = 6 \otimes [8 \oplus 4] \otimes [8 \oplus 19] \otimes [3 \oplus 9] = 6 \otimes 8 \otimes 19 \otimes 9 = 42.$$

Le dateur pour T_4 est identique puisque les mêmes événements composés sont considérés dans un ordre d'exécution différent.

Comme attendu, la synchronisation forcée entre opérations simultanées de durées différentes augmente le makespan⁹. A contrario, dans l'approche proposée par le nouveau produit synchrone, les dates d'achèvement des opérations simultanées sont plus flexibles et lorsqu'une opération vient d'achever son exécution, une autre peut exploiter ses ressources immédiatement, réduisant le makespan. Il est rappelé que, seule la syntaxe du modèle à base d'automates a été abordée ici (spécifiquement l'alphabet). Rien ne s'oppose à ce que cette proposition soit exploitée en synthèse d'automates (max, +) ou (min, +), pour répondre à des spécifications de contrôle¹⁰.

9. Ce qui est pourtant le but dans un modèle basé sur les automates (max, +).

10. en sachant qu'au niveau syntaxique les automates (min, +) et (max, +) ne peuvent pas être différenciés.

4.4.5 Extensions Envisagées pour $n > 3$

Pour les cas de plus de 3 automates locaux modélisant l'exclusion mutuelle entre les opérations il est possible de généraliser l'approche. Ici, seuls les aspects fondamentaux de la méthodologie sont décrits pour des exemples avec $n = 4$ et $n = 6$ pour illustration.

Système à 4 automates ($n=4$)

Pour 4 automates locaux, le classement répondant à la définition 4.13, est présenté par (4.22). Dans le cas le plus général, où tous les types de synchronisation surviennent, les dateurs globaux sont définis par (4.23-4.26).

$$y_{G_1} \leq y_{G_2} \leq y_{G_3} \leq y_{G_4} \quad (4.22)$$

$$y_{G_1} = y_{G_{1p}} \otimes y_{s(G_1, G_2)} \otimes y_{s(G_1, G_3)} \otimes y_{s(G_1, G_4)} \otimes y_{s(G_1, G_2, G_3)} \otimes y_{s(G_1, G_2, G_4)} \otimes y_{s(G_1, G_3, G_4)} \otimes y_{s(G_1, G_2, G_3, G_4)} \quad (4.23)$$

$$y_{G_2} = y_{G_{2p}} \otimes y_{s(G_1, G_2)} \otimes y_{s(G_2, G_3)} \otimes y_{s(G_2, G_4)} \otimes y_{s(G_1, G_2, G_3)} \otimes y_{s(G_1, G_2, G_4)} \otimes y_{s(G_2, G_3, G_4)} \otimes y_{s(G_1, G_2, G_3, G_4)} \quad (4.24)$$

$$y_{G_3} = y_{G_{3p}} \otimes y_{s(G_1, G_3)} \otimes y_{s(G_2, G_3)} \otimes y_{s(G_3, G_4)} \otimes y_{s(G_1, G_2, G_3)} \otimes y_{s(G_2, G_3, G_4)} \otimes y_{s(G_1, G_3, G_4)} \otimes y_{s(G_1, G_2, G_3, G_4)} \quad (4.25)$$

$$y_{G_4} = y_{G_{4p}} \otimes y_{s(G_1, G_4)} \otimes y_{s(G_2, G_4)} \otimes y_{s(G_3, G_4)} \otimes y_{s(G_1, G_2, G_4)} \otimes y_{s(G_2, G_3, G_4)} \otimes y_{s(G_1, G_3, G_4)} \otimes y_{s(G_1, G_2, G_3, G_4)} \quad (4.26)$$

Comme indiqué précédemment ($n = 3$), plusieurs inégalités peuvent être obtenues afin d'analyser le comportement temporel entre les sous-systèmes. Ainsi, de (4.22) les expressions $y_{G_1} \leq y_{G_2}$, $y_{G_2} \leq y_{G_3}$, et $y_{G_3} \leq y_{G_4}$ peuvent être obtenues, et de manière transitive : $y_{G_1} \leq y_{G_3}$, $y_{G_1} \leq y_{G_4}$, et $y_{G_2} \leq y_{G_4}$.

Pour 4 automates locaux ($n = 4$), il y a 6 paires possibles (6 possibilités pour le scénario d'une paire en synchronisation). L'analyse déterminant l'alphabet pour chacune de ces 6 possibilités reste identique. En considérant seulement la paire en synchronisation (G_1, G_2), de la règle générale de classement (et les inégalités associées) il est possible de définir les expressions (4.27-4.32). De cet ensemble d'inégalités, les relations temporelles entre des ensembles d'opérations s'exécutant simultanément peuvent être déduites. La Fig. 4.12 montre l'ordonnancement qualitatif optimal proposé basé sur cette configuration.

$$y_{G_1} \leq y_{G_2} \quad \Rightarrow \quad y_{G_{1p}} \leq y_{G_{2p}} \quad (4.27)$$

$$y_{G_2} \leq y_{G_3} \quad \Rightarrow \quad y_{G_{2p}} \otimes y_{s(G_1, G_2)} \leq y_{G_{3p}} \quad (4.28)$$

$$y_{G_3} \leq y_{G_4} \Rightarrow y_{G_{3p}} \leq y_{G_{4p}} \quad (4.29)$$

$$y_{G_1} \leq y_{G_2} \Rightarrow y_{G_{1p}} \otimes y_{s(G_1, G_2)} \leq y_{G_{3p}} \quad (4.30)$$

$$y_{G_2} \leq y_{G_4} \Rightarrow y_{G_{2p}} \otimes y_{s(G_1, G_2)} \leq y_{G_{4p}} \quad (4.31)$$

$$y_{G_1} \leq y_{G_4} \Rightarrow y_{G_{1p}} \otimes y_{s(G_1, G_2)} \leq y_{G_{4p}} \quad (4.32)$$

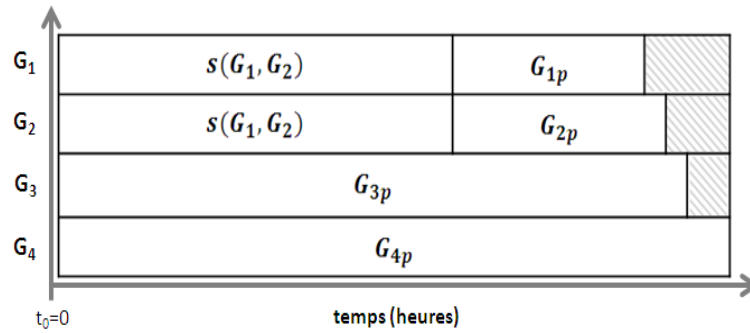


FIGURE 4.12: Ordonnancement qualitatif optimal pour $n = 4$ et (G_1, G_2) comme la seule paire en synchronisation.

Cet ordonnancement reste intuitif puisqu'une seule paire d'automates en synchronisation est considérée. Le parallélisme est exploité au plus tôt avec l'exécution du bloc en synchronisation avec les événements privés possibles dans les automates G_3 et G_4 , comme rapporté dans l'hypothèse 4.3. Les segments grisés dans l'ordonnancement indiquent les intervalles des possibles dates d'achèvement des blocs précédents (en cohérence avec les expressions (4.27-4.32)).

L'ordonnancement qualitatif répond à une logique d'intégration de blocs permettant de placer des ensembles d'opérations (i.e. blocs) sur l'axe temporel tel que des intervalles oisifs pour les ressources soient évités, et tel que toutes les opérations soient exécutées au plus tôt.

Pour une meilleure illustration, l'exemple de la Fig. 4.13 présente une instance du terminal pétrolier modélisée par 4 automates locaux. Il est rappelé que la représentation mono-état vise à illustrer de manière simple l'exclusion mutuelle des opérations de transfert ou maintenance sollicitant des ressources conflictuelles.

Dans la Fig. 4.13, les 4 automates se synchronisent sur l'opération r_6 . D'autre part, les paires synchronisées sont : (G_1, G_3) , par les opérations r_2 et m_6 , (G_2, G_4) par r_4 , et (G_3, G_4) par r_5 . Suivant la règle de classement de dateurs, et la logique de blocs, la Fig. 4.14 présente les résultats finaux. Naturellement, avec l'augmentation de la complexité des relations entre sous-systèmes, la structure de l'ordonnancement se complexifie. La Fig. 4.14a) donne les expressions des dateurs selon la règle de classement, et la Fig. 4.14b) présente l'ordonnancement obtenu. Pour cet ordonnancement, il a été supposé que $y_{s(G_1, G_3)} < y_{s(G_2, G_4)}$ afin d'illustrer que l'exécution de ces blocs d'opérations pour 2 paires en synchronisation ne se terminent pas forcément au

même instant. Les blocs grisés qui suivent les blocs $s(G_1, G_3)$ dénotent l'intervalle des dates d'achèvement possibles.

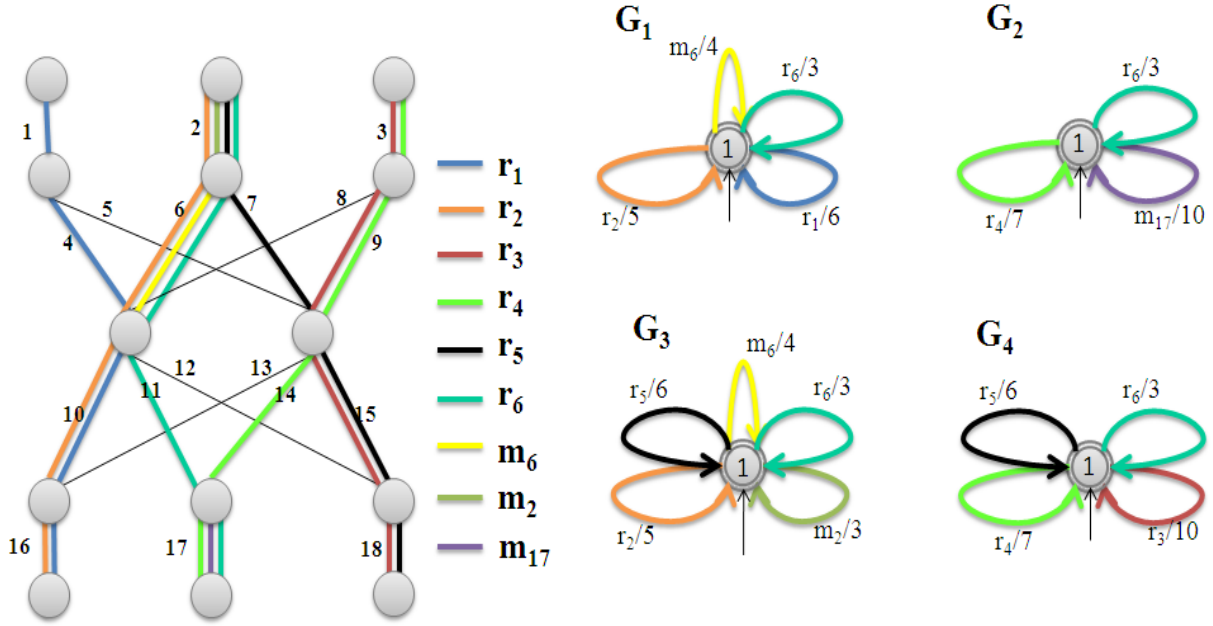


FIGURE 4.13: Exemple pour 4 automates locaux (représentation simplifiée à un seul état pour faciliter la compréhension).

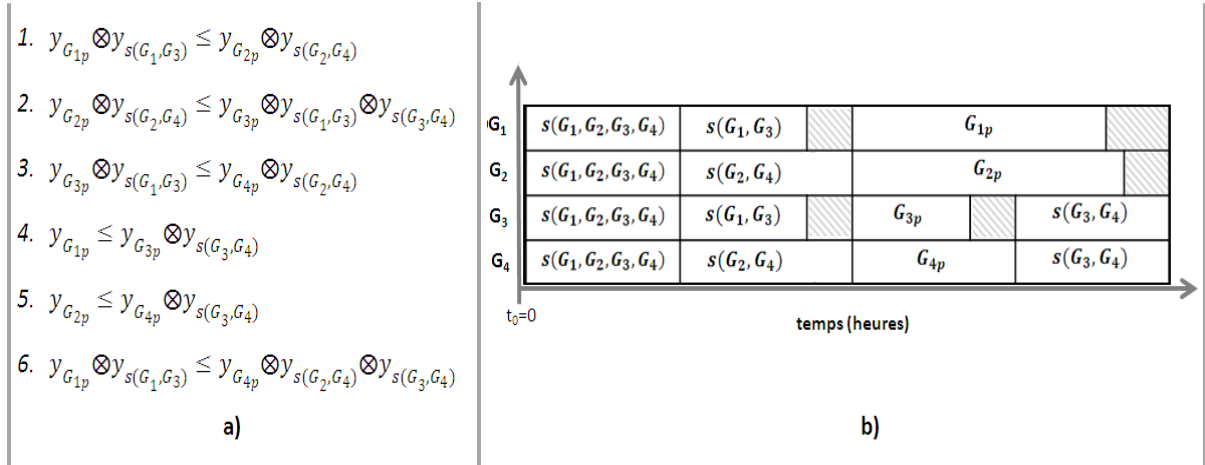


FIGURE 4.14: Résultats d'ordonnancement pour l'exemple dans la Fig. 4.13.

Plusieurs scénarios de synchronisation possibles sont définis lorsqu'un système comporte 4 automates locaux. Pour une seule paire d'automates en synchronisation 6 alternatives sont possibles, $\binom{6}{2} = 15$ possibilités pour 2 paires, et $\binom{6}{i}$ pour i paires $\forall i \in \{1, \dots, 6\}$. Aussi, d'autres scénarios incluent : les tierces, la synchronisation de 4 automates, et des combinaisons de tous les scénarios susmentionnés (e.g. 2 paires et une tierce). Ceci donne une idée de l'augmentation de la complexité et de la raison pour laquelle une formulation générique n'a pas été effectuée pour tous

les cas possibles. Pourtant, une méthodologie basée sur un classement de dateurs a été proposée dans ce chapitre pour déterminer le micro-alphabet pour l'ordonnancement optimal.

D'autres problèmes peuvent être étudiés avec la démarche proposée. Par exemple, si un ensemble spécifique d'opérations doit être exécuté au début ou à la fin de la séquence, alors le classement d'automates et l'analyse par blocs permettra identifier les conséquences de telles contraintes au niveau de l'ordonnancement.

Système à 6 automates ($n=6$)

Un exemple pour $n = 6$ est présenté dans la Fig. 4.15. Il décrit la synchronisation de 3 paires d'automates : (G_1, G_4) , (G_2, G_5) , et (G_1, G_6) . L'ordonnancement présenté est optimal et les segments grisés indiquent les dates possibles d'achèvement des blocs d'opérations (il a été supposé¹¹ que $y_{s(G_2, G_5)} > y_{s(G_1, G_4)}$). L'exécution de $s(G_1, G_6)$ est conditionnée par l'achèvement de G_{6p} , puisqu'il est connu (par la règle de classement, i.e. $y_{G_6} \geq y_{G_1} \Rightarrow y_{G_{6p}} \geq y_{s(G_1, G_4)} \otimes y_{G_{1p}}$) que G_{1p} achèvera avant ou en même temps.

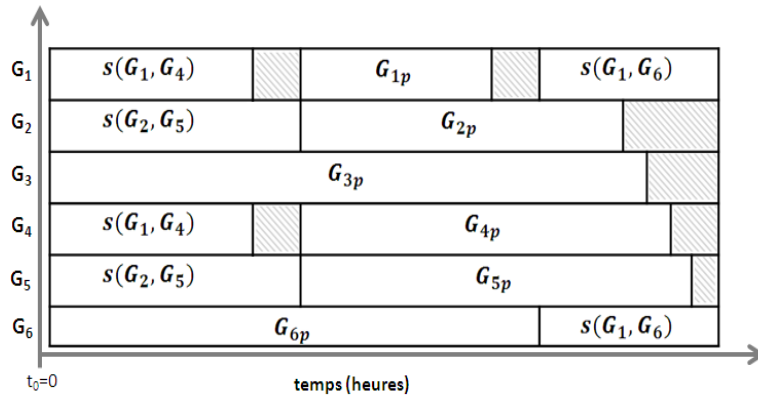


FIGURE 4.15: Résultats pour l'ordonnancement qualitatif pour $n = 6$ et 3 paires d'automates en synchronisation : (G_1, G_4) , (G_2, G_5) et (G_1, G_6) .

L'approche proposée dans ce chapitre est ainsi bien adaptée aux systèmes pour lesquels un ensemble d'opérations doit être exécuté au plus tôt sous conflit de partage de ressources. Elle relève d'un classement d'automates en fonction de l'identification des dateurs, ce qui permet ensuite une étude des ordonnancements qualitatifs par une vue d'intégration de blocs (d'événements privés ou de synchronisation). Ces ordonnancements minimisant le makespan permettent de déduire l'alphabet sous-jacent. L'avantage d'un cadre mathématique formel, en comparaison aux techniques approximatives (tels que heuristiques), provient du fait que le comportement du système peut être analysé à partir d'un modèle de référence.

11. pour illustrer que l'exécution des blocs d'opérations pour 2 paires en synchronisation ne se terminent pas forcément au même instant.

Ce travail présente les premiers efforts pour définir un produit synchrone basé sur la décomposition de transitions pour générer un comportement temporel global au plus tôt. Le classement d'automates à base de dateurs permet de dériver les conditions formelles construisant les ordonnancements optimaux par une abstraction de blocs d'opérations.

La méthodologie pourrait être aussi exploitée pour analyser les conséquences de certaines manipulations telles que : 1) l'ajout ou suppression de certaines opérations dans le comportement global du système ; ou 2) l'affectation de ressources différentes pour les opérations générant des changements logiques au niveaux des conflits et temporels dans l'ordonnancement. Des analyses pourraient être appliquées au modèle mathématique visant à déduire les changements opérationnels pour obtenir le comportement souhaité.

L'approche est à ce point plus adéquate pour des systèmes dont la modélisation doit reposer sur des approches formelles, et implique des développements plus élaborés et denses lorsque n augmente. Cependant, la formulation de l'alphabet reste complexe.

Conclusions Partielles

Une approche a été proposée pour la détermination d'un nouvel alphabet pour le produit synchrone d'automates tropicaux sous comportement au plus tôt. L'approche est appliquée à la problématique d'engagement de ressources conflictuelles pour un ensemble d'opérations de transfert et de maintenance. La modélisation des conflits par automates locaux est intuitive et les ordonnancements optimaux forcés sont obtenus par une analyse des informations temporelles des sous-systèmes, notamment à l'aide d'un cadre mathématique basé sur des dateurs. Un micro-alphabet pouvant décrire les ordonnancements optimaux est obtenu par une abstraction d'exécution de micro-événements, ce qui permet de flexibiliser la représentation du parallélisme.

L'approche présentée s'est limitée à la syntaxe du produit. Les travaux futurs envisagés à partir de ces résultats incluent la définition des aspects de sémantique (i.e. le langage) et la synthèse de contrôleurs pour répondre à des spécifications supplémentaires. Dans le cadre prospectif de la théorie de la commande, les domaines d'application de l'approche sont divers, incluant des systèmes exécutant de nombreuses opérations avec des ressources limitées et au plus tôt (e.g. systèmes de production, systèmes embarqués et réseaux de différente nature tel que les télécommunications et le transport).

Chapitre 5

Conclusions et Perspectives

5.1 Conclusion Générale

Cette recherche a porté sur l'étude des approches tropicales pour des problèmes d'optimisation dans un terminal maritime pétrolier en vue d'assister les opérateurs de supervision. Les approches proposées restent applicables aux réseaux de flux en général, voire des systèmes de nature différente. Une recherche sur diverses techniques alternatives de modélisation a permis de constater : 1) une complémentarité des techniques en fonction des applications (i.e. réseaux de Petri vs algèbre $(\max, +)$, TPN vs automates temporisés, théorie des graphes vs algèbres tropicales) ; 2) des avantages des approches tropicales pour le problème d'optimisation traité ; et 3) des perspectives pour des nouvelles propositions considérant des approches alternatives. Dans le cadre de la modélisation, les approches retenues ont été l'algèbre $(\max, +)$ et les automates tropicaux. D'autre part, dans le cadre de la technique de résolution, les algorithmes génétiques ont été retenus du fait de son importance dans la communauté scientifique et d'une implémentation directe, ouvrant ainsi un spectre de possibilités pour la transposition des propositions présentées ici sur des approches alternatives de l'intelligence artificielle.

La puissance de l'algèbre $(\max, +)$ pour la modélisation des phénomènes de synchronisation a été exploitée en vue de la synthèse du modèle en une contrainte unique. Les phénomènes de conflit de partage de ressources ont été modélisés par des variables de décision conduisant à des modèles $(\max, +)$ non linéaires. Les avantages de manipulation des systèmes $(\max, +)$ linéaires ont été exploitées par des propositions de priorisation d'opérations conflictuelles (résultant en une linéarisation du fait de l'affectation des variables de décision). Ces derniers résultats présentent des approches candidates pour des applications industrielles dans lesquelles une telle priorisation est envisageable. Dans ces cas, la détermination de l'ordonnancement consiste alors à un simple produit matriciel $(\max, +)$.

Les caractéristiques du système industriel étudié ont permis d'identifier des objectifs non commensurables qui ont été représentés par un modèle d'optimisation multi-objectif. Une approche a été proposée par l'intégration de modèles $(\max,+)$ non linéaires avec des algorithmes génétiques. Les bases de cette approche sont une modélisation intuitive et concise des contraintes du système, suivie de l'application d'un algorithme génétique adapté sous une structure génomique hiérarchique permettant un contrôle entre les gènes complémentaires d'un même niveau (en vue d'éviter un éloignement de l'espace de faisabilité). Il a été possible d'obtenir des systèmes d'équations $(\max,+)$ linéaires pour chaque individu (déterminant les dateurs des opérations) facilement exploitables. Ainsi, les meilleures solutions représentant les compromis les plus avantageux entre plusieurs objectifs sont atteintes.

En vue d'assister les opérateurs de supervision, les résultats obtenus permettent de présenter des solutions candidates complétant son expertise en temps réel. Sachant que dans le cas d'étude, ainsi que dans diverses applications, les opérateurs disposent rarement d'informations sur les impacts financiers de leur décisions¹, les résultats fournis permettent bien d'intégrer ces informations à celles gérées par les opérateurs (telles que topologie du réseau, disponibilité des dispositifs et programmes de maintenance), sans les présenter de manière explicite et en permettant d'améliorer la qualité des décisions.

Le domaine spécifique considérée relève des systèmes à événements discrets. À ce titre, des techniques formelles ont également été abordées. Du fait du caractère temporel de la problématique, les automates tropicaux ont été exploités pour une modélisation simple des conflits de partage de ressources. Sachant qu'un comportement au plus tôt supposerait une exploitation maximale du parallélisme entre opérations, un nouveau produit synchrone d'automates tropicaux a été considéré. Ainsi, la contribution développée, porte sur la détermination des aspects syntaxiques de ce produit (l'alphabet). Les formulations proposées du nouvel alphabet portent sur un cadre théorique basé sur des dateurs. Il est ainsi possible de classer les sous-systèmes (automates locaux), et de déterminer des ordonnancements au plus tôt par une analyse de décomposition des opérations en opérations atomiques flexibilisant le parallélisme et, par conséquent, l'exploitation au plus tôt des ressources. Les alphabets deviennent plus denses, en comparaison avec des produits synchrones pour des automates $(\max,+)$ modélisant un comportement au plus tard. Le résultat global des investigations dans le cadre formel est l'alphabet pour un produit d'automates permettant d'optimiser le comportement (à une exécution au plus tôt) lors de la détermination du modèle global du système et en absence d'une commande par supervision.

Les travaux de recherche se sont déroulés en interaction avec les partenaires industriels fournissant les données nécessaires pour les spécifications opérationnelles du système et la validation des propositions. Ce dynamisme a permis ainsi une modularisation maîtrisée des avancées scientifiques. Des approches mono-objectif ont été proposées par l'exploitation des

1. souvent dû à des raisons de confidentialité.

modèles $(\max, +)$ non linéaires, étendues ensuite au multi-objectif.

5.2 Perspectives

Dans le cadre des techniques formelles, en partant des développements proposés ici pour l'alphabet du nouveau produit d'automates tropicaux, les aspects sémantiques tels que le langage devraient être abordés. Cela permettra de compter sur un modèle formel pour la détermination du comportement global du système. Ensuite, l'exploitation des approches de commande par supervision pourra être considérée en vue de restreindre le comportement du système par rapport à des contraintes supplémentaires.

Au niveau de la maintenance, et dans un cadre opérationnel spécifique, l'inclusion des aspects de maintenance prédictive seraient à explorer, en vue de déterminer le degré de flexibilité pour l'avancement ou décalage des opérations de maintenance en fonction de la dégradation des ressources.

L'exploration des techniques alternatives de modélisation a donné des perspectives pour envisager des propositions hybrides avec les algèbres tropicales. Dans le cadre des automates cellulaires, en sachant qu'ils peuvent être exploités pour des problèmes d'ordonnancement, il semble prometteur d'explorer l'intégration des algèbres tropicales dans la définition des règles d'évolution.

Enfin, vis-à-vis de l'apport proposé pour l'intégration des algorithmes génétiques avec l'algèbre $(\max, +)$, il peut être considéré qu'une intégration analogue émerge pour tout système pouvant s'exprimer de manière linéaire, de forme $X = AX \oplus b$, lors de l'affectation des variables de décision. Ainsi, des problèmes d'optimisation exprimés sur l'algèbre $(\min, +)$ pourraient être traités de manière équivalente à l'approche proposée ici avec les algorithmes génétiques. En outre, des intégrations avec d'autres approches de l'intelligence artificielle (e.g. essais particuliers et colonies de fourmis) sont à exploiter. Enfin, en termes de performance, des algorithmes autres que le SPEA peuvent être exploités dans le cadre des propositions présentées dans ces investigations.

Glossaire

- *Alignement* : chemin reliant 2 nœuds par des segments. Dans le cas de pipelines, un alignement est formé par des dispositifs de contrôle de flux (vannes dans ce mémoire) ainsi que par des segments de pipelines.
- *Algèbres Tropicales* : ensemble des algèbres nommées ainsi à cause de son précurseur Imre SIMON, [19]. Parmi les algèbres tropicales, les algèbres $(\max, +)$ et $(\min, +)$ peuvent être citées.
- *Automate $(\max, +)$* : automate avec des transitions pondérées par des gains. Le coût associé à un mot w dans l'automate est le coût *maximal* parmi les coûts de tous les chemins décrits avec le même mot w .
- *Automate $(\min, +)$* : automate avec des transitions pondérées par des coûts. Le coût associé à un mot w dans l'automate est le coût *minimal* parmi les coûts de tous les chemins décrits avec le même mot w .
- *Automate Mono-état* : automate tropical modélisant 'n' opérations conflictuelles (mutuellement exclusives) à l'aide d'un état et 'n' transitions (à la fois sortantes et entrantes car l'état final est le même état initial).
- *Automate Tropical* : terme utilisé pour dénoter un automate $(\max, +)$ ou $(\min, +)$ de manière indistincte.
- *Batch* (anglicisme) : terme dénotant un lot de pétrole (de quantité et spécifications données) demandé par un client.
- *Capacité Opérative* : nombre maximum d'alignements disjoints qui peuvent être habilités simultanément dans un réseau. Défini dans [1].

- *Commensurable* : 2 éléments sont commensurables s'ils partagent la même mesure et, par conséquent, peuvent pas être évalués de manière groupée.
- *Commutation* : passage du fonctionnement d'un dispositif d'un état à un autre. Ici, utilisé pour dénoter les passages de fermeture à ouverture, ou vice versa, pour les vannes.
- *Dateur* : variable décrivant une date fondamentale dans l'exécution d'un événement. Elle correspond à la date de début d'une opération dans le cadre de l'ordonnancement d'opérations (Chapitre 2), et à la date d'achèvement d'un événement ou séquence d'événements dans le cadre d'automates tropicaux (Chapitre 4).
- *Habilitation* (d'un alignement) : capacité de faire transférer du pétrole du point source au point puits en considérant que toutes les vannes pour le transfert sont ouvertes et toutes les vannes adjacentes sont fermées.
- *Makespan* (anglicisme) : temps d'exécution de l'ensemble des opérations dans un système où chacune s'exécute une seule fois.
- *Micro-événement* : événement de durée atomique (1 unité de temps), voir définition [4.1](#).
- *Macro-événement* : suite de n micro-événements. Un macro-événement possède une durée de n unités de temps, voir définition [4.3](#).
- *Micro-événement Composé* : représentation de l'exécution de micro-événements de manière simultanée dans un ensemble d'automates. Dans chaque automate s'exécute un seul micro-événement, voir définition [4.4](#).
- *Macro-événement Composé* : représentation de l'exécution de macro-événements de manière simultanée dans un ensemble d'automates. Dans chaque automate s'exécute un macro-événement de même durée que dans les autres, voir définition [4.5](#).
- *Micro-événement (Macro-événement) Vide* : micro(macro)-événement dénotant qu'aucune opération ne s'exécute pendant 1 unité de temps, dans le cas du micro-événement vide, ou pendant n unités de temps, pour le cas du macro-événement vide, voir définition [4.7](#).
- *Opération* : terme utilisé pour faire référence aux activités de maintenance, ainsi qu'aux activités de transfert de batch de pétrole aux clients, i.e. opération de maintenance ou

opération de transfert.

- *Pipeline* (anglicisme) : canalisation, configuration de tuyaux.
- *Ressource* : dispositif devant être affecté à une opération lors de l'exploitation du réseau. Ici, les ressources sont représentées par les vannes.
- *Simultanée (Exécution)* : terme utilisé pour faire référence à des opérations qui s'exécutent en parallèle mais qui ne débutent ou terminent pas forcément à la même date.
- *Synchronisée (Exécution)* : une exécution synchronisée de 2 ou plus opérations suppose qu'elles débutent et terminent en même temps. Ici, le terme est utilisé lors que le même événement s'exécute en même temps sur un ensemble d'automates, i.e. les automates sont dit synchronisés par cet événement.
- *Solver* (anglicisme) : logiciel d'optimisation permettant d'obtenir, à travers d'algorithmes dédiés et prédéfinis, des solutions pour des instances des modèles d'optimisation fournis.
- *Transition Atomique* : fait référence à une transition à une durée d'une unité de temps.

Bibliographie

- [1] J. Rojas-D'Onofrio. Capacité opérative des réseaux de transfert de pétrole., 2011. Thèse doctorale. Lyon, France : INSA de Lyon. Laboratoire Ampère UMR CNRS 5005.
- [2] J. Rojas-D'Onofrio, J. González, E. Boutleux, and E. Niel. Path search algorithm minimizing interferences with envisaged operations in a pipe network. In *Proceedings of the European Control Conference - ECC'09*, 2009.
- [3] J. Rojas-D'Onofrio, J. Márquez, E. Boutleux, and E. Niel. Path search algorithm for connections with pumps in crude oil pipe networks. In *18th IFAC World Congress*, 2011. Milano (Italy) August 28.
- [4] K. Quintero, E. Niel, and J. Rojas-D'Onofrio. Optimizing process supervision in a flow network in terms of operative capacity & failure risk. In *15th International Congress on Automation, Systems and Instrumentation*, 2011.
- [5] R. Karuppiah, K.C. Furman, and I.E. Grossmann. Global optimization for scheduling refinery crude oil operations. *Computers & Chemical Engineering*, 32(11) :2745–2766, 2008.
- [6] A. Nait-Sidi-Moh, M.-A. Manier, A. El Moudni, and M. Wack. Petri net with conflicts and (max,plus) algebra for transportation systems. In *11th IFAC Symposium on Control in Transportation Systems*, volume 11, pages 548–553, 2006.
- [7] I. Nasri, G. Habchi, and R. Boukezzoula. An algebraic max-plus model for HVLV systems scheduling and optimization with repetitive and flexible periodic preventive maintenance : just-in-time production. In *9th International Conference of Modeling, Optimization and Simulation - MOSIM'12*, 2012.
- [8] K. Balachandran, J.D. Bendtsen, R.L. Olsen, and J.M. Pedersen. Energy resource management in smart grid. In *Complexity in Engineering (COMPENG), 2012*, pages 1–6, 2012. Publisher : IEEE. DOI : 10.1109/CompEng.2012.6242960.
- [9] A.K. Hutzschenreuter, P. Bosman, and H. La Poutre. Evolutionary multiobjective optimization for dynamic hospital resource management. In Springer, editor, *Evolutionary Multi-Criterion Optimization*, volume 5467, pages 320–334. 2009. Proceedings of the 5th International Conference, EMO 2009, Nantes, France, April 7-10, 2009.

- [10] S.-H. Ok, W.-J. Seo, J.-H. Ahn, S. Kang, and B. Moon. An ant colony optimization approach for the preference-based shortest path search. In *Communication and Networking*, pages 539–546. Springer, 2009. DOI : 10.1007/978-3-642-10844-0_63.
- [11] T. Erlebach. Approximation algorithms for edge-disjoint paths and unsplittable flow. In *Efficient Approximation and Online Algorithms*, volume 3484, pages 97–134. Springer, 2006. DOI : 10.1007/11671541_4.
- [12] S. Yuan, B. Raghavachari, and A. Patel. Finding maximum reliable path in mesh networks under multiple failures. In *Proceedings, 19th International Conference on Computer Communications and Networks (ICCCN 2010)*, pages 1–6, 2010. IEEE.
- [13] S. Gyapay, A. Pataricza, J. Sziray, and F. Friedler. Petri net-based optimization of production systems. In *6th IEEE International Conference on Intelligent Engineering Systems*, pages 465–469, 2002.
- [14] M.J.C. e Silva, W.J. Silva, and P.R.M. Maciel. Modelling and analysis in production system : an approach based on petri net. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 4354 – 4359, 2004. DOI : 10.1109/ICSMC.2004.1401216.
- [15] Wang J. Modeling closed loop supply chain system based on stochastic fuzzy petri net. In *International Conference on Management Science and Engineering (ICMSE)*, pages 598 – 603, 2013. DOI : 10.1109/ICMSE.2013.6586341.
- [16] O.H. Roux, D. Delfieu, and P. Molinaro. Discrete time approach of time petri nets for real-time systems analysis. In *Proceedings of 8th IEEE International Conference on Emerging Technologies and Factory Automation. IEEE. DOI : 10.1109/ETFA.2001.997686.*, 2001.
- [17] M. Alsaba, J-L. Boimond, and S. Lahaye. Sur la commande des systèmes flexibles de production manufacturière par l’algèbre des diïdes. *Revue e-STA, Sciences et Technologies de l’Automatique*, 4(2) :3247–3259, 2007.
- [18] W. Ait-Cheik-Bihi, A. Nait-Sidi-Moh, and M. Wack. Conflict management and resolution using $(\max, +)$ algebra : Application to services interaction. In *Evaluation and Optimization of Innovative Production Systems of Goods and Services : 8th International Conference of Modeling and Simulation - MOSIM’10*, 2010.
- [19] I. Simon. Limited subsets of a free monoid. In *Proceedings of the 19th Annual Symposium on Foundations of Computer Science, IEEE Computer Society*, pages 143–150, 1978.
- [20] F. Baccelli, G. Cohen, G. Jan-Olsder, and J-P. Quadrat. *Synchronization and Linearity an Algebra for Discrete Event Systems*. Wiley, 2001.
- [21] M. Gondran. Algèbre linéaire et cheminement dans un graphe. *RAIRO - Operations Research*, 9 :77–99, 1975.

- [22] G. Cohen, S. Gaubert, and J-P. Quadrat. L'algèbre des sandwichs. *Pour la Science*, (328) : 56–63, 2005.
- [23] R. Cuninghame-Green. *Lecture notes in economics and mathematical systems*. Springer-Verlag. Berlin-Heidelberg New York, 1979.
- [24] M. Andersen. Max-plus algebra : properties and applications, 2002. Projet de Master en Mathématiques. University of Wyoming, USA.
- [25] A. Nowak. The tropical eigenvalue-vector problem from algebraic, graphical, and computational perspectives, 2014. Honors Theses (Paper 97). Department of Mathematics. Bates College.
- [26] L. Houssin. Cyclic jobshop problem and (max,plus) algebra. In *World IFAC Congress (IFAC 2011)*, pages 2717–2721, 2014.
- [27] M. Akian, S. Gaubert, and A. Guterman. Tropical polyhedra are equivalent to mean payoff games. *International Journal of Algebra and Computation*, 22(1), 2012. DOI : 10.1142/S0218196711006674.
- [28] B. Heidergott, G.J.and Olsder, and J.V.D. Woude. *Max Plus at work*. Princeton Press, 2006.
- [29] L. Houssin, S. Lahaye, and J-L. Boimond. Commande en juste-à-temps sous contraintes de systèmes (max,+)-linéaires. *Journal Européen des Systèmes Automatisés - JESA*, 39(1-3) : 335–350, 2005.
- [30] Z. Königsberg. Modeling, analysis and timetable design of a helicopter maintenance process based on timed event petri nets and max-plus algebra. *Neural, Parallel & Scientific Computations*, 18(1) :1–12, 2010.
- [31] T.L. Turocy and B. Von Stengel. Game theory. *Encyclopedia of Information Systems*. Elsevier, pages 403–420, 2003. doi :10.1016/B0-12-227240-4/00076-9.
- [32] C. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, 2nd edition, 2006.
- [33] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126 (2) :183–235, 1994. doi :10.1016/0304-3975(94)90010-8.
- [34] B. Bérard. An introduction to timed automata. In *Control of Discrete-Event Systems. Lecture Notes in Control and Information Sciences*, volume 433, pages 169–187. Springer, 2013. Chapter 9. DOI : 10.1007/978-1-4471-4276-8₉.

- [35] Y. Abdeddaim, A. Kerbaa, and O. Maler. Task graph scheduling using timed automata. In *Proceedings of the International Parallel and Distributed Processing Symposium*, 2003. IEEE. DOI : 10.1109/IPDPS.2003.1213431.
- [36] Y. Abdeddaim and O. Maler. Job-shop scheduling using timed automata. *Computer Aided Verification. Lecture Notes in Computer Science*, 2102 :478–492, 2001. DOI : 10.1007/3-540-44585-4_46.
- [37] K.T. Seow, M. Gai, and T.L. Lim. A temporal logic specification interface for automata-theoretic finitary control synthesis. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation - ICRA. IEEE*. DOI : 10.1109/ROBOT.2005.1570178, 2005.
- [38] M. Wang, Y. Qian, and X. Guang. Improved calculation method of shortest path with cellular automata model. *Kybernetes*, 41(3) :508 – 517, 2012.
- [39] F. Seredynski and A.Y. Zomaya. Sequential and parallel cellular automata-based scheduling algorithms. *IEEE Transactions on Parallel and Distributed Systems*, 13(10) :1009 – 1023, 2002. DOI : 10.1109/TPDS.2002.1041877.
- [40] S. Gaubert. Performance evaluation of $(\max, +)$ automata. *IEEE transactions on Automatic Control*, 40(12) :2014–2025, 1995.
- [41] S. Lahaye, J. Komenda, and J. Boimond. Modélisation modulaire à l’aide d’automates $(\max, +)$. In *CIFA*, pages 1036–1041, 2012.
- [42] S. Lahaye, J. Komenda, and J. Boimond. Compositions of $(\max, +)$ automata. *Discrete Event Dynamic Systems*, 2014. Springer. DOI : 10.1007/s10626-014-0186-6.
- [43] J. Komenda, S. Lahaye, and J. Boimond. Supervisory control of $(\max, +)$ automata : A behavioral approach. *Discrete Event Dynamic Systems*, 19(4) :525–549, 2009.
- [44] J. Komenda, S. Lahaye, and J-L. Boimond. Le produit synchrone des automates $(\max, +)$. *Special Issue of Journal Européen des Systèmes Automatisés - JESA*, 43(7-9) :1033–1047, 2009.
- [45] S. G. Ponnambalam, N. Jawahar, and B. S. Girish. An ant colony optimization algorithm for flexible job shop scheduling problem. *New Advanced Technologies*, 2010. Available from : [http ://www.intechopen.com/books/new-advanced-technologies/](http://www.intechopen.com/books/new-advanced-technologies/) Cité le 15 Octobre 2014.
- [46] P. Udhayakumar and S. Kumanan. Integrated scheduling of flexible manufacturing system using evolutionary algorithms. *International Journal of Advanced Manufacturing Technologies*, 61(5-8) :621–635, 2012. DOI : 10.1007/s00170-011-3727-2.
- [47] H. Pierreval and L. Tautou. Using evolutionary algorithms and simulation for the optimization of manufacturing systems. *IIE Transactions*, 29 :181–189, 1997.

- [48] M. Basseur, E.-G. Talbi, A. Nebro, and E. Alba. Metaheuristics for multiobjective combinatorial optimization problems : Review and recent issues, 2006. Institut National de Recherche en Informatique et en Automatique [Research Report] RR-5978.
- [49] K. F. Man, K. S. Tang, and S. Kwong. Genetic algorithms : Concepts and applications. *IEEE Transactions on Industrial Electronics*, 43(5) :519–534, 1996.
- [50] M. Gen. *Springer Handbook of Engineering Statistics : Genetic Algorithms and Their Applications*. Springer, 2006.
- [51] P. R. Srivastava and T.-h. Kim. Application of genetic algorithm in software testing. *International journal of software engineering and its applications*, 3(4) :87–95, 2009.
- [52] M. Ehrgott. *Multicriteria Optimization*. Springer, 2nd. edition, 2005.
- [53] E. Zitzler, M. Laumanns, and S. Bleuler. A tutorial on evolutionary multiobjective optimization. In edited by Springer, editor, *Metaheuristics for Multiobjective Optimisation*, volume 535, pages 3–37. Springer Berlin Heidelberg, 2004.
- [54] C. Coello, G. Lamont, and D. Van Veldhuizen. Evolutionary algorithms for solving multi-objective problems. 2nd edition. In *Genetic and Evolutionary Computation*. Springer, 2007. Chapter 2.
- [55] A. Ghosh and S. Dehuri. Evolutionary algorithms for multi-criterion optimization : a survey. *International Journal of Computing Information Sciences*, 2(1) :38–57, 2004.
- [56] A. Berro. Optimisation multiobjectif et stratégies d’évolution en environnement dynamique, 2001. Thèse Doctorale (Chapitre 2). Université des Sciences Sociales Toulouse I.
- [57] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2 : Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K.C. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EURO-GEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.
- [58] J. M. A. Pangilinan and G. K. Janssens. Evolutionary algorithms for the multiobjective shortest path planning problem. In *International journal of computer and information science and engineering*, pages 54–59, 2007.
- [59] K. Quintero, E. Niel, J. Aguilar, and L. Piétraç. (max, +) optimization model for scheduling operations in a flow network with preventive maintenance tasks. In *Lecture Notes in Engineering and Computer Science : Proceedings of The World Congress on Engineering and Computer Science, WCECS 2013*, volume 2, pages 1036–1041, 2013. 23-25 October.
- [60] Lindo Systems Inc., 2013. www.lindo.com.

- [61] S. Gaubert. *Methods and applications of $(\max, +)$ linear algebra*. Lecture Notes in Computer Science. Springer, 1997.
- [62] K. Quintero, E. Niel, J. Aguilar, and L. Piétrac. Scheduling operations in a flow network with flexible preventive maintenance : A $(\max, +)$ approach. *Engineering Letters*, 22(1) :24–33, 2014.
- [63] E. Cabau. Introduction à la conception de la sûreté, 1999. Cahier Technique N 144.
- [64] W.J. Moore and A.G. Starr. An intelligent maintenance system for continuous cost-based prioritisation of maintenance activities. *Computers in Industry - Special issue : E-maintenance*, 57(6) :595–606, 2006.
- [65] K. Quintero. Capacité de reconfiguration dans les réseaux de pipelines, 2010. Rapport de Stage - Master Recherche. Master GEGP, Laboratoire Ampère. INSA Lyon.
- [66] K. Quintero, E. Niel, J. Aguilar, and Piétrac L. A cost-criticality based $(\max, +)$ optimization model for operations scheduling. In Edited Book by Springer, editor, *Transactions on Engineering Technologies*. Springer, 2015. prévu pour 2015.
- [67] K. Quintero, E. Niel, and J. Aguilar. $(\max, +)$ model for alignment selection and schedule optimization in a flow network. In *12th International Conference of Numerical Analysis and Applied Mathematics*, 2014. Rhodes, Greece.
- [68] C.-L. Lin, C.-H. Huang, and C.-W. Tsai. Structure-specified real coded genetic algorithms with applications. *Advanced Knowledge Based Systems : Model, Applications & Research*, 1 :160–187, 2010.
- [69] Ö. Yeniay. Penalty function methods for constrained optimization with genetic algorithms. *Mathematical and Computational Applications*, 10(1) :45–56, 2005.
- [70] L. Daviaud. Comportements asymptotiques des automates max-plus et min-plus, 2014. Thèse doctorale. Université Paris Diderot (Paris 7) Sorbonne Paris Cité. Laboratoire d’Informatique Algorithmique : Fondements et Applications CNRS.
- [71] T. Colcombet and L. Daviaud. Approximate comparison of distance automata. In *STACS 2013 - 30th International Symposium on Theoretical Aspects of Computer Science*, 2013. 574–585.
- [72] M. Kammoun, Z. Achour, and N. Rezg. Air traffic management using petri net synthesis tools. In *8th International Conference of Modeling and Simulation - MOSIM’10. ‘Evaluation and optimization of innovative production systems of goods and services’*, 2010. Hammamet - Tunisia.