



HAL
open science

Optimisation Heuristique Distribuée du Problème de Stockage de Conteneurs dans un Port

Meriam Kefi

► **To cite this version:**

Meriam Kefi. Optimisation Heuristique Distribuée du Problème de Stockage de Conteneurs dans un Port. Informatique [cs]. Ecole Centrale de Lille, 2008. Français. ⟨NNT : ⟩. ⟨tel-00366467⟩

HAL Id: tel-00366467

<https://theses.hal.science/tel-00366467v1>

Submitted on 7 Mar 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

N° d'ordre : 76

ECOLE CENTRALE DE LILLE

THÈSE

En co-tutelle avec l'Ecole Nationale des Sciences de l'Informatique de Tunis

présentée en vue
d'obtenir le grade de

DOCTEUR

en

Spécialité : Automatique et Informatique Industrielle

par

Meriam KEFI GAZDAR

DOCTORAT DELIVRE PAR L'ECOLE CENTRALE DE LILLE

Titre de la thèse :

Optimisation Heuristique Distribuée du Problème de Stockage de Conteneurs
dans un Port

Soutenue le 23 Juin 2008 devant le jury composé de :

M. Faouzi	GHORBEL	Prof. ENSI	Président
M. Moncef	TAGINA	Prof. ENSI	Rapporteur
M. René	MANDIAU	Prof. UVHC	Rapporteur
M. Khaled	GHÉDIRA	Prof. ENSI	Directeur de thèse
M. Pascal	YIM	Prof. ECL	Directeur de thèse

Thèse préparée dans le Laboratoire LAGIS, Ecole Centrale de Lille
Ecole Doctorale SPI 072

*A mes très chers parents, dont leur témoin éducatif est transmis à leurs enfants ;
A mon cher époux Achraf ;
A mes chers frères ;
A tous les miens.*

Remerciements

J'exprime mes profonds remerciements à Monsieur Khaled Ghédira, professeur à l'ENSI, directeur de cette thèse en Tunisie, responsable du laboratoire SOIE et directeur de l'ENSI qui n'a jamais compté précieux conseils et judicieuses suggestions. Qu'il soit assuré de ma gratitude et de ma reconnaissance pour son soutien durant ces années de recherche.

Je tiens à remercier également Monsieur Pascal Yim, Professeur à l'ECL, qui m'a co-encadré au cours de ces années dans le cadre d'une cotutelle, pour son aide et son soutien inconditionnels. Mes plus vifs remerciements vont également à Monsieur Faouzi Ghorbel, professeur à l'ENSI, pour avoir accepté de présider le jury de cette thèse.

Je tiens à exprimer toute ma reconnaissance à Monsieur René Mandiau, professeur à l'Université de Valenciennes et du Hainaut-Cambrésis, et Monsieur Moncef Tagina, Maître de conférences à l'ENSI, pour avoir accepté de rapporter cette thèse et pour leurs critiques constructives qui ont permis d'améliorer la qualité de ce manuscrit.

J'adresse mes remerciements également à Monsieur Ouajdi Korbaa, Maître de conférences à l'ISITC de Hammam Sousse pour m'avoir suivie de près tout au long de cette thèse.

J'aimerais remercier aussi Monsieur Mohamed Ben Ahmed, Professeur émérite à l'ENSI d'avoir apporté des jugements très constructifs permettant l'amélioration du manuscrit et je lui en suis reconnaissante.

Mes remerciements vont aussi à l'Office de la Marine Marchande et tout particulièrement Monsieur Foued Ben Othman, directeur de l'OMMP et Madame Olfa Elphil pour s'être intéressés à mon travail et m'avoir aidé à accomplir mon étude expérimentale au sein du port de Radès.

Je remercie Mme Safya Belghith, Professeur à l'ENIT et Directrice de l'ESTI de Tunis pour m'avoir aidé à mener à bien ce travail.

Mes remerciements vont à mes collègues de SOIE, LI3, ENSI, ESTI et LAGIS avec qui j'ai passé des moments inoubliables. J'en garderai de bons souvenirs.

Merci à mes parents et mes deux frères pour leur soutien moral et l'intérêt qu'ils ont toujours porté à ce que je faisais ...

Merci à Achraf pour avoir subi plus que les autres ma thèse et pour sa présence inestimable à mes cotés.

Ma profonde gratitude à tous les membres de ma famille, et particulièrement une pensée à tous ceux qui me sont chers et qui ne sont plus parmi nous.

Il faut faire de la vie un rêve et d'un rêve une réalité.

Pierre Curie

Résumé

Les terminaux à conteneurs constituent des interfaces inter-modales essentielles pour le réseau de transport mondial. Une manutention efficace des conteneurs dans des terminaux est d'une importance cruciale pour la réduction des coûts de transport et la détermination des plans d'embarquement. Dans ce rapport de thèse, nous proposons principalement une approche de résolution distribuée à travers la description d'un modèle d'optimisation heuristique distribuée baptisé COSAH (*C*Ontainer *S*tacking via *multi-Agent* approach and *H*euristic method) qui permet de simuler, résoudre et optimiser l'espace de stockage disponible pour manier les départs et les arrivées des conteneurs dans un port fluvial ou maritime. Autrement dit, COSAH permet de minimiser le nombre total de mouvements parasites tout en respectant des contraintes dynamiques d'espace et de temps.

Les performances de COSAH sont ensuite évaluées sur des instances générées aléatoirement, ainsi que des instances extraites de la réalité d'un port maritime tunisien : le port de Radès. En effet, nous avons procédé à une étude expérimentale implémentant et comparant COSAH à la version centralisée associée, toutes deux basées sur un algorithme de recherche non informée et un algorithme de recherche informée. Les résultats obtenus, présentés et illustrés, montrent l'efficacité de COSAH en particulier, et d'une méthode d'optimisation heuristique distribuée alliant les deux concepts : Agent et Heuristique, en général.

Mots clés : approche multi-agent, éco-résolution, méthode heuristique, réseaux de Petri colorés, résolution distribuée de problèmes, stockage de conteneurs, transport conteneurisé.

Abstract

Container terminals are essential intermodal interfaces in the global transportation network. Efficient container handling at terminals is of vital importance for the reduction of transportation costs as well as the determination of shipping schedules. In this thesis report, we propose a distributed solving approach through the development of a distributed *Heuristic-Optimization model denoted COSAH (C*Ontainer *S*tacking via *multi-Agent* approach and *H*euristic method) that allows to simulate, solve and optimize the amount of storage space for handling container departures and arrivals within a fluvial or maritime port. COSAH allows to minimize the expected total number of rehandles while respecting spatio-temporal constraints. Performances of COSAH are then evaluated based on random-generated instances, as well as on instances extracted from a real-world case study : the port of Radès. Indeed, we conducted an experimental study implementing and comparing COSAH to the associated centralized version based on non informed-search algorithm and an informed-search algorithm. The obtained results, presented and illustrated, evince the efficiency of COSAH in particular, and of a distributed heuristic-optimization method combining two concepts : Agent and Heuristic, in general.

Keywords : colored Petri nets, container stacking, containerized transport, distributed problem-solving, eco-solving, heuristic method, multi-agent approach.

Table des matières

Table des matières	1
Table des figures	5
Liste des algorithmes	7
Introduction Générale	9
I État de l’art	15
1 Transport Conteneurisé	17
1.1 Introduction	17
1.2 Le transport	18
1.3 Le transport maritime	19
1.3.1 Le transport de marchandises	19
1.3.2 Les trois révolutions du transport maritime	21
1.4 Le transport fluvial	22
1.5 La conteneurisation	23
1.5.1 La révolution du conteneur	23
1.5.2 1956, naissance d’un concept	24
1.5.3 Le principe de la standardisation	24
1.5.4 Manutention et expéditions	25
1.5.5 Les opérations dans un terminal à conteneurs	26
1.5.6 Les systèmes d’information	26
1.5.7 La multimodalité <i>Multimodal Transport</i>	28

1.5.8	Avantages et Inconvénients de la conteneurisation	29
1.6	Les ports conteneurisés	30
1.7	Conclusion	31
2	Problème de Stockage de Conteneurs (PSC)	33
2.1	Introduction	33
2.2	Processus dans un terminal à Conteneurs	34
2.2.1	Planification de navires	35
2.2.2	Transport sur le quai <i>Quay Transport</i>	39
2.2.3	Ordonnancement des équipements de manutention	41
2.3	Problème de Stockage de Conteneurs -PSC <i>Container Stacking Problem</i>	44
2.3.1	Définition du PSC	47
2.3.2	Formulation du PSC	47
2.3.3	Problèmes analogues	51
2.3.4	Méthodes existantes pour la résolution du PSC	52
2.4	Classification des travaux selon les ports conteneurisés dans le monde	59
2.5	Bilan et synthèse	59
II	Approche proposée	63
3	Méthodes et approches adoptées	65
3.1	Introduction	65
3.2	Les systèmes multi-agent	66
3.2.1	Pourquoi avons-nous opté pour les systèmes multi-agent ? .	66
3.2.2	Quelques concepts	68
3.2.3	Résolution par coordination : l'éco-résolution	75
3.2.4	Formalisation des Systèmes Multi-Agent	77
3.3	Les réseaux de Petri	81
3.3.1	Définition d'un réseau de Petri	81
3.3.2	Réseaux de Petri colorés et la plate-forme CPN-Tools . . .	82
3.4	Optimisation heuristique	83
3.4.1	Que faire face à un problème NP-dur ?	83

<i>Table des matières</i>	3
3.4.2 Classes d'heuristiques	86
3.5 Conclusion	90
4 Le modèle COSAH : Un modèle d'Optimisation Heuristique Distribuée du PSC	91
4.1 Introduction	91
4.2 Modèle de base : ContEco	92
4.2.1 Description de la version relaxée du PSC : PSCR	93
4.2.2 Description de ContEco	95
4.3 Le modèle COSAH	101
4.3.1 Description du problème étudié	102
4.3.2 Description du modèle COSAH	104
4.4 Conclusion	132
5 Validation Expérimentale	135
5.1 Introduction	135
5.2 Implémentation	136
5.2.1 Critères de choix	136
5.2.2 Présentation de JADE	137
5.3 Expérimentations	138
5.3.1 Instances	138
5.3.2 Résultats numériques	138
5.4 Cas d'étude : Port de Radès	145
5.4.1 Présentation du port	146
5.4.2 Expérimentations	149
5.5 Conclusion	150
Conclusion Générale	151
Bibliographie	167

Table des figures

1.1	Le <i>turnover</i> mondial de conteneurs (en million EVP)	21
2.1	Succession des opérations dans un terminal à conteneurs	35
2.2	Types d'équipements de manutention	44
2.3	Illustration de la représentation d'un état	48
2.4	Modèle incertain d'optimisation à plusieurs étapes	50
2.5	Exemple de manipulation des cubes	52
2.6	Analogie entre le PSC et le problème du monde des cubes	53
3.1	Evolution des paradigmes de programmation	68
3.2	Tir d'une transition sensibilisée	82
4.1	Schéma Conceptuel de l'approche proposée	92
4.2	Configuration initiale et configuration finale à atteindre	95
4.3	Composants d'un agent Conteneur et ses interactions	96
4.4	Module d'autonomie d'un agent Conteneur	99
4.5	Module de réception d'un agent Conteneur	99
4.6	Message " Tu_me_gênes (G) "	100
4.7	Message " Chercher_Place (CP) "	100
4.8	Message " Sur_Toï (ST) "	101
4.9	Message " Tu_es_Libre (L) "	101
4.10	Composants du module de prise de décision d'un agent Conteneur	124
5.1	Comparaison DistNIS par rapport à CentNIS	139
5.2	Comparaison de DistIS par rapport à CentIS	140
5.3	Comparaison de CentIS par rapport à CentNIS	141

5.4	Comparaison de DistIS par rapport à DistNIS	142
5.5	Mesure de performance de COSAH par rapport à la borne maximale	143
5.6	Exemple de calcul de la borne maximale du nombre de mouvements	144
5.7	Sensibilité de DistIS et de DistNIS à la hauteur moyenne	145
5.8	Mesure de performance de COSAH par rapport à la politique actuelle du port	150

Liste des Algorithmes

4.1	Comportement global des agents Conteneur	110
4.2	Comportement de départ des agents Conteneur A	111
4.3	Messages échangés entre les agents Conteneur	112
4.4	Comportement des agents Conteneur recevant le message gêné() .	113
4.5	Comportement des agents Conteneur perturbés	113
4.6	Comportement des agents Conteneur recevant le message Cher- cherPlace()	113
4.7	Comportement des agents Conteneur recevant le message JeSuis- Libre()	114
4.8	Comportement des agents Conteneur recevant le message Désolé()	115
4.9	Comportement des agents Conteneur recevant le message Migrer ()	116
4.10	Comportement des agents Conteneur recevant le message Confir- mer ()	116
4.11	Comportement des agents Conteneur recevant le message Rejeter ()	117
4.12	Comportement des agents Conteneur recevant le message TuEs- Libre ()	117
4.13	Traitement des arrivées à l' $i^{\text{ème}}$ départ	119
4.14	Algorithme d'optimisation heuristique	131

Introduction Générale

Cette introduction a pour objectif de définir notre problématique de recherche et de présenter le positionnement scientifique de notre démarche pour y répondre :

Problématique

Le transport fluvial et maritime devient, de nos jours, de plus en plus important et représente une alternative crédible et intéressante au transport terrestre et aérien. L'interdépendance entre commerce et flux de biens et de services, en voie d'évolution continue, fait que ce système de transport représente une préoccupation d'une importance cruciale. En effet, les différentes compagnies et entreprises fournissent de grands efforts dans le but de détecter des possibilités de réduction des coûts associés à ce transport. Ces coûts sont affectés par les temps morts dans le cas de navires improductifs immobilisés dans un port, et aux échanges entre les différents modes de transport dans les ports (terrestre, ferroviaire ou navigable) dans le cas de transport multimodal. C'est donc la recherche à les réduire au minimum afin d'écourter la distance navire-zone de stockage et d'équilibrer les deux cycles : le cycle Bord (Grue de quai-quai) et le cycle Terre (quai-véhicules-zone de stockage). L'une des solutions retenues entre autres, est la conteneurisation qui s'est véritablement généralisée pour le transport maritime et fluvial des marchandises solides sur toutes les mers du globe. Cette solution a permis de réaliser des économies via la standardisation des dimensions, la sécurité des marchandises, l'accélération des opérations de manutention favorisant l'automatisation du chargement (ou gerbage) et du déchargement (ou enlèvement) et l'atténuation des ruptures de charge lors des transbordements d'un mode de transport à l'autre.

La conteneurisation peut induire cependant des coûts élevés liés aux investisse-

ments en espace de stockage, à l'approvisionnement et entretien des équipements, aux temps morts ou mouvements inutiles durant le processus déchargement-stockage-chargement des conteneurs et au déséquilibre du flux de marchandises imposant des transports de conteneurs vides.

Toutefois, pour être compétitif, le transport fluvial ou maritime doit améliorer ses services et baisser ses coûts. De ce fait, plusieurs interrogations s'imposent : Comment peut-on assurer un meilleur cheminement de marchandises ? Comment peut-on minimiser les retards ?, Comment doit-on procéder afin d'augmenter les capacités de stockage ? etc. Toutes ces questions se posent au niveau des différentes activités s'exerçant dans un port à conteneurs, et constituent des objectifs soucis des théoriciens et praticiens tel qu'ils sont décrits dans [SVS04, GG06].

Dans le contexte d'un port fluvial ou maritime devant gérer un trafic de conteneurs au croisement de voies terrestres, ferroviaires et fluviales/ maritimes et également responsable de la gestion d'un stock de conteneurs sur son site, la gestion des quais constitue un ensemble de problèmes de décision à résoudre pour faire face à un afflux important de conteneurs et une augmentation de la taille des navires. On est mené soit à augmenter la taille des ports, alternative coûteuse en terme de ressources spatiales et matérielles mais facile car elle ne demande aucun effort d'optimisation, soit à améliorer la productivité, solution plutôt raisonnable mais demandant de grands efforts de réflexion à l'échelle pratique et théorique.

Dans notre cadre, améliorer la productivité revient à minimiser le temps passé par un conteneur ou un navire dans un port que ce soit en import ou en export.

Néanmoins, l'optimisation de la gestion et de la logistique des conteneurs dans un port et plus précisément la gestion fine de placement des conteneurs apparaît comme un problème à part entière avec ses difficultés propres jouant un rôle prépondérant dans la minimisation des temps morts. Ce problème est connu dans la littérature par *Container Stacking (Storage ou encore Handling) Problem*. Il sera désigné dans la suite par l'acronyme PSC : " Problème de Stockage de Conteneurs ". Il s'agit de la manutention des conteneurs en import ou en export qui peut conduire à de nombreux déplacements ou remaniements inutiles, consistant à dépiler un nombre de conteneurs pour atteindre un conteneur bien déterminé placé en dessous de la pile. L'objectif donc est de réduire ces déplacements appelés mouvements parasites ou remaniements *Shifting moves/ unproductive moves*. Ce

problème est modélisé par un système complexe, dynamique et incertain dans lequel les interactions entre ses différents éléments (conteneurs, navires, ressources matérielles, etc.) sont en perpétuelle évolution (arrivées imprévues de conteneurs, départs estimés pour les conteneurs en export, changements continus de variables) nécessitant, de ce fait, la remise en question de sa structure et de sa dynamique.

En résumé, la problématique définie ci-dessus s'énonce par : Comment développer une méthode capable de modéliser, simuler, résoudre et optimiser le processus de stockage de conteneurs dans un port ?

Objectifs

Ainsi décrit et par analogie aux problèmes classiques de planification, le problème de stockage de conteneurs PSC est considéré NP-difficile et NP-complet. Une solution optimale de ce problème est un plan d'actions de dépilement/empilement de conteneurs de longueur minimale, c'est-à-dire constitué d'un nombre de mouvements de déplacements minimal, tenant compte des contraintes dynamiques que ce soit en terme d'espace ou de temps. Les interactions, les collaborations et les dynamiques qui en découlent font que le PSC est modélisé par un système complexe d'une part, c'est à dire constitué de nombreux composants en interaction dynamique entre eux et avec le monde extérieur ; et d'autre part ouvert, c'est à dire capable de maintenir ses fonctions malgré les arrivées et départs de ses composants. Peu de travaux de recherche lui ont été consacrés. En effet, nous avons constaté que ces travaux utilisent pour la plupart des modèles analytiques (les modèles mathématiques et la programmation dynamique) régis par des lois stochastiques, et des modèles de simulation comme ceux qui font appel à l'approche orientée objet ou l'approche multi-agent. Nous retrouvons aussi des méthodes incomplètes tels que les algorithmes génétiques qui s'intéressent à minimiser le temps passé par un navire au quai d'un port de terminaux à conteneurs. Ces modèles s'avèrent très limités face à la dynamique et la nature combinatoire de ce problème.

Étant donné ses multiples aspects, il serait intéressant d'aborder globalement et en profondeur cette problématique dans cette thèse. Nous aurons recours à deux notions fondamentales : Les Systèmes Multi-Agent, et les Heuristiques. D'une

part, les systèmes multi-agent connaissent actuellement un certain engouement, et constituent une solution proposée par la communauté de l'intelligence artificielle distribuée qui semble adaptée à de nombreux problèmes en raison de leur capacité à aborder les systèmes complexes et ouverts.

D'autre part, les heuristiques permettent de fournir des solutions réalisables de qualité acceptable souvent très proches de l'optimal et déterminées en un temps raisonnable (complexité polynomiale). Elles sont situées à mi-chemin entre les méthodes exactes de recherche opérationnelle qui se limitent à la nature combinatoire de ces problèmes rendant difficile la résolution des problèmes réels tels que le PSC et les métaheuristiques dont l'inconvénient relève de leur sensibilité aux différents problèmes d'optimisation combinatoire vu qu'elles nécessitent un contrôle précis des paramètres d'exécution et souvent des changements dynamiques de ces paramètres pour s'adapter à chaque problème.

Notre travail consiste alors à associer les concepts de systèmes Multi-agent et de l'optimisation heuristique, et ce en proposant un modèle d'optimisation heuristique distribuée baptisé COSAH pour *COntainer Stacking based on multi-Agent approach and Heuristic method* qui allie d'une part, les caractéristiques des systèmes Multi-agent pour la prise de décision, la communication et la coopération inter agents à l'image de conversation humaine et d'autre part, la mise en face d'une certaine "intelligence" supportée par des heuristiques. Nous pensons que la combinaison des deux notions : distribution et heuristique qui, séparément, fournissaient de bons résultats, représente une potentialité d'amélioration intéressante.

Dans un contexte bien déterminé, selon des motivations et des enjeux que nous allons définir, et considérant des hypothèses de travail que nous allons décrire, le modèle COSAH doit optimiser le placement des conteneurs dans la zone de stockage (statique ou dynamique selon les départs ou les arrivées de navires ou camions ou de conteneurs en import ou en export) en minimisant le nombre de mouvements improductifs respectant un nombre de contraintes spatio-temporelles et tenant compte de plusieurs paramètres pouvant catégoriser les conteneurs et définir une certaine priorité entre eux (destination, taille, date de départ, horizon des arrivées des barges ou navires associés).

Organisation du Rapport

L'organisation de cette thèse suit une progression ordonnée. Ce document se décompose en deux parties correspondant au cheminement de notre démarche scientifique.

Partie 1 : État de l'art

La première partie expose notre contexte de travail dans deux chapitres. Le premier chapitre présente le transport conteneurisé, contexte de notre problématique, et définit les différentes terminologies et concepts intrinsèques. Le deuxième chapitre survole, en premier lieu la littérature des problèmes de décision rencontrés et découlant de l'activité et de la logistique des conteneurs sur un terminal à conteneurs. En deuxième lieu, ce chapitre se concentre sur le problème étudié : le PSC en faisant un tour d'horizon des problèmes analogues, et des méthodes de résolution qui lui sont dédiées. Cela permet de souligner la complexité, la dynamique et l'ouverture caractérisant le PSC.

Cette partie nous permet ainsi de nous positionner par rapport à notre problématique et de spécifier nos orientations et nos choix qui serviront de fondements à notre proposition.

Partie 2 : Approche Proposée

La deuxième partie sera dédiée à l'approche de résolution distribuée que nous proposons pour la résolution et l'optimisation du problème de stockage de conteneurs. Elle s'articule autour de trois chapitres : le premier chapitre présente d'abord les concepts clés de notre réflexion : le paradigme Multi-agent, le formalisme réseaux de Petri colorés et les méthodes heuristiques, afin de définir quelques notions que nous retrouvons au fil de cette thèse et de justifier des choix que nous adoptons dans notre proposition.

Le deuxième chapitre présente notre proposition constituée de deux contributions : D'abord, une première contribution relevée par un modèle multi-agent de base baptisé ContEco traitant une version relaxée du PSC désignée par PSCR. Ce modèle est conçu et développé par analogie au modèle d'éco-résolution développé pour le problème de planification classique " le monde des cubes ". En outre, ce

chapitre expose une formalisation de ce modèle utilisant la méthode formelle : les réseaux de Petri colorés.

Ensuite, une deuxième et principale contribution relevée par le modèle COSAH. Ce chapitre analyse en détails l'architecture, les connaissances de base, et la dynamique des agents mis en jeu. Il s'attarde sur la méthode heuristique mise en oeuvre dans ce modèle ainsi que sur les étapes attractives du comportement des agents.

Le troisième chapitre valide notre modèle COSAH en procédant à une étude empirique et comparative implémentant et simulant celui-ci, et à des expérimentations visant à évaluer ses performances via des jeux de tests générés aléatoirement. Il propose un prototype s'appuyant sur la plate-forme multi-agent JADE et intégrant toutes les particularités inhérentes à notre problème et au modèle COSAH. Il décrit en outre l'environnement de l'implémentation, les paramètres d'expérimentations et les résultats obtenus. Il vérifie enfin la validité de notre proposition utilisant un jeu de tests réels dans le cas d'un port maritime : le port de Radès en Tunisie.

Enfin nous concluons notre travail tant au niveau des apports que des ouvertures envisagées.

Première partie

État de l'art

Chapitre 1

Transport Conteneurisé

1.1 Introduction

Indubitablement, le transport joue un rôle capital au sein de l'économie de par son omniprésence dans la chaîne de production et ce, à toute échelle géographique. Le transport se conçoit comme une composante intégrale du cycle de production-consommation. En fait, bien souvent les entreprises et les individus doivent prendre des décisions sur les routes à emprunter afin d'acheminer du fret ou des individus à travers l'espace économique. Il n'est pas exceptionnel de constater que les coûts de transport peuvent représenter 20% du prix total d'un bien. Le choix du mode de transport pour acheminer de la marchandise ou des individus d'une origine à une destination devient grand dans un contexte marqué par la production de biens de consommation légers et par des techniques de production de plus en plus avancées. Ce choix dépend d'un nombre de facteurs telles que la nature des biens, les infrastructures disponibles, les origines et les destinations.

Les modes de transport sont : le transport terrestre, le transport fluvial ou maritime et transport aérien. Ils représentent une part importante de la géographie des transports en tant que pourvoyeurs de mobilité.

Le transport maritime ou fluvial a été traditionnellement un vecteur de commerce et de progrès très fortement concurrencé aujourd'hui par le chemin de fer et la route. Comparées à d'autres modes de transport, les voies navigables offrent

plusieurs avantages sur lesquels les constructeurs ont su capitaliser. Leur coût est faible, ramené à la tonne-kilomètre. En effet, 25000 milliards de Tonnes-Km de fret parcourent les océans annuellement comparés à 7000 pour le rail et 3 000 pour la route. Il s'agit de pas moins de 71% de tout le fret mondial transporté. Les dernières années, les compagnies leaders de ce mode de transport peu consommateur d'énergie ont su le fiabiliser et l'adapter aux grands volumes de fret. Comme les modes aérien et terrestre, le transport maritime évolue sur son espace propre : un espace à la fois géographique par ses attributs mais aussi stratégique par son rôle.

Les plus récentes avancées technologiques touchant au transport par voie navigable ont cherché à modifier les écluses et les canaux, à augmenter la taille, à automatiser et spécialiser les navires (porte-conteneurs, vraquier, navire-citerne etc.), et à déployer la conteneurisation du transport. Ces transformations sont à l'origine en partie, du développement d'un trafic maritime attentif à la demande accrue en combustibles ainsi qu'à la géographie des grands marchés céréaliers. Dans ce qui suit, nous proposons une définition profonde des notions ci introduites, s'avérant nécessaire pour délimiter le cadre général de la problématique soulevée dans ce travail. Nous nous référons à Steenken et al. [SVS04] et à des statistiques extraites des notes d'ISEMAR [NS002, NS003].

1.2 Le transport

Le transport est le déplacement de personnes ou de biens d'un endroit à un autre. Les transports modernes constituent un système. Chaque sous-système (selon le mode de transport) est constitué d'une infrastructure (linéaire pour les transports terrestres et ponctuelle pour les transports maritimes et aériens), de véhicules (individuels ou regroupés en rames) ou de flux continus (pour les transports par conduites : gazoducs, oléoducs) et de techniques d'exploitation particulières.

De nos jours, le transport de personnes (voyageurs) et le transport de marchandises (fret), plus ou moins confondus jusqu'à une époque récente, constituent deux systèmes de plus en plus indépendants, même s'ils utilisent parfois les mêmes infrastructures et plus rarement les mêmes véhicules. L'ensemble des opérations

de transport de fret, ainsi que tous les services impliqués dans la réception, la livraison et la manutention des biens pour que ceux-ci soient livrés au moment voulu chez le destinataire constitue la logistique.

1.3 Le transport maritime

Associée aux transports terrestres que furent le portage (à dos d'animal ou d'homme) et le roulage (voies romaines), la navigation maritime constitua le premier système de transport. Elle favorisa, en premier lieu, les littoraux situés sur les mers les plus propices au trafic maritime à courte distance (cabotage), particulièrement de l'Asie du Sud-est, et plus encore celui de la Méditerranée et de l'Europe du nord-ouest. C'est donc d'abord de ces mers fermées que partit le développement des transports. La navigation maritime connut un premier saut technologique avec l'apparition au XIXe siècle de la machine à vapeur. Celle-ci permit en effet d'accélérer les transports sur l'eau en s'affranchissant des contraintes naturelles (vents et courants) pesant sur les routes maritimes.

L'avènement du commerce électronique et des flux dématérialisés ne sonne pas le glas des transports internationaux. La conclusion en ligne de contrats portant sur des biens meubles accompagnera, au contraire, le développement de la fonction logistique et des transports internationaux.

Les sociétés de transport rapide sont ainsi indispensables aux opérateurs du commerce électronique pour assurer la livraison des produits commandés sur les réseaux. En amont de la relation entre le distributeur et le client final, l'activité économique repose sur des échanges entre des partenaires éloignés et le transport maritime s'avère alors indispensable. Une augmentation du tonnage et du trafic maritime accompagne ainsi la croissance de la production mondiale. Cela vaut bien sûr pour le transport de pondéreux et de matières premières, mais au-delà pour les biens de consommation et l'ensemble de la production industrielle.

1.3.1 Le transport de marchandises

Les navires appartiennent généralement à des armateurs. Ceux-ci peuvent exploiter directement leurs navires sur des lignes régulières ou au contraire les louer

à l'affrètement sous la forme du *tramping*. Afin de réduire les coûts, les navires sont de plus en plus spécialisés à un type de cargaison (pétrolier, méthanier, bananier, minéralier, vraquier, porte-conteneurs, etc.). Ils sont aussi de plus en plus grands, de manière à transporter des quantités de plus en plus importantes, ce qui pose le problème du changement d'échelle des volumes transportés sur terre et sur mer : 3 000 tonnes au plus par train ou convoi fluvial, 100 à 200 fois plus sur mer. Les plus récents porte-conteneurs peuvent emporter quelque 6 000 conteneurs qui, une fois à terre, nécessiteront 6 000 camions ou 30 trains complets ! Ce problème ne peut être résolu que par la capacité de stockage des installations portuaires ou bien par la transformation des matières importées sur place dans les zones industrialo-portuaires.

Un navire est improductif lorsqu'il est immobilisé dans un port. La réduction de ces temps morts a d'abord été obtenue par l'accélération des opérations de manutention grâce à l'unification des charges qui permet l'automatisation du chargement et du déchargement. D'autre part, dans le transport maritime, une large partie des coûts est affectée aux échanges dans les ports. On a donc cherché à les réduire au minimum en supprimant les ruptures de charge de la marchandise entre les différents modes de transport.

L'une des solutions retenues est la spécialisation des navires et des postes d'amarrage, réservés à un seul type de cargaison (*terminal offshore* et *sea-line*). Les porte-barges *Lighter Aboard Ship-LASH* permettent de rentrer des barges fluviales dans un navire de mer, la manœuvre pouvant s'opérer au large des côtes. Le transroulage (*Ro-Ro : Roll on-Roll off*), qui consiste à embarquer les semi-remorques directement avec leur tracteur sur le navire *ferry*, s'est généralisé sur les traversées courtes de l'Amérique du nord, de l'Extrême-Orient, et surtout de l'Europe.

Enfin, la conteneurisation (*Lo-Lo : Lift on-lift off*) s'est véritablement généralisée pour le transport maritime des marchandises solides sur toutes les mers du globe. Ces dernières années, l'utilisation de conteneurs pour le transport maritime intercontinental a augmenté. La figure 1.1 extraite de [Tur] montre la croissance du *turnover* mondial de conteneurs. Commencer avec 50 million EVP en 1985, ce *turnover* a atteint plus de 350 million EVP en 2004. Une croissance continue est attendue dans les prochaines années, surtout entre l'Asie et l'Europe.

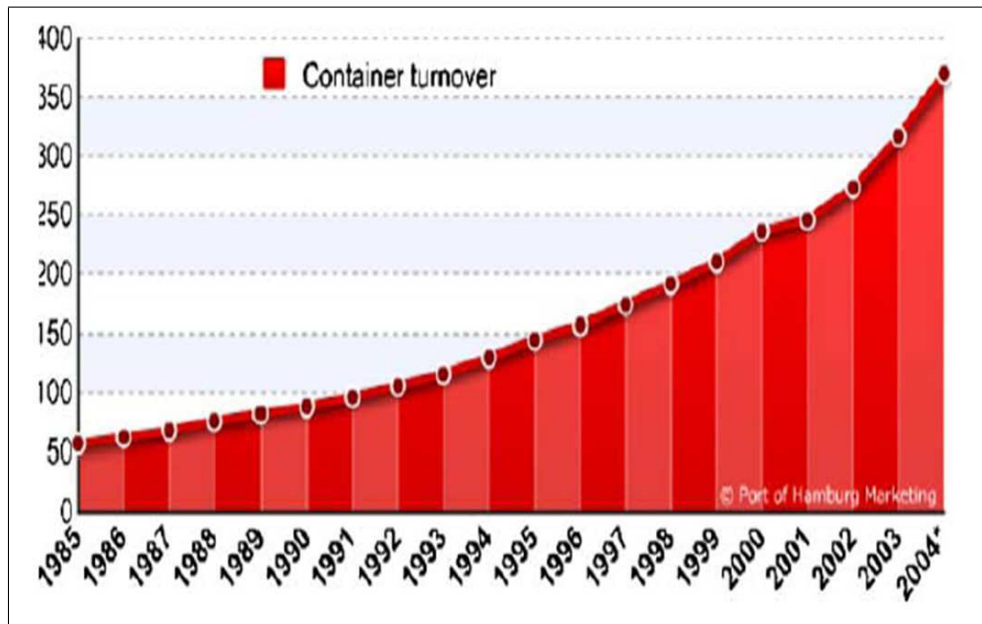


FIG. 1.1 – Le *turnover* mondial de conteneurs (en million EVP)

Cependant, cette croissance nécessite de grandes surfaces de stockage et des équipements très lourds de manutention : les portiques transconteneurs. Ces équipements ne prennent place qu'en certains points (grands ports et portes des grandes agglomérations) qui deviennent ainsi de véritables plates-formes multimodales de redistribution.

1.3.2 Les trois révolutions du transport maritime

Le secteur du transport maritime qui est un " instrument " privilégié du commerce international, a lui-même connu plusieurs révolutions qui ont concouru à une réduction des coûts du transport.

1.3.2.1 Première révolution

Cette révolution, s'est faite sur l'initiative d'un opérateur américain, Malcom Mc Lean, à la fin des années 1950. Il a donné naissance à la " conteneurisation ". Le transport des marchandises dans des conteneurs constitue une forme de standardisation dans le secteur maritime. Les conteneurs constituent des unités

standards qui peuvent faire l'objet d'un chargement et d'un déchargement plus rapide. Le stockage est également plus facile pendant les périodes de transit.

1.3.2.2 Deuxième révolution

C'est celle du transport multimodal. Pour arriver à une destination, la marchandise transportée dans un conteneur, empruntera successivement plusieurs moyens de transport : navire, route, rail, etc.

1.3.2.3 Troisième révolution

Actuellement en marche, c'est celle de l'informatisation et de l'automatisation. L'informatisation permet l'accélération de la circulation des documents de transport maritime et la réduction des coûts traditionnellement associés. Au-delà, elle permet au transporteur de développer des services personnalisés comme le suivi des marchandises au cours du voyage.

1.4 Le transport fluvial

Principal mode de transport préindustriel, l'utilisation de la voie d'eau s'est largement développée avec la construction des grands canaux, d'abord latéraux aux rivières, puis de jonction inter bassins. C'est dans ce cadre que se multiplièrent les ouvrages d'art : innombrables des écluses, dont certaines parfois groupées en escalier, des souterrains, ou des ponts-canaux. D'autres ouvrages, plus élaborés, furent conçus ultérieurement, tels que les ascenseurs à navires, les écluses à grand gabarit, ou encore les plans inclinés.

Ce réseau important (qui cumula 12 780 km au XIXe siècle, mais qui n'en compte plus que 8 000 km actuellement) et relativement dense, fit l'objet d'une première phase de modernisation au XIXe siècle dans le cadre du plan Freycinet destiné à unifier l'ensemble des voies navigables au gabarit de 350 tonnes. À partir du début du siècle, le halage hippomobile fit souvent place au tracteur électrique, avant d'être lui-même détrôné par l'introduction de la péniche automotrice.

Dans la seconde moitié du XIXe siècle, le chemin de fer, disposant d'un réseau plus dense et offrant une plus grande rapidité de transport, exerça une concurrence

victorieuse sur le transport fluvial, souvent au moyen d'une politique tarifaire très agressive. Dans ce contexte, le réseau de voies navigables demeura très largement à l'écart de toute modernisation dans la première moitié du XXe siècle, au fur et à mesure que s'imposait le monopole du chemin de fer.

Dans cette même période, les techniques de transport fluvial évoluaient rapidement : accroissement du gabarit (3 000 tonnes aux normes internationales), utilisation de convois poussés, puis de barges porte-conteneurs et de navires fluvio-maritimes. Cette mutation exigeait une modernisation profonde des voies navigables (chenalisation des fleuves, canalisation des rivières, réduction du nombre et agrandissement notable des écluses, élévation des ponts pour respecter le tirant d'air).

1.5 La conteneurisation

La conteneurisation s'est développée durant ces vingt dernières années. Le conteneur est un moyen de transport qui a permis de réduire les coûts et les délais grâce à la standardisation. Dans cette section, nous voyons de plus près les concepts liés à la conteneurisation, ses composants, ses avantages et ses inconvénients.

1.5.1 La révolution du conteneur

En moins de 50 ans, ce dernier s'est imposé comme le premier moyen d'échange de biens de consommation à l'échelle mondiale entraînant une véritable révolution dans les transports mondiaux. En 1960, la rotation d'un cargo de ligne de 10 000 tonnes de capacité, déployée sur le trajet Europe-Japon-Europe prenait cinq mois. Près de la moitié du temps était passée au port, avec des escales atteignant parfois quatre ou cinq jours. En 2000, un grand porte-conteneurs offre une capacité de 60 000 tonnes et boucle le même trajet en deux mois avec des escales qui durent d'une dizaine à 36 heures. En même temps, l'automatisation a fait son chemin : le cargo fonctionnait avec un équipage de 35 hommes alors qu'aujourd'hui les porte-conteneurs n'ont plus besoin que d'une quinzaine de marins pour naviguer. Plus rapides, plus sûrs, plus performants, les transports maritimes sont aussi moins

coûteux : l'acheminement d'un conteneur chargé de 400 téléviseurs entre Taïwan et le Havre coûte 3000 EUR environ. Le prix de ce transport aurait coûté au moins trois fois plus cher dans les années 1960.

1.5.2 1956, naissance d'un concept

Le transport maritime conteneurisé est né sous l'impulsion d'un entrepreneur américain Malcolm Mac Lean qui, en 1956, adapta 4 navires pour transporter des remorques de camions par voie maritime. En 1956, l'Ideal X reliait New York à Houston avec 58 remorques à son bord. L'expérience se révélant positive, Mac Lean franchit véritablement le pas en conteneurs manutentionnés qui sont identiques dans leur conception.

1.5.3 Le principe de la standardisation

Un conteneur *container*, est une " boîte " rectangulaire de dimension universelle : la clé de son succès réside dans sa standardisation. Les conteneurs secs *dry* de vingt et quarante pieds de longueur (environ six et douze mètres) sont les plus utilisés. Ils servent au transport des marchandises dites sèches, conditionnées en caisses, cartons, balles, palettes, etc. Mais d'autres conteneurs plus spécifiques ont été créés comme les conteneurs-citernes *tank container* et les plein-ciel *open top*, les conteneurs frigorifiques *reefer*. Le conteneur standard de vingt pieds sert d'unité de référence pour estimer les capacités d'un navire et évaluer les flux. On parle alors en Equivalent Vingt Pieds-EVP *Twenty Equivalent Unit-TEU* ce qui correspond à un volume utile de $33 m^3$.

Chaque conteneur est identifié par une série d'inscriptions permanentes sur ses parois (propriétaire, numéro d'immatriculation, masse brute maximale, tare et charge utile). Presque toute la vaste gamme des marchandises transportées jadis comme marchandises non groupées peuvent faire actuellement l'objet d'un transport par conteneur. Ce sont essentiellement des biens d'équipement et de consommation plus ou moins élaborés qui empruntent cette voie, mais le conteneur s'ouvre également aux produits en vrac si ce choix présente des avantages : logistique facilitée pour les petits lots de céréales, les opérations de transvasement réduites pour les produits chimiques, la possibilité de services porte à porte pour

les fruits, etc.

Concluons que la spécialisation des sites de production à l'échelle planétaire, la recherche des moindres coûts de production, la tendance d'assurer un service rapide, fiable et de qualité supérieure, de même que l'appel croissant à la sous-traitance et la spécialisation des unités de production ont entraîné une augmentation des échanges entre usines et industries, et entre producteurs et consommateurs. La simplification des circuits d'échanges effectués par les gestionnaires et les logisticiens passe bien souvent par l'utilisation du conteneur, interchangeable (rapidité de déplacement), repositionnable, multimodal (intégration facile de tous les modes de transport). Le conteneur a rendu possible la fabrication mondiale, dont la croissance a, à son tour, accéléré le développement du conteneur et en a fait le moyen de transport préféré et un auxiliaire de la production industrielle.

1.5.4 Manutention et expéditions

Deux termes définissent le chargement et le déchargement de la marchandise d'un conteneur : l'empotage et le dépotage. Tout ou presque peut se côtoyer dans un conteneur : des bouteilles d'huile avec des planches à voile, des tétines de biberons avec des caméras vidéo, etc. ; Toutefois le bon sens interdira l'empotage de produits dangereux avec des denrées alimentaires. Mais, il existe des cas de contaminations moins évidents, comme un chargement de carottes contaminées par l'odeur des oignons présents dans le même conteneur. L'empotage des marchandises dangereuses est quant à lui soumis à une réglementation stricte, contenue dans le code *IMDG (International Maritime Dangerous Goods)* et une signalétique spécifique est apposée sur le conteneur en fonction de la nature du produit. Il existe deux possibilités d'expédier les marchandises : soit par conteneur complet *Full Container Load-FCL*, c'est la solution la plus répandue où toute la marchandise appartient à un même client qui loue la boîte. Soit par groupage maritime *Less than Container Load-LCL*, cette méthode est utilisée pour les petits envois (1 à 10 m^3). Dans ce cas, on regroupe les lots afin d'obtenir un conteneur complet et l'empotage est effectué par la compagnie maritime ou l'organisateur de transport (transitaire).

1.5.5 Les opérations dans un terminal à conteneurs

Les opérations de manutention se réalisent au terminal à conteneurs (ensemble de quais et parcs de stockage spécialisés par type de marchandises). Les navires se placent à quai au regard des portiques (grues de quai pour soulever les conteneurs). A bord du navire, les dockers désarriment les conteneurs qui sont liés les uns aux autres par les pièces de coin durant la traversée. Le portiqueur peut alors placer le *spreader* (structure où sont fixés les verrous permettant d'accrocher et de soulever le conteneur) à l'aplomb du conteneur et commencer le déchargement. Au pied de chaque portique un homme ou un système vidéo veille pour repérer l'immatriculation du conteneur et préciser sa position (rangement dans le parc de stockage ou placement sur remorque ou wagon) à un autre docker présent dans un cavalier gerbeur *straddle carrier* qui va se charger de la manœuvre.

Une fois le déchargement réalisé, les manœuvres s'inversent pour les opérations de chargement. A peine quelques heures suffisent. L'évolution technologique liée à la conteneurisation a profondément modifié les conditions de travail des dockers : ils sont moins nombreux, mais plus spécialisés et qualifiés. Les dockers préparent le matériel, participent à l'ouverture des panneaux de cale, guident les conducteurs de portiques et pilotent les chariots élévateurs à terre. Le pointeur est responsable de la gestion du parc à conteneurs. Depuis son terminal informatique, il affecte les marchandises à des emplacements précis en fonction de leurs destinations. Il est également chargé d'identifier et de contrôler les conteneurs qui quittent le terminal. Le planificateur de navire *ship planner* est chargé d'organiser le plan de chargement sur un navire : il doit attribuer à chaque conteneur un emplacement précis à bord du navire. Il veille à ce que la stabilité du navire soit respectée. En effet, le placement des conteneurs est effectué de façon à faciliter leur déchargement dans la chronologie de leurs destinations c'est-à-dire les ports de débarquement.

1.5.6 Les systèmes d'information

Les gigantesques progrès informatiques ont permis d'accélérer et de simplifier le transit des navires et des marchandises dans les ports ; en voici deux exemples mis en place par le port autonome de Nantes/Saint-Nazaire, le système GIM-

NAUTE (Gestion Informatique des Mouvements des Navires Avec Utilisation des Technologies Extranet), qui permet à l'ensemble des intervenants (agents maritimes, capitainerie, pilotes, lamaners, etc.) de correspondre et d'intervenir en temps réel grâce au réseau informatique, remplaçant le téléphone et le fax. Le système ADEMAR (Accélération des Expéditions Maritimes) est un réseau informatique opérationnel au Havre depuis 1983 qui gère en temps réel toutes les opérations administratives, douanières et logistiques concernant le passage des marchandises par le port.

L'informatique est aussi présente pour vérifier et seconder l'homme dans toutes les tâches nécessaires sur les terminaux (contrôles douaniers, positionnement des conteneurs sur les navires et les parcs de stockage, etc.). Au Havre toujours, le système Sycoscan des Douanes permet le contrôle des conteneurs sur un terminal informatique via un scanner placé dans un tunnel par où transitent les boîtes qu'il n'est plus nécessaire d'ouvrir pour en vérifier le contenu.

Depuis la moitié des années 1990, les systèmes *Global Positioning Systems-GPS* sont installés dans les terminaux à conteneurs. Initialement, ils ont été utilisés pour identifier automatiquement la position des conteneurs dans la zone de stockage. Par la suite, la mise en place de *Differential GPS* [Ste99] est devenue nécessaire tenant compte de la taille des conteneurs et de l'aspect de la zone de stockage. Les composants d'un DGPS ne sont plus installés dans les conteneurs mais au sommet des équipements de transport et de stockage. La position d'un conteneur est exprimée en terme de coordonnées dans la zone de stockage et transmise à l'ordinateur au moment de l'empilement et de dépilement de ce conteneur. Des systèmes alternatifs aux *DGPS* tels que les systèmes optiques comme *Laser Radar* sont utilisés aussi pour repérer les conteneurs.

Des transpondeurs *Transponder* et des circuits électriques sont utilisés pour guider les grues de transport et les véhicules automatiquement guidés tandis que les *DGPS* sont utilisés pour guider les cavaliers gerbeurs automatiquement et d'autres équipements de manutention.

1.5.7 La multimodalité *Multimodal Transport*

Les transporteurs maritimes ont développé des chaînes de transport à l'intérieur des continents. Le caractère d'interchangeabilité du conteneur est à l'origine de l'essor des réseaux de transport mondiaux et des chaînes de transport qui associent le rail, la route et le fleuve au transport maritime pour concevoir un transport de bout en bout et de porte à porte. Les conteneurs peuvent être acheminés indifféremment par camions, barges, wagons ou navires. L'accroissement du trafic conteneurisé induit un formidable développement des transports de pré et post-acheminement, c'est à dire apportant les marchandises du continent jusqu'au navire de chargement et l'emmenant du navire déchargé à sa destination finale sur le continent. Le parcours terrestre est fondamental car il est le premier et le dernier maillon de la chaîne logistique multimodale.

La route est un élément stratégique qui présente des avantages certains : souplesse, limitation des transferts de charge, assurance du suivi, etc. Mais les camions ne peuvent pas transporter plus de deux EVP ou un 40 pieds en même temps : il en résulte une augmentation du trafic routier et une saturation de certaines voies.

Les dessertes ferroviaires et fluviales offrent les solutions les plus adaptées pour faire face à la massification et à la congestion du trafic routier. La commission européenne vise à établir sur le territoire communautaire des corridors de fret ferroviaires *freeways* pour faciliter la desserte des continents : ce sont des voies dédiées aux transports de marchandises en liaison avec l'utilisation des trains-blocs et des convois de trains de conteneurs qui sillonnent l'Europe. Aux Etats-Unis, les trains-blocs *double stacks* (conteneurs sur deux étages) peuvent atteindre une longueur de cinq kilomètres et relient la côte est à la côte ouest constituant un véritable " pont terrestre " ou " pont ferroviaire ".

Aujourd'hui, le souci d'un développement durable élargit considérablement, quand cela est possible, la place du fluvio-maritime dans la circulation des conteneurs à l'intérieur des continents. Les conteneurs sont empilés sur plusieurs niveaux sur des barges qui assurent des liaisons entre les ports maritimes et les ports fluviaux, voir entre les ports et les usines et dépôts quand ils sont localisés le long de la voie d'eau.

Le cabotage (transport côtier) repose sur les navires nourriciers *feeders*, qui relient les ports moyens aux ports principaux et aux *hubs* où les navires-mères font escale. Les *feeders* alimentent ainsi les navires transocéaniques en conteneurs et redistribuent les cargaisons dans les plus petits ports de ces mêmes grands navires. Cette organisation de collecte et de redistribution des marchandises crée un véritable réseau mondial de transport et de distribution, complété par les corridors de fret ferroviaires et les autoroutes continentales.

1.5.8 Avantages et Inconvénients de la conteneurisation

Le déplacement des marchandises par conteneurs a plusieurs avantages par rapport au transport de marchandises non groupées. Le conteneur favorise l'automatisation de plusieurs opérations de manutention, ce qui accélère le chargement et le déchargement et autorise un transfert plus rapide d'un mode de transport à un autre. En outre, le conteneur protège les marchandises des intempéries et du chapardage et la manutention, devenant plus simple, se traduit par une quantité moindre de dommages.

Les avantages inhérents aux conteneurs ont pu être réalisés grâce aux capitaux considérables que les exploitants ont investis en navires, terminaux, grues, wagons à deux niveaux de chargement et autres installations ou équipements de manutention spécialisés.

Toutefois, les opérateurs, les ports et les autorités de tutelle faisant en sorte de s'adapter aux contraintes de la logistique moderne, se heurtent à un certain nombre de limites de la conteneurisation. Les contraintes physiques aussi bien dans les transports courts que sur longues distances nécessitent un minimum de protection préventive. En effet, les températures élevées ainsi que la ventilation réduite dans le conteneur entraînent très souvent de lourds dommages si l'on n'y prend pas garde. La température provoque la dessiccation des marchandises (cacao) ou des emballages (carton, bois d'arrimage ou de calage, bois de palette, etc.). La vapeur d'eau, une fois libérée, se condense dans les zones les plus froides du conteneur et en général lors de l'alternance jour-nuit. C'est alors une pluie qui semble retomber sur les marchandises, avec des dégâts irrémédiables faciles à imaginer.

1.6 Les ports conteneurisés

Dans le transport par voies navigables qu'est le couple chemin de fer-navire à vapeur, c'est le port, lieu d'échanges privilégié, qui constitue le rouage essentiel, avec ses immenses triages dans les zones portuaires et son lacis de voies ferrées sur tous les quais, entre les différents bassins. Le développement des ports est d'autant plus important que la machine à vapeur ait permis un très grand accroissement de la taille des navires, donc de leur capacité. De nombreux bassins sont creusés dans les ports qui s'agrandissent en raison de la croissance du trafic. Ces bassins, qui sont édifiés à partir de 1830, offrent des dimensions inconnues jusqu'alors (de l'ordre du kilomètre en longueur). Ils constituent les premiers éléments des paysages portuaires traditionnels (grues, entrepôts, marchandises en vrac sur les quais, voies ferrées) en continuité directe avec la ville portuaire. Dans les mers à forte marée, les bassins sont accessibles à marée haute grâce à des écluses : ce sont les bassins à flot. Dans les mers sans marées, ce sont des darses. Il s'en est suivi la dissociation de la ville portuaire traditionnelle et du port proprement dit, c'est-à-dire de la fonction technique de chargement et déchargement des navires, maintenance et avitaillement, d'une part, et de la fonction commerciale, d'autre part. Cette évolution s'est traduite par le glissement du port vers l'aval par rapport à la ville et par le passage de la notion de port à celle de terminal spécialisé, c'est-à-dire un simple poste technique de manutention, souvent totalement automatisé, dépourvu de toute urbanisation (Antifer, par exemple, pour Le Havre). Comme pour les réseaux linéaires, la densité des ports ou des terminaux apparaît très inégale selon les régions du monde. Il en est de même du niveau de développement technique. Les deux zones du globe les plus densément pourvues en ports sont l'Europe : de la Baltique à la Méditerranée, et l'Asie du sud-est, la seconde présentant cependant un niveau général de développement technique moindre que la première, sauf au Japon et dans les autres ports de premier niveau.

Un port moderne est un ensemble de terminaux spécialisés. Les plus grands (voir tableau 3.1) [Por06] jouent le rôle de plate-forme de concentration / éclatement de dimension continentale.

Ainsi, les ports principaux *main ports* tels que : Los Angeles, Rotterdam, Hong Kong, etc. ont-ils émergé. Dans certains, les *hubs*, les conteneurs ne font

TAB. 1.1 – Les 10 premiers ports conteneurs dans le monde en 2006

Ports	Trafic en milliers d'EVP
Singapour	24 792
Hong Kong	23 234
Shanghai	21 710
Shenzhen	18 468
Pusan	12 030
Kaoshiung	9 774
Rotterdam	9 690
Dubai	8 923
Hamburg	8 861
Los Angeles	8 469

que transiter entre les grands navires mères est-ouest et les relais nord-sud ou entre les navires mères et les *feeders* régionaux. Parmi les *hubs* les plus importants, on trouve Algésiras (Espagne), Gioia Tauro (Italie) et Kingston (Jamaïque). Ainsi, un conteneur qui doit se rendre de Tokyo à Abidjan empruntera un navire mère de 5000 EVP reliant le Japon à l'Europe du nord avec une escale à Algésiras (près de Gibraltar) où le conteneur sera débarqué pour être aussitôt réembarqué sur un navire de 1500 EVP qui descend des ports de la mer du nord vers l'Afrique de l'ouest. Le trajet Tokyo / Abidjan aura duré une quarantaine de jours. Toute l'organisation portuaire est donc très hiérarchisée et repose autant sur un petit nombre de grands ports mondiaux que sur un nombre plus important de ports moyens.

1.7 Conclusion

Nous avons tenté dans ce chapitre introductif de décrire le cadre général de notre travail et de connaître au mieux les nouveaux concepts s'y posant, et ce dans le but de contourner et de maîtriser la problématique ci étudiée.

Comme nous le constatons, la conteneurisation et la logistique des terminaux à conteneurs mettent en questions plusieurs problèmes de décisions éma-

nant des différentes opérations effectuées au sein d'un terminal à conteneurs. Ces problèmes peuvent être classés parmi les problèmes de transport et d'ordonnement [BGAB83] comme par exemple le problème de tournées de véhicules [TV02] et les problèmes d'affectation qui sont largement abordés dans la littérature. Dans le chapitre suivant, nous présentons un état de l'art exhaustif de ces problèmes. Nous nous penchons sur le Problème de Stockage de Conteneurs-PSC auquel nous nous intéressons dans cette thèse.

Chapitre 2

Problème de Stockage de Conteneurs (PSC)

2.1 Introduction

De nos jours, les navires de conteneurs sont déchargés et chargés dans de larges terminaux à conteneurs. La figure 2.1 illustre le processus de déchargement et de chargement dans un terminal à conteneurs. Ce processus peut être divisé en opérations ou activités qui vont être présentées en détail dans les sections suivantes. Lorsqu'un navire arrive au port, des portiques ou des grues de quais *Quay Cranes-QC's* déchargent les conteneurs en import par ce navire. Puis, ces conteneurs sont transférés de ces grues à des véhicules se déplaçant entre le navire et la zone de stockage des conteneurs. Cette zone est constituée d'un nombre de baies où les conteneurs peuvent être stockés en piles pendant une certaine période. Des équipements comme les grues ou les cavaliers gerbeurs *straddle carriers* sont affectés à ces baies.

Un cavalier gerbeur peut à la fois transporter des conteneurs et les stocker dans une zone de stockage. Il est possible aussi d'utiliser des véhicules dédiés au transport des conteneurs. Si un véhicule atteint la zone de stockage, il dépose sa charge ou bien une grue de stockage décharge le conteneur et le stocke dans la zone. Après une certaine période, les conteneurs sont retrouvés par les grues dans la zone de stockage pour être conduits sur des véhicules vers différents moyens de

transport : barges, navires, trains ou camions selon différents modes de transport : voie navigable, voie ferroviaire ou voie terrestre. Pour le chargement des conteneurs en export sur un navire, ces activités sont exécutées dans l'ordre inverse. La plupart des terminaux utilisent des équipements manuellement guidés : les cavaliers gerbeurs *straddle carriers*, les grues *cranes* et les systèmes multi-remorque *multi-trailer systems*. Toutefois, quelques terminaux sont automatisés. Dans de tels terminaux, deux différents types de véhicules automatiques sont utilisés pour acheminer les conteneurs d'un mode de transport à un autre : des véhicules automatiquement guidés *Automated Guided Vehicles-AGV's* ayant besoin d'une grue pour charger et décharger un conteneur, et des véhicules élévateurs automatiques *Automated Lifting Vehicles-ALV's* capables d'élever un conteneur de manière autonome. La comparaison et la décision du choix entre ces deux types d'équipements sont discutées dans [VH04]. De plus, le processus d'empilement peut être effectué automatiquement via des grues d'empilement automatiques.

L'évolution du transport conteneurisé incite au développement des terminaux à conteneurs, autant de la logistique des conteneurs que des équipements techniques de manutention. Une concurrence entre les ports et un développement continu de l'activité portuaire, particulièrement entre les plus proches géographiquement, a accompagné cette évolution. En conséquence, l'objectif global est la minimisation du temps passé par un navire dans le port tenant compte des différentes opérations se déroulant dans un port.

L'objectif de ce chapitre est de fournir une classification de ces opérations. En se basant sur cette classification, nous présentons une revue de littérature des travaux inhérents qui décrivent différentes approches incluant des méthodes exactes, des méthodes heuristiques ainsi que des approches basées sur des simulations, centralisées et distribuées.

2.2 Processus dans un terminal à Conteneurs

Dans ce qui suit, nous allons parcourir les différentes activités rencontrées dans un terminal à conteneurs 2.1) allant du déchargement d'une barge ou d'un navire (camion ou train) jusqu'à leur chargement sur camion ou train (barge ou navire) pour les conteneurs en import (en export). Nous nous intéressons aussi

aux différents équipements de manutention qui leur sont associés, et discuter des différents problèmes qui en découlent. Nous allons aussi survoler quelques travaux inhérents incluant les méthodes exactes, les méthodes heuristiques et les approches basées sur des simulations en se référant à [VK03, SVS04, GG06].

2.2.1 Planification de navires

Le processus de planification de navires se compose de trois sous-opérations : l'allocation des postes à quai, l'arrimage de conteneurs et l'allocation des portiques (ou grues de quais).

2.2.1.1 Allocation des postes à quai *Berth Allocation*

La stratégie Premier Arrivé Premier Servi *First In First Out-FIFO* semble être une manière rationnelle d'affecter les navires arrivant aux postes à quai disponibles, mais elle peut être inefficace dans le cas où le temps de manipulation des navires dépendrait des postes à quai correspondant. En effet, dans le cadre de manipulation des conteneurs, une politique d'empilement de conteneurs consis-

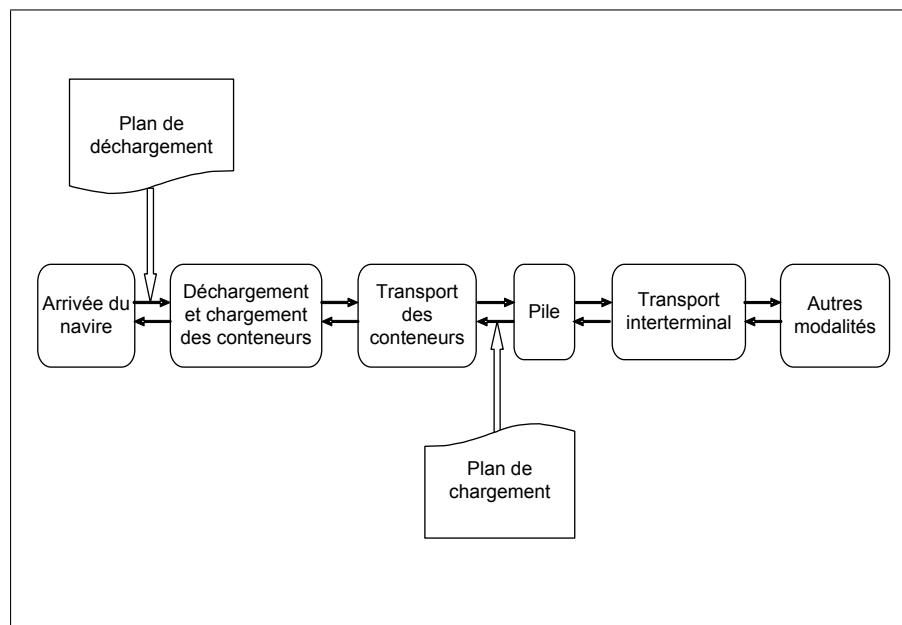


FIG. 2.1 – Succession des opérations dans un terminal à conteneurs

tant à empiler les conteneurs à charger dans un même navire à l'intérieur d'une zone de stockage dédiée.

En conséquence, il est préférable que les postes à quai soient proches de cette zone. Lors de l'allocation d'un navire à un poste à quai, on doit tenir compte de la disponibilité des postes à quai et des grues responsables du chargement et du déchargement du navire. Différents travaux de recherche traitent ce problème nommé : le problème d'allocation de postes à quai *Berth Allocation Problem-BAP*.

Le problème d'allocation de postes à quai, dit encore le problème d'affectation des navires aux postes à quai, a pour objectif de minimiser la somme du temps d'attente et du temps de manutention des navires. Chaque poste à quai ne peut supporter qu'un seul navire à la fois, et le temps de manipulation d'un navire dépend du poste à quai associé.

Dans [LM01], les auteurs présentent un modèle de réseaux de files d'attente et une étude expérimentale pour la simulation de l'arrivée, l'allocation et le départ de navires.

Dans [INP01], Imai et al. considèrent plusieurs versions relatives à ce problème dont les plus étudiées : le BAP statique et le BAP dynamique : dans la première version, il est supposé que tous les navires arrivent déjà au port au début de l'horizon de planification. Dans ce cas, il n'existe pas de temps morts entre les navires manipulés dans un certain poste à quai. Le problème ainsi défini peut être formulé par un problème d'affectation à deux dimensions pouvant être résolu en un temps polynomial. Imai et al. considèrent le BAP statique à objectifs multiples : la minimisation de la somme du temps d'attente et de manipulation des navires et la minimisation de l'insatisfaction du navire dépendant de l'ordre de l'affectation. Notons que l'ordre optimal d'affectation, tenant compte du temps d'attente et du temps de manutention, peut ne pas correspondre à l'ordre dans lequel les navires sont arrivés.

Le BAP dynamique suppose que les navires arrivent tout au long de la période de planification, donc il peut y avoir un temps d'inactivité entre deux navires consécutifs desservis dans un poste à quai. Le problème ainsi défini ne peut pas alors être résolu en un temps polynomial. Imai et al. [INP01] considèrent aussi le BAP dynamique où il n'est permis de manipuler qu'un seul navire à la fois tant que la longueur totale des navires est inférieure à celle du poste à quai considéré. En

outre, ils incluent la contrainte additionnelle qu'un navire ne peut être servi dans un poste à quai que si la profondeur de l'eau nécessaire pour sa mobilisation *draft* est inférieure à la profondeur de l'eau au poste à quai. [Lim98] étudie ce problème comme étant une version particulière du problème du paquetage *bin-packing* à deux dimensions.

Dans [PK01] étudient l'allocation des postes à quai en supposant que les navires sont proches de la zone où les conteneurs associés sont empilés, et visent à minimiser le temps *turnaround* des navires. [KM03] propose un modèle de programmation linéaire en nombres entiers mixte *Mixed Integer linear Programming-MIP* pour formuler le BAP, et appliquent l'algorithme de recuit simulé pour le résoudre. Les résultats expérimentaux ont montré que les solutions retrouvées par le recuit simulé sont similaires aux solutions optimales retrouvées par le modèle MIP.

Dans [GC04], Guan et Cheung proposent une procédure se basant sur un arbre de recherche et des heuristiques pour résoudre de problèmes de grande taille dans le but de minimiser le temps de flux total *Total weighted flow time*.

2.2.1.2 Arrimage de conteneurs *Container Stowage*

Il s'agit de la planification d'arrimage des conteneurs dans un navire. En général, un navire fait escale dans un nombre de ports où les conteneurs seront chargés et déchargés. Ceux qui sont chargés sont destinés à un nombre de ports successifs tout au long la route du navire en question.

Ce problème considère l'affectation des conteneurs à différentes positions dans le navire tout en maintenant sa stabilité et en minimisant le nombre de mouvements inutiles ou parasites *unproductive moves*. Ces derniers apparaissent dans un port lorsqu'on y dispose de conteneurs stockés au-dessus d'autres destinés à ce port, alors ils doivent être déchargés et rechargés de nouveau. Le placement des conteneurs dans le navire doit alors être accompli de manière à maintenir sa stabilité, sa torsion et sa force. Évidemment, les conteneurs lourds sont généralement stockés au fond du navire, les conteneurs légers sont alors empilés au-dessus [KPR00]. Notons que les conteneurs de tailles différentes ne peuvent pas être empilés les uns sur les autres.

En plus du poids et de la taille, d'autres critères sont pris en considération :

les conteneurs frigorifiques par exemple demandent une puissance électrique existant seulement dans des positions spécifiques dans le navire, certains conteneurs transportant des matériaux dangereux exigent des conditions d'arrimage bien déterminées distinctes de celles des autres. Outre ces restrictions techniques, la destination des conteneurs doit être prise en compte.

Avriel et al. [APS00] se concentrent sur la planification d'arrimage dans le but de minimiser le nombre de mouvements unproductifs. Ils proposent un modèle de programmation linéaire binaire.

Dans [WR00], les auteurs divisent ce problème en deux sous-problèmes selon un niveau de planification stratégique et tactique. Ils utilisent la méthode *Branch and Bound B&B* pour résoudre le problème d'affectation de conteneurs à une zone dans le navire, et la méthode de recherche tabou pour déterminer un plan détaillé d'affectation des conteneurs à des emplacements spécifiques dans le navire. De bons résultats sont obtenus en un temps raisonnable.

Les deux travaux [RW02, WRW01] présentent un algorithme génétique pour générer automatiquement des plans d'arrimage stratégiques.

La simulation et l'optimisation en ligne sont considérées dans [Win00] et [SWZ01]. Les temps d'attente des grues et la congestion des équipements de transport sont minimisés afin d'éviter la réduction de la productivité. Le travail de [SWZ01] présente un modèle *MIP* pour modéliser un ordonnancement juste à temps *Just-in-time JIT* d'une portique, et des méthodes exactes et heuristiques pour la résolution.

A un niveau plus détaillé, l'affectation des navires aux postes à quai peut être accomplie tenant compte de l'affectation de portiques aux navires. Certainement, le nombre de portiques affectées à un navire influe le temps de manipulation de ce dernier. Ceci donne naissance au problème d'ordonnancement des grues *Crane scheduling problem* qui a été étudié par [Dag89].

[Bis03] étudie ce problème comme étant une composante d'un plus grand problème appelé *Multiple-crane-constrained Vehicle Scheduling and Location problem-MVSL*. Il développe une méthode heuristique pour minimiser le temps total passé par un ensemble de navires aux quais.

Ce problème est aussi classé en deux versions : le problème statique où un nombre de portiques doit servir un nombre de navires tout en minimisant les

coûts associés au retard des navires, en supposant que tous les navires à manipuler soient arrivés au poste à quai à l'instant zéro.

Le problème d'ordonnancement statique des grues peut être généralisé afin de supporter le cas de navires arrivant à différentes dates et les contraintes de capacité relatives aux postes à quai : c'est un problème d'ordonnancement dynamique des grues.

2.2.2 Transport sur le quai *Quay Transport*

Après une observation des différents terminaux à conteneurs à travers le monde entier, on a pu constater que trois systèmes de manipulation pour le transport sur le quai sont les plus usités :

1. Le premier système utilise les grues d'empilement *stacking cranes* pour la récupération des conteneurs empilés dans la zone de stockage. Ces grues peuvent être soit *rubber-tired* ou *rail-mounted*. Selon le premier mode, les grues peuvent généralement se déplacer d'une rangée de piles à une autre, et ce suivant l'emplacement des conteneurs destinés à un navire bien déterminé. Après avoir retrouvé le conteneur recherché, la grue d'empilement le charge sur un véhicule transporteur, généralement un wagon transporteur de terminal *truck terminal* ou un véhicule guidé automatisé *Automated Guided Vehicle-AGV*, capable de transporter un ou deux conteneurs. Toutefois, quelques terminaux utilisent les systèmes multi-remorque *multi-trailer* capable de transporter 10 TEU. Le véhicule du terminal conduisent les conteneurs vers la grue de quai appropriée, qui se charge de leur soulèvement. Dans la plupart des terminaux, les grues d'empilement et les wagons transporteurs du terminal sont manuellement dirigés (plusieurs terminaux à Rotterdam possèdent un équipement automatisé).
2. Un système alternatif pour le transport des conteneurs dans un quai consiste à utiliser des cavaliers gerbeurs *straddle carriers* ou *yard machines*. Ces derniers combinent les propriétés d'une grue et celles d'un véhicule. En effet, ils sont capables de retrouver les conteneurs recherchés et de les amener vers les portiques. Étant donné que ces véhicules peuvent eux-mêmes charger et décharger les conteneurs, ils n'ont pas besoin d'attendre une grue à condition

qu'il existe une zone tampon de capacité suffisante. En vue de ces différents arguments, les cavaliers gerbeurs semblent être une composante préconisée de l'équipement d'un terminal.

Cependant, ils sont trop coûteux et beaucoup plus incertains qu'un système alliant des grues d'empilement et des véhicules de terminaux. De plus, les cavaliers gerbeurs nécessitent plus d'espace pour pouvoir fonctionner.

3. Le troisième système possible n'est plus utilisé : il consiste à empiler tous les conteneurs sur des châssis. Ensuite, seuls les wagons transporteurs du terminal sont utilisés pour conduire les remorques aux portiques. Ce système a l'avantage de pouvoir accéder directement à chaque conteneur ce qui facilite le fonctionnement dans un terminal, mais son inconvénient majeur est de requérir un grand espace de stockage puisque chaque conteneur est empilé sur un châssis.

Le choix entre ces trois alternatives est un problème typiquement stratégique. L'un des paramètres intervenant dans ce choix, est celui de l'emplacement géographique du terminal. Par exemple, les terminaux asiatiques (Hong Kong et Singapour) utilisent le premier système vu que les grues d'empilement permettent un empilement plus élevé (atteignant 6 au port de Busan au Japon), en comparaison au deuxième système, qui est limité à un empilement de 3 à 4 conteneurs (exp. : port de Radès en Tunisie). D'autre part, si l'espace de stockage est largement vacant, par exemple dans les ports des USA, un système utilisant des remorques est de prédilection pour la facilité relative de faire fonctionner un tel système.

En Europe, la situation est différente : bien que l'empilement est relativement peu élevé (2 à 4 conteneurs), les cavaliers gerbeurs (Hambourg, Bremen, Rotterdam et Antwerp), les grues d'empilement automatiques *Automated Stacking Crane -ASCs* et les véhicules automatiques guidés *Automated Guided Vehicles AGVs* (Rotterdam) sont tous utilisés.

En vue de comparer et de décider quel système il faut utiliser, Vis et al. [VKS01] ont procédé à une étude par simulation. Ils prétendent qu'en moyenne, le nombre de cavaliers gerbeurs requis uniquement pour le transport des conteneurs et non pas pour l'empilement est environ 30% de moins que le nombre de véhicules automatiques guidés requis. Cette constatation encourage l'utilisation des cavaliers gerbeurs, mais tenant compte d'autres paramètres tels que le coût

et la fiabilité, ces équipements seront moins encouragés.

Dans ce contexte, le travail de (Yim et al.) traite un problème peu abordé, c'est le problème d'affectation des conteneurs dans un port fluvial, qu'ils appellent aussi problème de rotation de barges et d'affectation de conteneurs. Il s'agit de minimiser le nombre de barges utilisées et de déterminer sur quelles barges les conteneurs seront chargés tout en satisfaisant un ensemble de contraintes additionnelles. Une fois que les conteneurs affectés aux barges retenues, le plan de chargement de chacune d'elles peut être déterminé en résolvant le problème d'arrimage *stowage problem* évoqué dans la section 2.2.2. Ce problème est clairement relié au problème de rotation des barges. Yim et al. Développent un modèle de programmation linéaire en nombres entiers -PLNE pour modéliser ce problème.

Dans ce qui suit, nous allons discuter des divers problèmes d'ordonnancement dérivés du problème principal de transport sur le quai. Ces problèmes paraissent au niveau de décision opérationnel. Rappelons que le but est d'optimiser l'utilisation de l'équipement disponible, et que l'objectif global est de minimiser le temps passé par les navires dans le port.

2.2.3 Ordonnancement des équipements de manutention

Plusieurs travaux s'intéressent au problème d'ordonnancement des équipements de manutention. Ces travaux se distinguent selon le type des équipements de manutention à ordonnancer. Nous pouvons les classer comme suit :

- Ordonnancement des Grues d'Empilement *Scheduling of Stacking Cranes*.
- Ordonnancement des Véhicules Automatiques Guidés *scheduling of AGVs*.
- Contrôle du trafic des Véhicules automatiques guidés *traffic Control of AGV's*.
- Ordonnancement intégré des grues d'empilement et des véhicules automatiques guidés *Integrated Scheduling of Stacking Cranes and AGVs*.
- Ordonnancement des cavaliers gerbeurs *Straddle Carriers*.

Un résumé de ces travaux est donné dans le tableau 2.1

En outre, le principe de fonctionnement ainsi qu'une étude comparative des performances des équipements de manutention dans un terminal à conteneurs est représentée dans la figure 2.2 extraite de [Equ].

Le problème de stockage de conteneurs auquel nous nous intéressons constitue aussi un problème de décision représentant l'activité de stockage de conteneurs, composante prépondérante de tout le processus de gestion d'un port. En consé-

TAB. 2.1 – Tableau récapitulatif des algorithmes d'ordonnancement des équipements de manutention

Classe de Problèmes	Référence	Problème étudié	Caractéristiques de la solution
Ordonnancement de Grues d'empilement	[KK98]	Optimisation des tournées d'une seule grue d'empilement	*Classement des conteneurs en catégories (taille, poids, port de destination, etc.). *Plusieurs baies peuvent contenir des conteneurs de même catégorie.
	[MJLZ ⁺ 00]	Redistribution des grues d'empilement de type rubber-tired	*Un terminal avec des zones de stockage auxquelles est associée une charge de travail lors de chaque déplacement. *Une grue d'empilement est autorisée à se déplacer d'un bloc à un autre. *Minimiser la quantité de la charge de travail.
	[ZWLL02]	Optimisation de la productivité d'un type de grues d'empilement de conteneurs : les Rubber Tyred Gantry Cranes-RTGs	*Retrouver les dates et les itinéraires de déplacement de ces grues de manière à minimiser la charge de travail totale retardée dans une zone de stockage. *Formulation via un modèle de programmation entière mixte. *Résolution par une relaxation lagrangienne.
Ordonnancement de des AGVs	[KB98]	Distribution d'AGVs	*Pour chaque conteneur, introduire des dates associées à des événements event times qui doivent être satisfaits (dates exactes de chargement /déchargement de chaque conteneur). *Réduire le problème à un problème d'affectation.
	[dM00]	Distribution d'AGVs	*Etudier des règles de distribution
	[MW01b]	Distribution d'AGVs	*Ajustement des règles de distribution dans le cas d'un port conteneurisé automatique
	[VKRP01]	Détermination du nombre d'AGVs requis dans un terminal à conteneurs semi-automatique	*Les conteneurs sont disponibles à des instants connus à court terme et changent constamment au cours du temps.
Ordonnancement des différents équipements de manutention	[KKHK04]	Ordonnancement des opérateurs <i>operator-scheduling problem</i>	*Affectation des opérateurs à des intervalles de temps associés à l'utilisation des équipements. *Formulation et résolution via une technique de satisfaction de contraintes (le temps d'exploitation, le temps d'arrêt, les préférences des opérateurs).

Contrôle des AGVs	Evers et Koppers (1996)		<p>*Flexibilité de représentation du système de transport.</p> <p>*Peu de communication nécessaire pour le contrôle des AGVs (les AGVs communiquent seulement avec des zones locales).</p> <p>*Dates de déplacement des AGVs stochastiques.</p> <p>*Assurer que les AGVs se placent dans la queue dans le bon ordre.</p>
	Duinkerken et Ottjes (2000)		<p>*Etudes par simulation étendues</p> <p>*Evaluer la performance d'un tel système sous un contrôle intelligent du trafic.</p>
Ordonnancement intégré des grues d'empilement et des AGVs	Meersmans et al. (2001)	Planification de l'activité des grues de quai, des grues d'empilement et des AGVs de manière intégrée	<p>*Modèles prenant en considération la typologie du terminal : cas où les AGVs passeraient par un point commun alors l'ordre dans lequel ils commencent à soulever leur conteneur suivant, détermine complètement la partie qui reste du plan.</p> <p>Cas contraire : formulation basée sur MIP dérivant des classes d'inégalités valides.</p>
Ordonnancement des cavaliers gerbeurs	[BRSV00]	Distribution des cavaliers gerbeurs	<p>*Algorithme génétique pour minimiser les retards de transferts aux portiques.</p> <p>*Une affectation statique puis une affectation dynamique des cavaliers gerbeurs (dans le cas de leur collision).</p> <p>*Résultats expérimentaux montrant qu'une affectation dynamique peut accroître considérablement la productivité.</p>
	[Win00]	Combinaison de la planification de l'arrimage d'un navire avec l'ordonnancement des cavaliers gerbeurs	<p>*Classement des conteneurs à charger, puis affectation d'une séquence des catégories à charger à chaque grue de quai.</p> <p>*Choix libre d'un conteneur spécifique durant le chargement d'un navire.</p>
	[KP99]	Optimisation des transferts de conteneurs par les cavaliers gerbeurs	<p>*Minimisation la durée passée par les navires dans le port. Proposer des modèles qui traitent les remaniements des conteneurs en introduisant les dates de setup.</p>

quence, nous y attardons et consacrons toute la suite de ce chapitre.

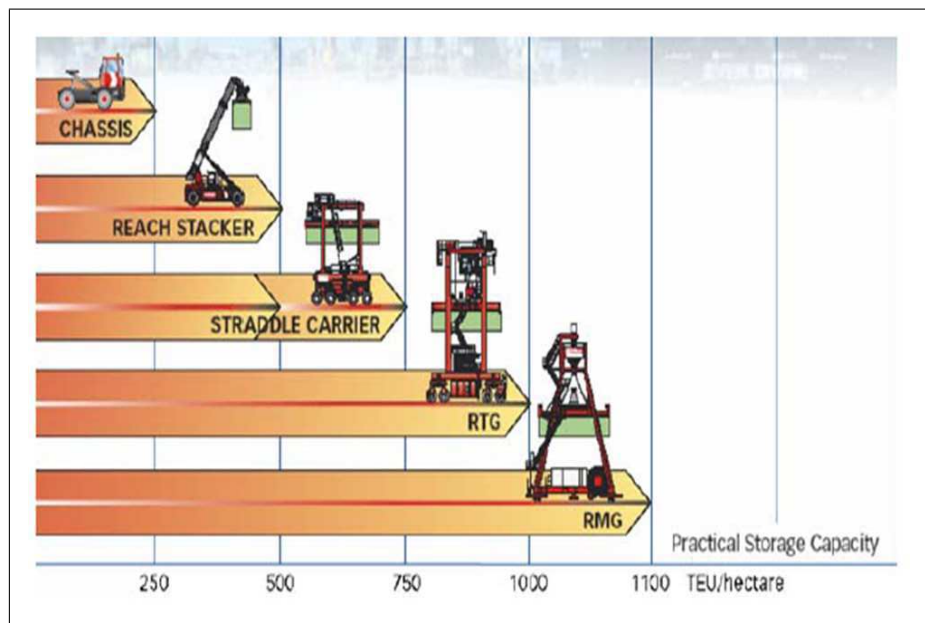


FIG. 2.2 – Types d'équipements de manutention

2.3 Problème de Stockage de Conteneurs -PSC

Container Stacking Problem

Récemment, vu la grande concurrence entre les ports, l'amélioration du service clientèle est devenu un problème important pour un port de terminaux à conteneurs. L'une des mesures de performance du service clientèle est le temps total passé par les navires aux quais d'un port *berthing time* ou *throughout time*. Ce temps est composé en majeure partie du temps de chargement et de déchargement des conteneurs. Dans le but de réduire le temps de chargement, il est nécessaire de choisir des emplacements de stockage pour les conteneurs en import ou en export de manière à pouvoir être chargés efficacement dans le navire, le camion, ou le wagon associé.

Dans cette section, nous nous focalisons sur l'opération alliant transbordement, empilement et dépilement des conteneurs. Cette opération constitue un problème représentant un maillon fort important de la chaîne des problèmes de décision rencontrés dans un terminal à conteneurs : c'est le Problème de Stockage de Conteneurs *Container Stacking Problem* ou *Container Handling Problem*. Ce

problème sera désigné dans la suite par l'acronyme PSC . Le processus de stocker et de retrouver les conteneurs doit être exécuté de manière à assurer un bon déroulement du reste des opérations ayant lieu dans le terminal. L'efficacité de ce processus dépend entre autres du taux d'occupation des zones de stockage ainsi que des stratégies définies pour le stockage et la recherche des conteneurs en import ou en export.

Commençons par introduire une notion de base rencontrée dans la réalité de l'activité de stockage de conteneurs, et adoptée tout au long de notre travail : La pile (appelée aussi colonne).

Dans notre contexte, la pile permet aux activités diverses de transport de se produire indépendamment l'une de l'autre. Si aucune pile n'existe, alors chaque arrivée de navire (ou autre mode) devrait être suivie directement par la décharge aux véhicules transporteurs. Ceci constitue une opération logistique complexe car toutes les arrivées devraient être coordonnées. Toutes ces contraintes causeraient beaucoup de congestion et seraient susceptibles aux retards inévitables.

Il existe différents systèmes d'empilement. Dans notre cadre, nous considérons l'empilement en piles dans une zone de stockage. En fait, c'est le système adopté dans la plupart des terminaux à conteneurs.

Quelques conteneurs comme par exemple les conteneurs frigorifiques *reefers* exigent des emplacements spéciaux où l'électricité peut être fournie. La détermination de la capacité ou encore la hauteur de la pile est un problème de conception important dans un terminal puisque la pile occupe un espace coûteux.

L'empilement élevé peut être préconisé, mais le nombre estimé de remaniements croît brusquement avec la hauteur d'empilement. Rappelons qu'un remaniement est un mouvement improductif ou encore parasite tel que défini ci-haut.

Un autre paramètre en relation avec la hauteur de la pile doit être pris en compte, il s'agit de sa disposition : quelle largeur et longueur doit-elle avoir ? Combien de piles seront-elles utilisées ? Ensuite, on doit décider dans la conception du type d'équipement qui sera utilisé pour l'empilement.

Souvent, les piles sont classées dans deux zones de stockage différentes : une zone d'import et une zone d'export. Les conteneurs en import arrivent d'habitude dans de grands navires porte-conteneurs, de manière quelque peu prévue. Cependant, ils vont probablement partir dans un ordre imprévisible, donc ils ne

peuvent pas être empilés qu'en haut.

Les conteneurs en export peuvent arriver aléatoirement, mais leur départ est d'habitude lié à un navire et de là, ils peuvent être empilés de meilleure manière. Au niveau tactique, on doit décider de l'affectation de la pile à de certaines activités (par exemple à des navires). Au niveau opérationnel, on doit décider du conteneur à empiler et de sa position pour éviter les remaniements autant que possible. Notons la relation étroite de ce problème avec le problème d'arrimage déjà présenté.

Dans le PSC il s'agit aussi de stocker les conteneurs les uns au-dessus des autres dans le but de minimiser le nombre de mouvements parasites. Cependant, le problème d'empilement est considéré difficile du fait de l'incertitude à propos des conteneurs qui seront recherchés avant les autres.

Pour des conteneurs en import, cette incertitude existe parce que les véhicules transporteurs arrivent plus ou moins aléatoirement pour soulever un conteneur bien déterminé. Pour des conteneurs en export, on connaît d'habitude le navire avec lequel ils vont partir. Cependant, le plan d'arrimage n'est disponible que quelques instants avant le chargement et ainsi, la plupart des conteneurs sont déjà stockés dans la pile. Cela produit des mouvements inutiles qui ralentissent l'opération de chargement. Dans le cas où le plan d'arrimage serait disponible plus tôt, les conteneurs pourraient être réorganisés dans la zone d'export. Cela mène à un empilement " idéal " et donc, à moins de charge de travail pendant l'opération de chargement du navire. [KB98] décrit une approche à deux phases visant à minimiser le nombre de conteneurs à déplacer et à le faire en la plus courte distance. Bien qu'une telle approche de réorganisation semble très intéressante, il n'est pas souvent possible de procéder ainsi.

Les notions de base du PSC ainsi éclaircies, il s'avère alors important de définir ce problème, de donner la formulation la plus adoptée qui aidera à définir clairement les objectifs, les contraintes et les paramètres du problème, parcourir quelques problèmes analogues, et pour finir, fournir une synthèse bibliographique des différentes méthodes existantes pour sa résolution.

2.3.1 Définition du PSC

Après avoir parcouru maints travaux consacrés au PSC, nous adoptons la définition du PSC suivante, donnée dans [KPR00] : à l'arrivée d'un conteneur en import ou en export, il s'agit de décider presque en temps réel de son emplacement exact parmi les emplacements vides de manière à rendre efficace son chargement sur un navire, camion ou train. En général, la détermination d'un emplacement doit s'effectuer de manière à minimiser le nombre des mouvements parasites ou improductifs pouvant avoir lieu lorsqu'on veut rapprocher un conteneur éloigné ou extraire un conteneur se trouvant en dessous d'autres au moment de son départ pour être chargé sur le navire, train ou camion associé.

2.3.2 Formulation du PSC

Nous présentons une formulation du PSC basée sur la programmation dynamique proposée dans [KPR00]. La programmation dynamique initialement désignée pour améliorer l'efficacité de résolution de certains programmes mathématiques en les décomposant en sous programmes plus petits et par la suite plus faciles à résoudre. La programmation dynamique résout un problème donné en le décomposant en états, chacun est associé à une variable de décision.

Les calculs faits au niveau des différentes étapes sont liés de manière récursive de façon à mener à la solution optimale du problème entier. L'appellation mathématique a été donnée à cette procédure mathématique car elle a été principalement utilisée dans la résolution de problèmes qui nécessitant une décision dans le temps, ce qui est le cas de notre problème étudié.

En fait, selon plusieurs travaux de recherche à propos du problème de stockage de conteneurs, cette méthode de modélisation s'avère la plus adéquate pour exprimer l'objectif et les contraintes de ce problème, et surtout son aspect dynamique (fenêtres temporelles glissantes caractérisant l'arrivée et le départ des conteneurs). L'utilisation d'un modèle basé sur la programmation dynamique implique l'utilisation d'un modèle d'optimisation à plusieurs étapes sous incertitude (figure 2.4). C'est la raison pour laquelle qu'on utilise une fonction objectif récursive.

2.3.2.1 Définitions

La définition d'un état et d'une étape dans notre problème, caractéristiques principales d'un modèle de programmation dynamique, s'avère nécessaire :

Étape *stage* : le nombre d'emplacements vides dans une baie *bay* de la zone de stockage considérée ?

État *state* : l'état correspondant à chaque étape est constitué du nombre d'emplacements vides dans chaque pile et de la catégorie à laquelle appartient le conteneur de plus haute priorité dans chaque pile. Les piles sont triées dans l'ordre décroissant du nombre d'emplacements vides. Dans le cas où le nombre d'emplacements vides est le même pour différentes piles, la pile de plus haute priorité est celle contenant la plus grande catégorie (figure 2.3).

2.3.2.2 Modèle de formulation

Soient :

i : étant donnée une zone de stockage divisée en plusieurs baies, i indique le numéro d'une baie donnée.

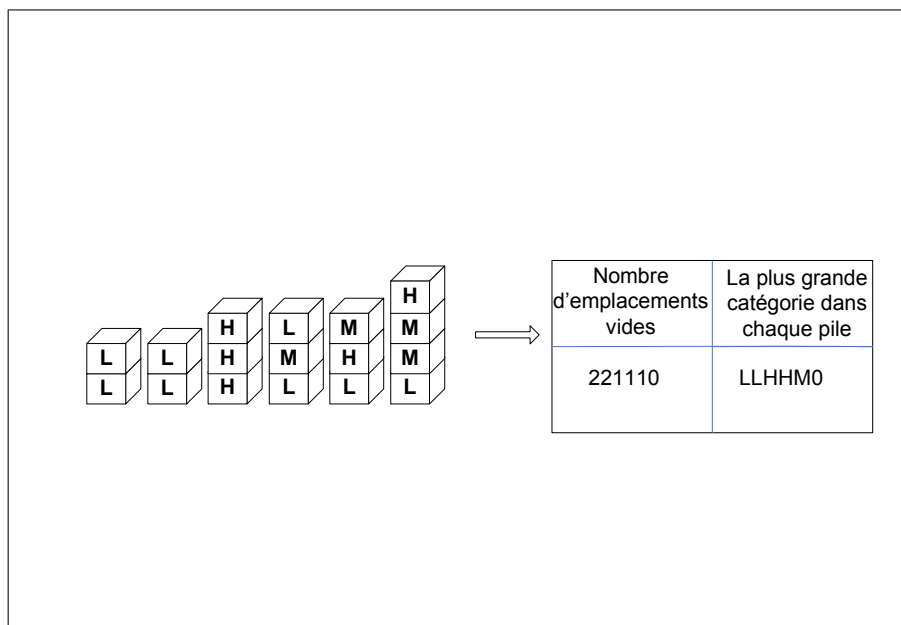


FIG. 2.3 – Illustration de la représentation d'un état

M : le nombre total de baies dans une zone de stockage.

N : le nombre total d'étapes.

n : le numéro de l'étape courante (le nombre d'emplacements vides).

X_n : l'état entrée de la $n^{\text{ème}}$ étape.

k_n : la catégorie du conteneur arrivant à la $n^{\text{ème}}$ étape (variable aléatoire).

D_n : le numéro $\{1, \dots, R\}$ de la pile allouée au conteneur arrivant à la $n^{\text{ème}}$ étape (variable de décision).

$P_n(k_n)$: la probabilité de l'arrivée d'un conteneur de catégorie k_n . Il est supposé que cette probabilité ne change pas tout au long du processus d'arrivée des conteneurs.

$r_n(X_n, D_n, k_n)$: le nombre marginal attendu de mouvements parasites cumulés à la $n^{\text{ème}}$ étape lorsqu'un conteneur de catégorie k_n est affecté à une pile D_n étant donné un état entrée X_n .

$t_n(X_n, D_n, k_n)$: la fonction de transfert d'état qui permet de mapper X_n, k_n et D_n à X_{n-1} .

$f_{i,n}(X_n)$: le nombre attendu de mouvements parasites dans la baie i dont l'état courant est X_n .

Y_i (variable de décision booléenne) = $\begin{cases} 1, & \text{Si un conteneur est placé dans la baie } i, \\ 0, & \text{Sinon.} \end{cases}$

$$\text{Minimiser } \sum_{i=0}^M Y_i f_{i,n}(X_n) \quad (2.1)$$

Où

$$f_{i,N}(X_N) = \sum_{k_N} P_N(k_N) \text{Min}_{D_N} [r_N(X_N, D_N, k_N) + f_{i,N-1}(X_{N-1})] \quad (2.2)$$

$$f_{i,0}(X_0) = 0 \quad (2.3)$$

$$X_{N-1} = t_N(X_N, D_N, k_N) \quad (2.4)$$

et S/C :

$$\sum_{i=0}^M Y_i = 1 \quad (2.5)$$

L'équation 2.1 exprime la fonction objectif : minimiser le nombre attendu total des mouvements de remplacement. Pour atteindre ce but, ne sont retenues que les zones avec le nombre minimum de mouvements parasites attendus.

Les équations 2.2 et 2.3 expriment la programmation dynamique utilisée pour modéliser la dynamique temporelle du problème, c'est une fonction récursive où la dernière étape est définie. Elle permet d'évaluer le nombre attendu de mouvements parasites pour une baie choisie, et de déterminer la meilleure pile pour stocker un conteneur qui arrive.

L'équation 2.4 exprime la fonction de transfert qui permet la mise à jour de l'état courant.

La contrainte 2.5 assure qu'une seule baie peut être choisie pour placer un nouveau conteneur qui arrive.

La résolution de ce problème d'optimisation se fait suivant le modèle d'optimisation, à plusieurs étapes, donné dans la figure 2.4.

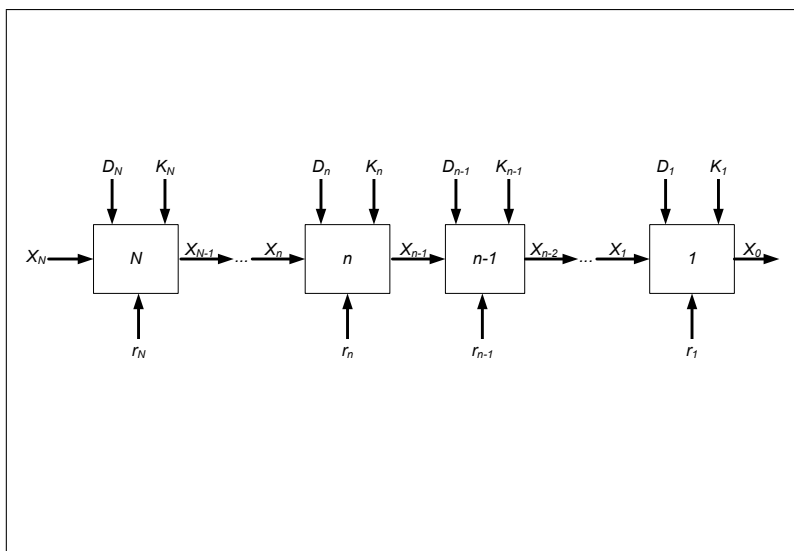


FIG. 2.4 – Modèle incertain d'optimisation à plusieurs étapes

2.3.3 Problèmes analogues

Dans cette section, nous essayons de situer le PSC par rapport à certains problèmes analogues abordés tels que le problème du monde des cubes et le problème de paquetage. Ces problèmes sont considérés comme étant des instanciations du problème classique d'ordonnement : le problème de planification. Dans cette vision, nous nous focalisons sur un problème connu d'intelligence artificielle : le monde des cubes.

2.3.3.1 Problème du monde des cubes

Le problème du monde des cubes dénoté *PMC Blocks World Planning Problem BWPP* est un problème d'intelligence artificielle Np-difficile et Np-complet [GS92] utilisé souvent pour illustrer le problème de planification. Dans le monde des cubes, le but est de convertir toutes les piles de cubes d'une configuration initiale à une configuration finale " But " constituée de cubes empilés les uns sur les autres (voir figure 2.5). Ils sont déplacés par un bras de robot qui ne peut déplacer qu'un cube à la fois du sommet d'une pile au sommet d'une autre (ou à la table pour constituer éventuellement une nouvelle pile). Ce cube ne doit pas être recouvert par un autre.

La table peut supporter un nombre infini de cubes, alors il n'y a aucune limite au nombre de piles qui peuvent être constituées.

Cette description est rencontrée souvent dans les travaux inhérents au problème du monde des cubes. Toutefois, selon [GS92], elle correspond à la version originale la plus connue de ce problème : *Elementary Blocks World-EBW*. Dans ce travail, nous retrouvons trois généralisations du EBW : *Varying block-size Blocks World-VBW*, *Limited table-capacity Blocks World-LBW*, et *Varying block-size, Limited table-capacity Blocks World-VLBW*. Parmi les méthodes de résolution du problème du monde des cubes, nous pouvons citer l'éco-résolution [Fer89, DD91]. Les algorithmes : *Unstack-Stack* (US), *Gupta-Nau* (GN1) [GS92], et *Gupta-Nau* (GN2) [GN91], comparés dans [ST95].

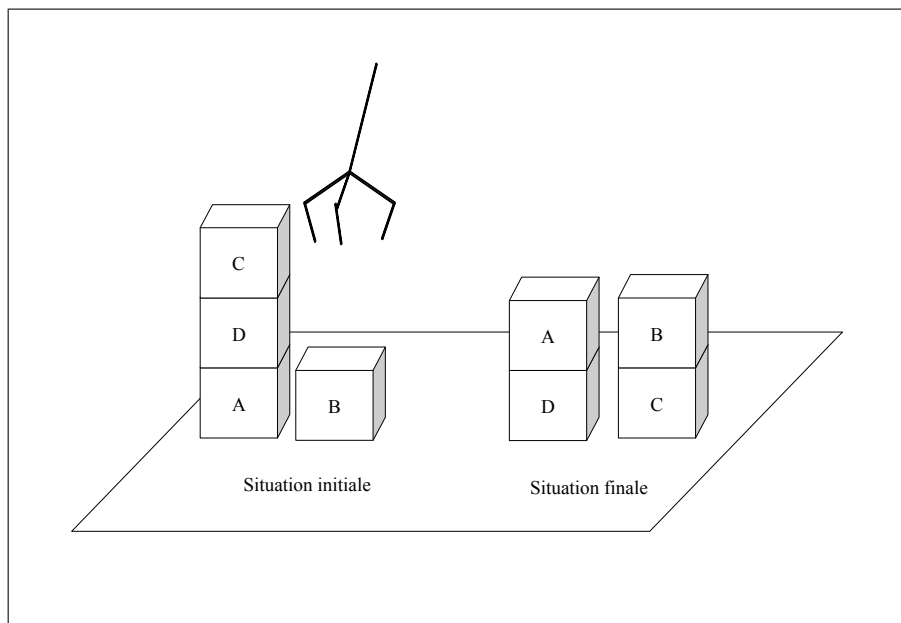


FIG. 2.5 – Exemple de manipulation des cubes

2.3.3.2 Analogie avec le Problème du monde des cubes

Suite à l'étude du PSC et du problème du monde des cubes, nous constatons une forte analogie entre les deux. En effet, Le PSC représente une instantiation particulière du problème du monde des cubes comme le montre la figure 3. Dans cette figure, une flèche définie de A vers B indique que B est une extension de A.

Cette analogie révélée nous permet, de situer le problème étudié par rapport à d'autres travaux, d'exploiter les méthodes appliquées déjà au problème du monde des cubes, qui ont montré une efficacité de modélisation et de résolution malgré qu'elles commencent à dater Et enfin, de supputer la complexité du PSC à une complexité exponentielle. Dès lors, le PSC est classé parmi les problèmes NP-complets et NP-difficiles.

2.3.4 Méthodes existantes pour la résolution du PSC

Dans la plupart des travaux inhérents, ce problème est traité de deux manières : calculs analytiques simplifiés ou études par simulation détaillées. La première est capable de traiter profondément les rapports entre les paramètres de ce

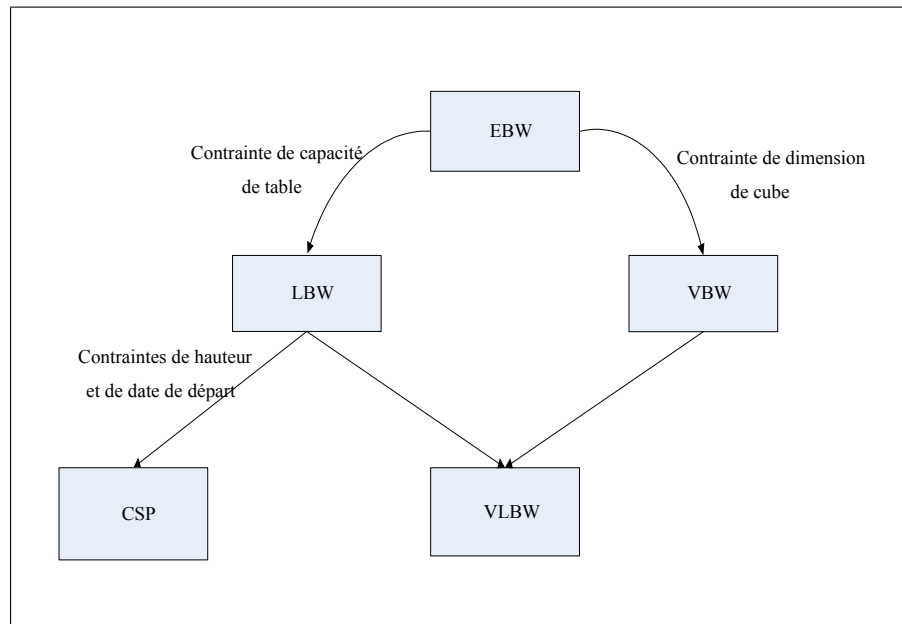


FIG. 2.6 – Analogie entre le PSC et le problème du monde des cubes

problème à un niveau abstrait. La deuxième peut entrer beaucoup plus dans les détails, mais elle prend du temps et seuls un nombre réduit de personnes la maîtrise. Aujourd’hui, aucune théorie d’empilement complète n’existe, et une bonne conception d’un empilement dépend non seulement des conditions de l’espace local, mais aussi des caractéristiques de l’information du flux entrant et sortant des conteneurs qui peuvent varier d’un port à un autre. Ci-dessous, nous donnons des exemples de travaux issus pour la majorité de ces approches.

Citons en d’abord les premiers : en 1985, McDowell et al.[McD85] analysent les problèmes de mouvements parasites de conteneurs en import et développent un modèle mathématique pour déterminer une configuration de stockage. En 1988, Chung et al.[CRM88] étudient une stratégie alternative pour l’opération de chargement dans laquelle un buffer de stockage est fourni devant l’aire de manœuvre *apron area* pour maximiser l’utilisation des équipements de soulèvement des conteneurs. Cette stratégie a été évaluée par simulation.

Sculli et Hui [SH88] étaient parmi les premiers chercheurs à développer des mesures pour la relation entre la hauteur d’empilement, l’utilisation (ou l’espace de rangement nécessaire) et les remaniements, et ce en appliquant une étude par

simulation complète.

Dans un travail datant de 1993, Taleb-Ibrahimi et al. [TICD93] proposent plusieurs stratégies pour l'allocation des emplacements de stockage disponibles aux conteneurs en export. Ils discutent la manière d'estimer les besoins en espace pour ces conteneurs et la manière d'en réduire les besoins en fournissant une zone temporaire de stockage *rough-stack* séparée de la zone de stockage *marshalling area* pour les conteneurs arrivés tôt. Ils considèrent un modèle cyclique des arrivées des navires et du stockage des conteneurs.

D'autres travaux comme ceux de De Watanabe (1991) plus populaires discutant de l'effet de différentes hauteurs d'empilement sont apparus dans le journal professionnel de systèmes de cargaison.

De Castilho et Daganzo [dCD93] continuent leurs études avec l'empilement de conteneurs en import : ils considèrent une stratégie qui garde la même hauteur pour les piles de conteneurs et une stratégie qui classe les conteneurs selon leurs dates d'arrivée.

En 1994, Kim et Kim proposent une méthodologie pour l'allocation de la zone de stockage aux conteneurs en export en considérant un modèle dynamique général des arrivées des conteneurs qui anticipe les efforts de soulèvement fournis par les équipements durant l'opération de chargement.

Dans [Kim97], Kim propose analytiquement et par simulation une discussion légèrement plus détaillée produisant des tables et des mesures considérant des piles de conteneurs de même taille.

Kim et Kim [KK98] ont étendu ces études en prenant aussi en considération le nombre des grues d'empilement. Ils ont développé un modèle de coût simple pour optimiser ce nombre en utilisant des approximations analytiques pour les diverses mesures de performance.

Une autre référence intéressante popularisée par Kim et Bea en 1998 [KB98] dont l'objectif est d'accélérer l'opération de chargement des conteneurs en export en minimisant les mouvements improductifs et les distances " parcourues " par ces conteneurs. Pour cela, ils proposent une méthode permettant d'atteindre un arrangement préférable à partir de l'arrangement courant des baies dans la zone de stockage tout en minimisant le nombre de conteneurs à déplacer et la distance parcourue qui s'en suit. Le problème est divisé en trois sous problèmes :

appariement des baies *bay matching*, planification des déplacements *move planning* et ordonnancement des tâches de déplacements *task sequencing*. Un modèle mathématique est utilisé pour la résolution de chacun de ces sous-problèmes.

En 1998, [GRZ98] présente un système d'aide à la décision pour la gestion d'un terminal à conteneurs intermodal. Dans leur modèle, Gambardella et al. traitent le problème d'affectation spatiale des conteneurs. Ils décrivent un premier modèle basé sur la programmation linéaire en nombres entiers pour optimiser le processus d'affectation, et un deuxième modèle de simulation à événements discrets du terminal.

Nous citons également le travail de Kozan en 1999 [KP99] utilisant les algorithmes génétiques pour réduire les temps de transfert et de manutention des conteneurs et par la suite le temps passé par un navire aux quais d'un port, et ce en accélérant les opérations de manutention. L'application de tels algorithmes présente des difficultés reposant sur le choix de bons paramètres tels que la taille d'une population, le nombre de génération, la probabilité de croisement et de mutation et les méthodes des opérateurs de reproduction.

La relation entre l'allocation de l'espace et le transport des conteneurs utilisant des véhicules est étudiée par Demir et al. [DLL⁺98] en ayant recours à l'optimisation combinatoire. Ils cherchent des stratégies de distribution de véhicules minimisant le temps total pour décharger tous les conteneurs du navire. Ils prouvent que ce problème est NP-dur et ressortent les plus mauvais ratios de performance absolus et asymptotiques relatifs à des heuristiques.

En 1999, Kim et al.[KK99] ont publié un article intéressant où ils traitent du problème d'allocation de la zone de stockage disponible aux conteneurs en import. Ils décrivent une stratégie de ségrégation. Dans cette stratégie, l'empilement de nouveaux conteneurs au dessus de conteneurs qui sont arrivés plus tôt est non permis. L'espace de stockage est alloué à chaque navire de manière à minimiser le nombre attendu total des mouvements de remplacement des conteneurs. Des modèles mathématiques et des procédures de résolution sont proposées pour l'obtention de solutions optimales dans trois cas différents : le taux d'arrivées des conteneurs en import est constant, cyclique, et dynamique. Une formule décrivant la relation entre la hauteur des piles de conteneurs et le nombre de mouvements parasites est dérivée.

Dans une autre étude, Dekker et al. [DVNM99] considèrent l'effet de générer des groupes de conteneurs échangeables au cours de l'empilement. Ils montrent qu'ayant de tels groupes, réduit considérablement non seulement le nombre de remaniements, mais aussi la charge de travail pendant le chargement du navire pouvant être également distribuée dans des systèmes automatisés. Malheureusement, de tels groupes échangeables sont seulement probables d'être générés pour des conteneurs en export, non encore pour des conteneurs en import. Pour ce dernier cas, l'empilement reste un problème rigoureux.

Chen et al. [CCH00] ont fourni une analyse statistique empirique de la performance réelle d'un terminal conteneurs Taiswanais. Le nombre de mouvements a été associé à la densité de stockage, au volume de conteneurs chargés et au volume de conteneurs déchargés.

Murty et al. [MJLZ⁺00] considèrent l'allocation de l'espace de stockage, tenant compte de la congestion qui peut apparaître si trop de véhicules transporteurs doivent soulever des conteneurs se trouvant dans une même zone. Ils proposent alors une méthode à deux phases : d'abord les conteneurs arrivants sont affectés aux différentes piles. Ensuite, chaque conteneur est mis dans sa meilleure position dans une pile, tenant compte de remaniements possibles.

Un travail intéressant popularisé aussi par Kim et al. en 2000 [KPR00] étudie le problème de détermination de positions de stockage de conteneurs en export, de manière à minimiser le nombre de mouvements de remplacement attendus lors de l'opération de chargement. Le poids est considéré comme un critère qui définit une certaine priorité entre les différents conteneurs à prendre en compte au cours de leur placement dans la zone de stockage. Ce travail fournit une étude comparative entre la résolution via un modèle de programmation dynamique et la résolution via le développement d'un arbre de décision.

Nous pouvons citer celui de Kozan [Koz00] qui étudie le problème de minimisation du temps total passé par un conteneur dans un terminal depuis son arrivée jusqu'à son départ (ce temps est égal à la somme du temps de manutention et du temps d'acheminement ou de déplacement des conteneurs). Il propose une formulation mathématique, et utilise le programme GAMS (*Generalised Algebraic Modeling System-GAMS*) pour le résoudre.

Un autre travail dédié au problème de stockage est proposé par Duinkerken et

al. en 2001 [DEO01], il fournit une étude empirique et comparative par simulation des différentes stratégies de stockage des conteneurs dans un terminal selon plusieurs paramètres.

Ambrosino et al. [AMS02] étudient l'impact de l'organisation de la zone de stockage dans l'activité d'arrimage de conteneurs en terme de mouvements improductifs des conteneurs en export. Ils traitent le problème en utilisant une approche heuristique basée sur une modèle de programmation linéaire en 0-1.

En 2002, Kim et al. [KK02] proposent une méthode pour déterminer l'espace de stockage optimal et le nombre optimal de grues de transfert utilisées pour la manutention des conteneurs en import. Un modèle de coût est développé pour l'aide à la décision. Ce modèle de coût est constitué du coût de stockage, du coût d'investissement en grues de transfert et du coût des opérations effectuées par les camions et les grues de transfert.

Un travail présenté dans Chen et al. [CFLR04] propose un modèle dans le cadre de la gestion du port de Singapour. Ce modèle allie différentes métaheuristiques (recherche tabou, recuit simulé et algorithmes génétiques) pour résoudre le problème d'optimisation de la zone de stockage dans un port *Port Yard Storage Optimization-PYSOP* par analogie au problème de paquetage à deux dimensions *two-dimensional Packing Problem* et ce en minimisant l'espace alloué à un navire en un intervalle de temps défini.

Toujours, par analogie au problème de paquetage, une étude plus récente détaillée dans Murty et al. [MLWL05] propose un système d'aide à la décision pour la minimisation du temps d'allocation des navires aux postes à quai, des ressources nécessaires pour la manutention, du temps d'attente des camions et des navires, et principalement la congestion dans les travées et les zones de stockage pour l'optimisation de l'utilisation de l'espace de stockage. Et ce, en minimisant le nombre de mouvements parasites de conteneurs en import et en export. Ce système est dédié au port de Hong Kong, il met en œuvre des modèles de programmation linéaire. Notons que ce système ne considère pas le problème de flux et d'ordonnancement de conteneurs.

La majorité de ces études existantes a été menée pour des ports maritimes particuliers (Singapour, Hong Kong, etc.). Dans le cadre des ports fluviaux, nous citons deux travaux récents [KP04] et [YHSK05] où ils utilisent un algorithme

de programmation dynamique basé sur une loi d'arrivée stochastique des barges pour résoudre le problème d'affectation en temps réel de conteneurs déchargés à des zones de stockage tout en minimisant le nombre de mouvements parasites. De nombreuses simulations ont été réalisées, en faisant varier la loi d'arrivée des barges, l'horizon de prévision, et le taux de remplissage de la zone de stockage. Une étude comparative a été élaborée en comparant le nombre de mouvements parasites obtenu pour cet algorithme, et pour une règle de décision simple consistant à placer le conteneur sur la pile la moins occupée, pour différents taux d'occupation de la zone de stockage.

Dans un travail récent [DVvA06], les auteurs étudient et évaluent par simulation de différentes stratégies de stockage de conteneurs spécifiques au terminal *ECT* du port de Rotterdam. Les mesures de performance sont le nombre de mouvements parasites et le pourcentage d'utilisation des grues d'empilement automatiques.

Les modèles décrits, dont la plupart sont analytiques ou de simulation, constituent des modèles centralisés. Nous avons retrouvé en opposé des modèles de simulation distribuée utilisant les systèmes multi-agent dont nous pouvons citer : [RJCB00, KD04, HWD03]. Par exemple, le travail récent de Henesey et al. [HWD03] décrit trois manières de stockage : un système de stockage à court terme pour les conteneurs qui peuvent être transbordés dans un autre navire, un système de stockage à long terme pour les conteneurs en attente d'inspection, et un système de stockage particulier pour les conteneurs frigorifiques.

Ces travaux proposent des modèles multi-agent simulant le port en entier. Il s'agit de définir un agent pour chaque opération du processus de la gestion portuaire.

Notre recherche bibliographique sur des problèmes voisins au PSC a révélé des travaux intéressants utilisant aussi les systèmes multi-agent telles que : la méthode d'éco-résolution appliquée au problème classique du monde des cubes [Fer95] et la résolution du problème d'affectation de tâches aux ressources [GV92, Ghe99, PCB06].

2.4 Classification des travaux selon les ports conteneurisés dans le monde

Dans cette section, nous donnons un tableau 2.2 qui résume les recherches récentes consacrées aux problèmes de décision rencontrés dans quelques uns des plus grands ports conteneurisés dans le monde en se référant à [RT]. La première et troisième colonnes de ce tableau montrent le nom du port conteneurisé et les références des travaux dédiés qui sont extraits de la littérature. La deuxième colonne présente les problèmes de décision discutés dans chaque port ainsi que les méthodes de résolution adoptées. Nous constatons que le problème le plus étudié dans ces ports est le problèmes de tournées de véhicules.

2.5 Bilan et synthèse

Dans ce chapitre, nous avons décrit successivement les opérations se produisant dans un terminal à conteneurs manuellement ou automatiquement géré. Tous les équipements mis en œuvre ainsi que les problèmes de décision inhérents sont aussi discutés, une revue de littérature est présentée pour chaque problème. Dans une deuxième partie nous nous sommes concentrés sur un problème particulier, celui que nous traitons dans cette thèse : Le problème de Stockage de Conteneurs PSC. Nous avons présenté d'abord nos motivations pour étudier ce problème, et les notions de base sur lesquelles il repose. Puis, une définition du problème en lui-même. Ensuite, un modèle de sa formulation par la programmation dynamique. Par la suite, le problème analogue : le problème du monde des cubes. Et enfin, les travaux de recherche qui lui ont été consacrés proposant des modèles analytiques et de simulation, centralisés et distribués.

Ce chapitre nous a permis essentiellement d'avoir une vue générale sur les méthodes de résolution dédiées à l'activité de stockage de conteneurs. Nous avons décelé les limites suivantes par rapport aux objectifs et contraintes que nous nous sommes fixés :

- Les travaux dédiés au PSC utilisent pour la plupart des modèles mathématiques et stochastiques pour sa résolution qui ne tiennent pas compte de la complexité, la dynamique et l'imprévisibilité de ce problème et se re-

TAB. 2.2 – Principaux travaux dans les plus grands ports du monde

Ports	Problème de décision et Méthodes de résolution	Références
1. Port de Hambourg, Allemagne 2. Port de Bremen, Allemagne	* Allocation de l'espace de stockage (<i>Mixed Integer Linear Programming-MILP</i>) * Génération de scénarios (simulation) * Ordonnancement de véhicules (algorithmes génétiques)	[SWZ01] [Har02] [BRVS00]
1. Contship La Spezia, Italie 2. Terminal maritime à Gênes, Italie	* Allocation de l'espace de stockage (Simulation) * Gestion de la zone de stockage (<i>Integer Programming</i>)	[GRZ98] [AMS02]
Port de pusan, Corée	* Allocation de postes à quais (<i>MILP</i>) * Allocation de postes à quais et affectation de portiques (Relaxation lagrangienne, Programmation dynamique)	[Moo01] [PK03]
Port de Rotterdam, Les Pays-Bas	* Ordonnancement de véhicules et de grues (<i>B&B, Beam Search Heuristic Method</i>) * Prediction et évitement de congestion (<i>Wait and Proceed Strategy</i>)	[MW01b] [MW01a] [MHGWCCP03]
Port de Singapour, Singapour	* Le système global (Simulation) * Distribution d' <i>AGVs</i> (<i>Network Flow Model</i>)	[LJI02] [CSN ⁺ 03]
Terminal à conteneurs No9 de Hong Kong (Nouveau), Chine	* Déploiement de <i>RTGs</i> (<i>Vogel Solution</i>) * Affectation de l'espace de stockage et tournées de véhicules (Programmation linéaire) * Allocation de l'espace de stockage (<i>Integer Programming</i>) *Déploiement de grues et de <i>RTGs</i>	[MLWL05] [ZLW ⁺ 01] [ZWLL02]
1. Terminal de taille réelle 2. Terminal virtuel	* Ordonnancement de véhicules (Algorithme heuristique/ relaxation lagrangienne) * Re-Ordonnancement de portiques et de véhicules (Système multi-agent)	[ZYHH02] [TH02]
Port réel (non mentionné) [TH02]	Le système global (Système multi-agent, non implémenté)	[RJCB00]

streignent à traiter le cas des arrivées des conteneurs en import ou en export exclusivement.

- Insuffisance de l'exploitation de méthodes de résolution distribuée comme les systèmes multi-agent, les quelques travaux élaborés sont uniquement simulés et non implémentés. De plus, ils simulent toute l'activité portuaire. Aucun travail ne s'est consacré à l'activité particulière de stockage de conteneurs.
- Peu de travaux s'intressent à optimiser le nombre de mouvements improductifs. Dans les quelques travaux munis d'une étude expérimentale détaillée, les mesures de performances sont souvent : le pourcentage d'utilisation des grues de quais ou le temps total de transfert de conteneurs.
- L'utilisation des méthodes exactes et en particulier la programmation dynamique a concerné l'activité de stockage des conteneurs arrivés et ne tiennent pas compte des départs de conteneurs en import ou en export. En outre, ces méthodes supposent une sous-structure optimale et supposent connues à l'avance l'arrangement initial et final des conteneurs dans une zone de stockage
- Absence d'études expérimentales pour le cas de minimisation de mouvements improductifs suite à des départs de conteneurs.
- Dans ces travaux, nous n'avons pas trouvé des benchmarks que nous pouvons utiliser pour mener notre étude expérimentale.
- Comme nous le constatons à partir du tableau 2.2, les travaux dédiés aux plus grands ports du monde ne traitent pas notre problème : Le PSC.
- L'analogie entre le problème de stockage de conteneurs et le problème du monde des cubes est faiblement exploitée.

Au vue des ces limites, il nous semble que certaines voies de recherche telles que les méthodes heuristiques, les méthodes de résolution de problèmes analogues et l'approche multi-agent peuvent se révéler fructueuses, particulièrement, en ce qui concerne la modélisation, la simulation et la résolution de problèmes d'optimisation combinatoire comme le PSC. C'est pourquoi nous avons décidé de développer deux modèles multi-agent : un premier modèle de base appelé ContEco inspiré de la forte analogie entre une version relaxée du PSC et le problème du monde des cubes, et principalement un deuxième modèle étendu appelé COSAH pour l'optimisation heuristique distribuée d'une version étendue du PSC. Les spécificités de

ces deux contributions seront exhaustivement détaillées dans le chapitre 4. Mais avant cela, dans le chapitre 3, nous présentons les méthodes et les concepts adoptés dans ces deux modèles. Nous donnons également, de manière plus synthétique nos arguments d'utiliser chaque méthode.

Deuxième partie

Approche proposée

Chapitre 3

Méthodes et approches adoptées

3.1 Introduction

Dans les chapitres précédents, nous avons dressé une étude bibliographique exhaustive concernant les problèmes de transport conteneurisé, le problème particulier de stockage de conteneurs- PSC, les problèmes analogues, et les méthodes de résolution qui leur ont été dédiées. Nous avons conclu par un travail de synthèse et critique de cet état de l'art qui nous a permis de situer notre thèse au sein des travaux existants, de montrer certaines limites et insuffisances quant aux approches de résolution du problème étudié, de mettre en avant son analogie avec le problème connu du monde des cubes, et en conséquence de justifier nos choix et directives quant à la modélisation, la simulation et la résolution de ce problème. Dans le reste de ce manuscrit nous allons alors nous intéresser aux différents méthodes et concepts adoptés dans nos deux modèles proposés : ContEco et COSAH, à savoir :

- Les systèmes multi-agent : étant une méthode de modélisation, simulation et résolution distribuée.
- L'optimisation heuristique : étant une méthode d'optimisation approchée.
- Les réseaux de Petri : étant des méthodes formelles pour la description et la formalisation.

Nous justifions également l'utilisation de chaque méthode et expliquons la nature et l'utilité de leur alliance par rapport à nos modèles proposés.

3.2 Les systèmes multi-agent

Les deux modèles proposés ContEco et COSAH sont deux modèles de résolution distribuée basés sur les systèmes multi-agent et que nous utilisons pour la modélisation, la simulation et la résolution du problème de stockage de conteneurs-PSC. Dans cette section, nous commençons d'abord par justifier notre intérêt à cette méthode en citant nos différents arguments, ensuite définir les différentes notions liées à cette méthode, exceptionnellement : l'éco-résolution, la coopération, la négociation, et la planification qui constituent les concepts clés de ContEco et COSAH. Enfin, nous abordons les différents aspects de la formalisation des systèmes multi-agent.

3.2.1 Pourquoi avons-nous opté pour les systèmes multi-agent ?

Nous donnons des raisons liées d'une part à la nature même du PSC, et d'autre part à l'application en elle-même des systèmes multi-agent :

- Un haut degré de dynamique et de complexité caractérise le PSC étant données des contraintes temporelles dynamiques, des fenêtres glissantes caractérisant les arrivées de navires (ou autre mode) et de nouveaux conteneurs peuvent être reçus de manière asynchrone. Comparant à d'autres problèmes tel que le problème d'affectation de tâches aux ressources, le problème de sac à dos, et le problème du monde des cubes. Bien qu'ils soient de même complexité, notre problème est plus difficile à résoudre empiriquement, puisque les contraintes temporelles dynamiques qui le caractérisent réduisent l'espace des solutions réalisables et les rendent plus difficiles à trouver dans certains cas. En outre, puisqu'il est dynamique, il ne s'agit plus de calculer une solution à partir de données stables mais de maintenir une solution acceptable malgré la réorganisation et les perturbations permanentes.
- Le problème de stockage de conteneurs est modélisé par un système complexe et ouvert. Il peut se rattacher d'autres systèmes tels que le problème de chargement des navires et le problème d'ordonnancement des équipements de manutention. De plus, il se caractérise, d'une part, par la nature

matérielle de ses éléments (conteneurs, piles, équipements, navires, etc.), et d'autre part, par la richesse des interactions entre eux.

- La nécessité d'existence de traitements coopératifs dans la réalité du système puisque deux conteneurs peuvent entrer en interférence, soit au niveau des buts (nouvel emplacement), des résultats (mouvements effectués), soit au niveau des procédures (détermination de nouveaux emplacements). Alors, il faut en sorte traiter ces interférences pour que l'activité de stockage d'un conteneur soit réalisée de façon à faciliter la réalisation de celle de l'autre ; ou la réalisation de la tâche commune (si elle existe) comme par exemple : libérer leur emplacement au profit d'un troisième conteneur. Cela nous mène à l'utilisation de méthodes de coopération empruntées à l'IAD telles que : la négociation et la planification distribuée entre agents.
- Le problème de stockage de conteneurs-PSC s'intègre dans le cadre des approches de résolution distribuée coopérative de problèmes *Cooperative Distributed Problem Solving-CDPS* [DLC89]. Dans ces approches, les agents coopèrent pour résoudre des problèmes qu'ils ne peuvent pas résoudre individuellement. Tout d'abord chaque agent utilise ses connaissances et ressources pour résoudre localement un ou plusieurs sous-problèmes. Les solutions partielles à ces sous-problèmes sont par la suite intégrées pour former la solution globale. Dans le cas du PSC, les conteneurs qui recherchent des emplacements ou bien qui veulent partir ont différentes perceptions de leur espace environnant constitué par les autres conteneurs, doivent coopérer pour résoudre leurs sous-problèmes et conflits pour atteindre une solution globale.
- La recherche dans le domaine des systèmes multi-agent revêt de plus en plus d'importance pour leur capacité à aborder des systèmes complexes c'est à dire les systèmes constitués de nombreux composants en interaction dynamique entre eux et avec le monde extérieur.
- En ce qui concerne la résolution distribuée via les systèmes multi-agent, nous avons constaté que les quelques travaux inhérents l'utilisent seulement pour simuler la gestion portuaire en entier.
- D'autres travaux ont montré que l'application des systèmes multi-agent est d'un grand apport dans la résolution de problèmes fortement combinatoire

(NP-complets et NP-Difficiles). tels que le problème de tournées de véhicules [BFG05], le problème de partitionnement de graphes [HG05], le problème d'ordonnancement [EG05], etc.

En vue de toutes ces raisons, les systèmes multi-agent constituent une réponse à notre problématique.

3.2.2 Quelques concepts

Depuis une dizaine d'années, les systèmes multi-agent ont connu un grand essor et sont appliqués à des domaines très variés comme, par exemple, le domaine de la simulation et de la vie artificielle, la robotique, le traitement d'images, etc. Cet essor manifeste des SMA émane de la critique des approches séquentielles ou fonctionnelles et de l'intelligence artificielle classique. En résumé, il est intéressant d'observer la place qu'occupent les systèmes multi-agent dans l'évolution des paradigmes de l'informatique selon [Jam05]. La figure 3.1 illustre cette place.

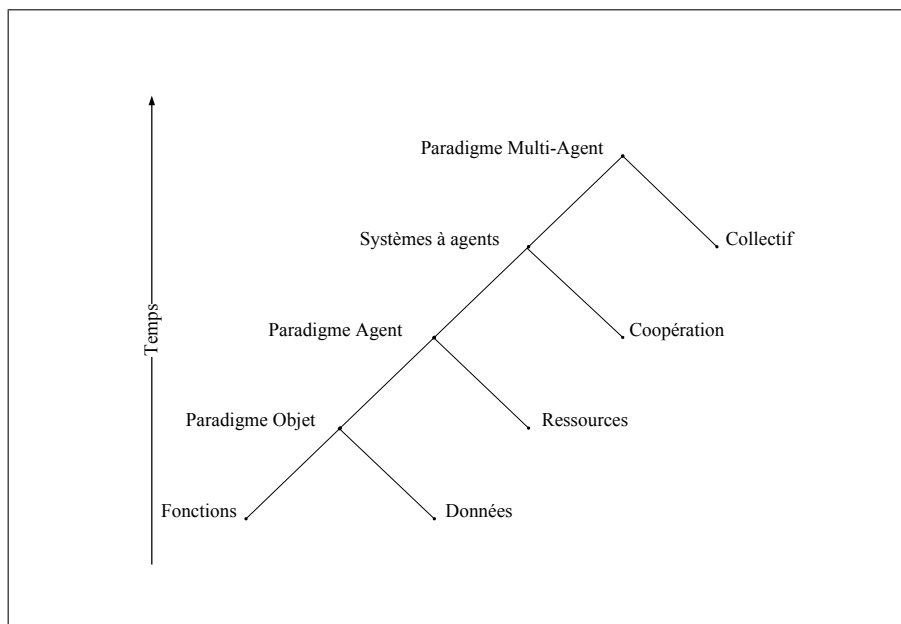


FIG. 3.1 – Evolution des paradigmes de programmation

Dans la suite, nous présentons les notions d'agents et de systèmes multi-agent, et détaillons les différentes questions que soulève l'approche SMA telles que : les

interactions et la coopération, la coordination, la planification et la communication. Nous insistons en particulier sur la coordination par éco-résolution [Fer95] et [Ghe99].

3.2.2.1 Qu'est ce qu'un agent

Il n'existe pas, actuellement, une définition de l'agent qui fasse foi dans le monde de l'intelligence artificielle distribuée. Il est donc nécessaire, pour avoir une bonne vision de ce concept, de confronter plusieurs de ces définitions, nous allons exploiter trois d'entre elles :

La définition de Jacques Ferber [Fer95] : " Un agent peut être défini comme une entité autonome, réelle ou abstraite, qui est capable d'agir sur elle même et sur son environnement, qui, dans un univers multi-agent, peut communiquer avec d'autres agents, et dont le comportement est la conséquence de ses observations, de ses connaissances et des interactions avec les autres agents ".

Cette définition aborde une notion essentielle : l'autonomie. En effet, ce concept est au centre de la problématique des agents. L'autonomie est la faculté d'avoir ou non le contrôle de son comportement sans l'intervention d'autres agents ou d'êtres humains. Une autre notion importante abordée par cette définition concerne la capacité d'un agent à communiquer avec d'autres.

Selon M. Wooldridge [Woo99], un agent est un système informatique capable d'agir de manière autonome et flexible dans un environnement. Par flexibilité on entend :

- Réactivité : un système réactif maintient un lien constant avec son environnement et répond aux changements qui y surviennent.
- Pro-activité : un système pro-actif (aussi appelé téléonomique) génère et satisfait des buts. Son comportement n'est donc pas uniquement dirigé par des événements.
- Capacités sociales : un système social est capable d'interagir ou coopérer avec d'autres systèmes.

Une définition plus récente donnée par [Wei99] : un agent est " une entité computationnelle " ou un robot, qui peut être vu comme percevant et agissant de façon autonome sur son environnement. On peut parler d'autonomie parce que son comportement dépend au moins partiellement de son expérience.

TAB. 3.1 – Typologie des agents

Agents cognitifs	Agents réactifs	Agents hybrides
<ul style="list-style-type: none"> * possèdent des capacités cognitives : anticiper, prévoir le futur, mémoriser des choses... ils réfléchissent. * proviennent d'une métaphore du modèle humain. * possèdent une représentation explicite de leur environnement, des autres agents et d'eux-mêmes. * sont aussi dotés de capacités de raisonnement et de planification ainsi que de communication. * sont structurés en société où il règne donc une véritable organisation sociale. * le travail le plus représentatif de cette famille d'agent porte sur le modèle <i>Believe Desir Intention BDI</i> [RG95]. * les sources de ces travaux sont les sciences humaines et sociales. 	<ul style="list-style-type: none"> * ont des capacités réactives : réagir directement à l'environnement perçu, par pulsion. * ne possèdent pas de moyen de mémorisation. * n'ont pas de représentation explicite de leur environnement. * fonctionnent selon un modèle stimuli/ réponse. * dès qu'ils perçoivent une modification de leur environnement, ils répondent par une action programmée. * l'exemple le plus célèbre est celui de la fourmilière étudié par Alexis Drogoul [Dro00]. * les sources des travaux sur ce type d'agents sont les sciences de la nature et de la vie. 	<ul style="list-style-type: none"> * ont des capacités cognitives et réactives. * conjuguent en effet la rapidité de réponse des agents réactifs ainsi que les capacités de raisonnement des agents cognitifs. * le modèle est un compromis autonomie/coopération et efficacité/complexité. * l'exemple de l'architecture ASIC [BD97] utilisée pour le traitement numérique d'images. * l'architecture ASTRO [DO98] développée pour être utilisée dans les systèmes multi-agents soumis à des contraintes de type temporelles.

Ces définitions sont le résultat de différentes approches. Il existe en fait plusieurs types d'agents, qui selon les capacités qu'ils possèdent, seront qualifiés de réactifs, cognitifs ou hybrides. Ces types d'agents sont décrits dans le tableau 3.1.

3.2.2.2 Qu'est ce qu'un Système Multi-Agent SMA ?

Ferber [Fer95] définit un système multi-agent de la manière suivante :

- " Un système multi-agent est un système composé des éléments suivants :
- Un environnement est un espace disposant généralement d'une métrique.
 - Un ensemble d'objets situés dans l'espace, ils sont passifs, ils peuvent être perçus, détruits, créés et modifiés par les agents.
 - Un ensemble d'agents qui sont les entités actives du système.
 - Un ensemble de relations qui unissent les objets entre eux.
 - Un ensemble d'opérations permettant aux agents de percevoir, de détruire, de créer, de transformer et de manipuler les objets.
 - Un ensemble d'opérateurs chargés de représenter l'application de ces opérations et la réaction du monde à cette tentative de modification (les lois de l'univers). "

Un système multi-agent est un ensemble homogène ou hétérogène d'agents situés dans un même environnement et qui interagissent. De ces interactions peuvent émerger différents phénomènes que l'on peut classer en trois catégories [Mar96] : l'émergence structurelle, l'émergence de comportements et l'émergence de propriétés. L'émergence traite donc de l'apparition soudaine non programmée et irréversible de phénomènes dans un système (éventuellement multi-agent) confirmant ainsi la maxime grecque " le tout est plus que la somme de chaque partie ".

Selon la vision de Demazeau [Dem95] :

SMA = Agent + Environnement + Interaction + Organisation.

Cette décomposition d'un SMA est dite décomposition Voyelles qui identifie un axe Agent, un axe Environnement, des Interactions et une structure Organisationnelle explicite ou non. Un SMA est généralement caractérisé par :

- chaque agent a des informations ou des capacités de résolution de problèmes limitées, ainsi il a un point de vue partiel.
- il n'y a aucun contrôle global du système multi-agent.
- les données sont décentralisées.
- le calcul est asynchrone.

Les types courants d'interaction dans un SMA incluent la coopération (travailler ensemble à la résolution d'un but commun), ; la coordination (organiser la résolution d'un problème de telle sorte que les interactions nuisibles soient évitées ou que les interactions bénéfiques soient exploitées) et la négociation (parvenir à

un accord acceptable pour toutes les parties concernées).

La recherche sur les systèmes multi-agent est ainsi une recherche sur :

3.2.2.3 La communication

Le problème de la communication entre agents peut se résumer aux questions suivantes : communiquer quoi, à qui ? quand ? Et comment ?

La communication est primordiale dans le sens où elle constitue un moyen pour acheminer et véhiculer les messages de toutes sortes qui permettent de résoudre les conflits de coopération et de coordination.

Plusieurs modèles de communication sont proposés en fonction du type de coordination entre agents. La plupart de ces modèles disposent de protocoles permettant un traitement logique de la communication en tant qu'acte. On distingue trois modèles de base que voici.

Communication par environnement

Les agents laissent des traces dans leur environnement (des signaux) indiquant leur présence et pouvant être perçues par d'autres agents. Ce type de communication est très courant dans les systèmes réactifs et qualifiés de communication non intentionnelle.

Communication par partage d'information ou blackboard (ou à contrôle centralisé)

Tous les agents travaillent sur un espace commun qui comprend toutes les informations nécessaires à la résolution du problème traité. Toute communication entre agents se fait par le biais de cet espace appelé base partagée ou tableau noir, d'où l'appellation du système *Blackboard*. Notons que ce mode de communication convient bien aux agents cognitifs. C'est un mode de communication intentionnelle qui a donné lieu à des architectures Blackboard où les agents sont plutôt appelés des sources de connaissances.

Communication par envoi de messages (ou à contrôle distribué)

Contrairement au modèle précédent, chaque agent a un point de vue local du problème et communique directement avec les autres agents qu'il connaît appelés

accointances. Ce type de communication est utilisé aussi bien pour les agents réactifs que les agents cognitifs et est qualifié aussi d'intentionnel.

Notons qu'il y a deux variantes de ce modèle : dans le cas où on connaîtrait le destinataire, il s'agit d'une communication point à point. Dans le cas contraire, c'est une communication par diffusion.

3.2.2.4 La coopération

Les agents travaillent à la satisfaction d'un but commun par le biais de l'échange d'informations ou individuel via la sous-traitance des buts.

L'objectif de la coopération est d'améliorer le mode de travail des agents en terme de :

- Validité et rationalité des informations échangées et des comportements.
- Efficacité des stratégies de résolution employées.
- Cohérence entre la planification locale et globale.
- Rééquilibrage dynamique de la charge de travail.

On distingue deux modèles de coopération : Collaboration par allocation des tâches, des informations et des ressources et Coordination d'actions et résolution de conflits.

Collaboration par allocation des tâches, des informations et des ressources

Cette allocation peut être statique c'est à dire que chaque agent connaît les compétences des autres agents (exp. : réseau d'accointances) ou dynamique où tous les agents sont contactés via différents types de messages, sans connaissances à priori, afin de choisir le " meilleur " (exp. : réseau contractuel [Smi80, SD80]). Cette dynamique fournit un système robuste reposant sur un espace de recherche robuste reposant sur un espace de recherche maximal.

Coordination d'actions et résolution de conflits

La résolution de conflits est mise en oeuvre soit par négociation, soit à l'aide d'un médiateur. Le moyen le plus usité et le plus étudié est la négociation. C'est l'activité qui consiste à échanger des informations entre les agents afin d'aboutir à un compromis mutuellement acceptable et un comportement convergent vers une

solution ou un état stable dans le cas de conflits. Elle est accomplie par échange de plans, de croyances ou d'informations, par engagements, par coordination d'actions ou par échange de solutions partielles.

Malone [Malone 90] note que deux des composantes fondamentales de la coordination entre agents sont l'allocation de ressources rares et la communication de résultats intermédiaires. Dans ce contexte, les agents doivent être capables de communiquer entre eux de façon à pouvoir échanger les résultats intermédiaires. Pour l'allocation des ressources partagées, les agents doivent être capables de faire des transferts de ressources. Ceci peut d'ailleurs imposer certains comportements à des agents particuliers. Le lecteur, intéressé par une étude détaillée de la coordination entre agents, pourra se référer à l'article de El-Fallah Seghrouchni [SH96]. Dans cet article, la coordination est vue plus particulièrement sous l'angle de la coordination d'actions par planification multi-agent. La coordination peut être assurée par la méthode d'éco-résolution.

3.2.2.5 La négociation

La négociation joue un rôle fondamental dans les activités de coopération en permettant aux personnes de résoudre des conflits qui pourraient mettre en péril des comportements coopératifs. Durfee et ses collègues [DLC89] définissent la négociation comme le processus d'améliorer les accords (en réduisant les inconsistances et l'incertitude) sur des points de vue communs ou des plans d'action grâce à l'échange structuré d'informations pertinentes. Le protocole du réseau contractuel *Contract-Net* a été une des approches les plus utilisées dans un SMA [Smi80]. Par exemple, Parunak (Parunak 96) a utilisé ce protocole pour développer un système de contrôle de production. Un autre protocole de négociation important a été proposé par Cammarata et ses collègues [CMS88] qui ont étudié les stratégies de coopération pour résoudre des conflits entre des plans d'un ensemble d'agents. Ces stratégies ont été appliquées au domaine du contrôle de trafic aérien.

3.2.2.6 Planification

Bond et Gasser [BG88] font remarquer que nous pouvons obtenir une plus grande coordination si on oriente les comportements des agents vers des buts communs en établissant explicitement une division du travail entre les agents. Des techniques comme la planification centralisée pour des groupes d'agents, la conciliation de plans, la planification distribuée, l'analyse organisationnelle, sont toutes des façons d'aider les agents à aligner leurs activités en assignant les tâches après avoir raisonné sur les conséquences de réaliser ces tâches dans des ordres particuliers. En planification multi-agent centralisée, un agent est responsable pour la création du plan qui spécifie les actions planifiées pour tous les agents concernés. C'est cette approche qui a été utilisée par Cammarata pour le problème du contrôle de trafic aérien. Une autre façon d'implémenter la planification multi-agent centralisée a été proposée par Georgeff [Geo83]. Dans cette approche, les plans des agents sont d'abord créés de façon individuelle, ensuite un agent centralisateur rassemble ces plans et les analyse pour identifier les conflits. L'agent centralisateur en question essaye de résoudre les conflits en modifiant les plans locaux des autres agents et en introduisant des commandes de communication afin que les agents se synchronisent de façon appropriée. Dans une approche de planification distribuée, les activités de planification sont réparties au sein d'un groupe d'agents. Cette approche est utilisée quand un seul agent ne peut pas avoir une vue globale des activités du groupe.

En général, la planification multi-agent nécessite une forme ou une autre de synchronisation de plans qui peut être réalisée à divers moments : pendant la décomposition de plans, pendant la construction de plans ou après celle-ci. Les plans des agents peuvent être en conflits en raison d'incompatibilités d'états des systèmes, de l'ordre des activités ou de l'usage des ressources. De tels conflits peuvent être résolus par un agent en particulier (coordinateur ou médiateur) ou une solution peut être obtenue par négociation.

3.2.3 Résolution par coordination : l'éco-résolution

Selon [Fer95] et [Dro00], l'éco-résolution est une méthode de résolution de problème qui prend le contre-pied des techniques classiques. Il s'agit de refor-

muler le problème de manière qu'il soit conçu comme un ensemble d'agents en interaction qui tentent de satisfaire individuellement leur propre but, plutôt que de le formaliser de manière globale et de définir par la suite des méthodes de résolution s'appliquant directement à sa définition. Elle s'est avéré être un outil extrêmement puissant de décomposition de problème. Sa proximité, d'une part, avec la décomposition orientée objet, et la simplicité, d'autre part, de son noyau de résolution (et des comportements afférents des agents constitutifs du problème) permettent d'aborder une gamme très étendue de problèmes, et d'utiliser un grand nombre de techniques complémentaires (apprentissage, résolution de contraintes, propagation de gradients) pour l'écriture d'heuristiques spécialisées. Le principe est d'appréhender un problème comme étant une population d'agents dont l'ensemble des comportements tend à aboutir à un état stable, que l'on appelle la solution du problème. Chaque agent répond aux principes d'autonomie et de localité, c'est-à-dire que les actions qu'il entreprend sont la conséquence à la fois de perceptions locales et des relations qu'il a établies avec d'autres agents. Au cours de leur comportement, ces agents effectuent parfois des actions particulières qui, si elles sont mémorisées, correspondent directement aux actions qui devraient être effectuées par un système centralisé (un programme de planification ou un individu) pour aboutir à un résultat. Le plan est alors obtenu comme un effet secondaire du comportement des agents.

Eco est un modèle général de système de résolution de problèmes fondé sur l'éco-résolution, qui se compose de deux parties :

Noyau : définit le protocole suivi par l'ensemble des agents. Ce noyau est complètement indépendant du domaine du problème à résoudre. Il consiste en une définition abstraite des agents du système appelés " éco-agents ", du protocole suivi par ces agents et des interactions de leurs différents comportements élémentaires.

Comportement spécifique : une partie dépendante du domaine d'application, où les comportements des agents spécifiques au domaine sont codés.

Un éco-agent est caractérisé par un but, un état interne (trois états sont possibles : satisfait, en recherche de satisfaction ou en recherche de fuite), des

actions élémentaires : liées au domaine, ce sont des actions de satisfaction et de fuite et des fonctions de perception des gêneurs (les agents qui empêchent l'agent courant d'être satisfait ou de fuir) et des dépendances (les agents dont l'agent courant est le but). Ces dépendances ne pourront être satisfaites que si l'agent courant se trouve lui-même satisfait.

Un éco-problème est défini par un ensemble d'agents dont chacun est caractérisé par un but et un comportement (automate et actions élémentaires dépendantes de l'application). On dispose d'une configuration initiale précisant l'état de départ des agents (recherche de satisfaction) et d'un critère de terminaison défini comme une fonction des états de satisfaction des éco-agents.

Nous pouvons citer comme exemples d'éco-problèmes : le problème du monde des cubes, le problème des tours de Hanoï, le problème du *Misachieving baby*, et le problème de Taquin pour lequel il a été montré [DD91] qu'une approche " auto-organisée " fournissait de meilleurs résultats que les approches heuristiques basées sur l'algorithme de recherche A^* , tout en autorisant la résolution d'instances de taille bien plus importante.

L'éco-résolution a été à l'origine de nombreux travaux de recherche en résolution distribuée de problèmes [GV91, Ghe99]. Elle a été appliquée aussi pour une problématique de simulation (modéliser des tas de granulaire sous forme multi-agent) et de résolution de problèmes (résoudre ces tas simulés en trouvant des lignes de force) [BZC99].

3.2.4 Formalisation des Systèmes Multi-Agent

Les concepteurs de SMA font usage d'un grand nombre de langages et de formalismes afin d'en cerner tous leurs aspects. Selon qu'on s'intéresse à l'implémentation ou à la formalisation des SMAs, à la représentation des connaissances des agents, à la définition de leur comportement ou à leurs communications. Les différents langages et formalismes intervenant dans la réalisation de SMA sont définis et classés dans [Fer95] selon cinq familles comme suit :

- Langages d'implémentation (L1)¹.

¹Un langage de programmation décrit les séquences d'instructions permettant aux agents d'exécuter leur comportement.

- langages de communication entre agents (L2)².
- Langages de description des comportements et des lois de l’environnement (L3).
- Langages de représentation de connaissances (L4)³.
- Langages de formalisation et de spécification (L5)⁴.

3.2.4.1 Langages de la famille : L1

Ces langages permettent :

- la programmation d’un SMA : recouvrir les structures informatiques utilisées pour les agents et l’environnement ainsi que les mécanismes informatiques permettant le parallélisme inter et intra-agents.
- l’implémentation effective des comportements, des activités d’envois et de réception de messages, de la perception d’objets et de l’ensemble des outils nécessaires pour mettre au point un SMA.

Ces langages sont par exemple : LiSP, C/C++, Prolog, SmallTalk, ou JAVA, et sont généralement les langages supportant des mécanismes d’exécution parallèle tels que les langages acteurs.

3.2.4.2 langages de la famille : L2

Ces langages assurent les interactions entre agents via la transmission d’informations et de demandes mutuelles de renseignements et de services, et permettent à des agents éventuellement hétérogènes de coordonner leurs actions, de coopérer pour atteindre un but commun et de s’intégrer à un groupe d’agents.

3.2.4.3 Langages de la famille : L3

Ces langages décrivent les comportements des agents et les lois de l’univers de l’environnement. Ils utilisent :

²Un langage de communication désigne des actes de langages.

³Un langage de représentation se réfère aux choses que l’agent peut connaître.

⁴Un langage de spécification représente les conditions que doivent respecter les entités présentes dans l’univers multi-agent

- Des formalismes adaptés pour la description de ces comportements ou ces lois à un niveau plus abstrait de manière à bénéficier de représentations indépendantes de l’implémentation et de la nature des agents.
- Des formalismes à niveau opératoire c’est à dire décrivant ce qui se passe de manière abstraite (exp. : langages à base de règles de production).
- Des formalismes issus de l’automatisme ou des systèmes distribués tels que les RdPs (le modèle BRIC).

3.2.4.4 Langages de la famille : L4

Ces langages sont utilisés par les agents cognitifs pour décrire des modèles internes du monde dans lequel ils évoluent et leur permettre de raisonner et de faire des prévisions sur l’avenir à partir des informations qu’ils disposent. Associés aux systèmes formels, ils permettent d’exprimer les états mentaux des agents et la représentation du contenu de leurs buts, leurs croyances et leurs comportements (exp. : langages d’IA comme les langages à base de règles, les tableaux noirs, et les langages de représentation structurée des connaissances tels que les réseaux sémantiques ou les Frames).

3.2.4.5 Langages de la famille : L5

Ce sont des langages situés au niveau le plus abstrait, ils permettent de formaliser les SMAs et les concepts d’interaction et d’intention, et de spécifier les conditions qui doivent être respectées lors de la modélisation et de l’implémentation de tels systèmes (exp. Langages issus des mathématiques).

Un protocole de négociation utilisera un langage de la famille L2, c’est à dire un langage de communication comme support de conversation, et un langage de la famille L3 pour décrire les mécanismes d’interprétation des messages et les fonctions d’évaluation des propositions évoquées de part et d’autre, le second interprétant les énoncés du premier. Les composants de la cognition ainsi que le mode de raisonnement des agents pourront être représentés dans des langages de la famille L1, l’interprète de ce langage étant généralement écrit en un langage de la famille L3.

Généralement, on peut se demander pourquoi les systèmes multi-agent au-

raient besoin de modèles ou de formalismes ? Ne se suffisent ils pas à eux mêmes. Le code des programmes permettant aux agents de s'exécuter n'est il pas suffisant à la compréhension de leur fonctionnement ? Tout n'est pas si simple. Pour un système multi-agent comme tout système d'information un tant soit peu complexe, analyser le comportement d'un programme à partir de son implémentation présente plusieurs inconvénients :

- Difficultés d'entrer dans le cœur d'un programme et d'analyser un comportement à travers l'implémentation.
- Un grand nombre de détails dans le code inutiles pour la compréhension du fonctionnement.
- Complexité de la compréhension du fonctionnement à partir du code d'agents en parallèle.
- Difficulté d'appréhender et de concevoir les systèmes multi-agent qui sont considérés des logiciels complexes.

Selon Ferber, une étape de formalisation s'avère pertinente pour représenter :

- Les actions des agents et de leurs conséquences dans l'environnement qui peut être complexe et d'une évolution autonome.
- Le fonctionnement d'un agent dans le comportement observable et dans la mécanique interne de l'agent.
- Les interaction des agents entre eux et en particulier les différents modes de communication, préluces aux formes organisées : coopération, coordination et résolution de conflits
- L'évolution du système multi-agent.

Peu de formalismes efficaces ont été définis : les automates à états finis ainsi que leur homologues à registres qui sont efficaces pour représenter des processus séquentiels, mais ils s'avèrent rapidement inefficaces lorsqu'il s'agit de prendre en compte les aspects du parallélisme, caractéristiques des SMAs, et les modèles algébriques d'équations différentielles très adaptés à une analyse formelle, mais inefficaces lorsqu'il faut représenter des entités autonomes en interactions ou des comportements qualitatifs. Alors, le besoin de disposer de formalismes plus élaborés devient primordial. Dans ce contexte, on peut citer que les réseaux de Petri nous semblent être un bon compromis entre les différents modèles cités. De nombreux travaux ont été réalisés sur la formalisation des SMAs par les réseaux de

Petri [SH94]. En effet, les réseaux de Petri peuvent être considérés comme des outils de modélisation à la fois graphiques et mathématiques. L’aspect graphique permet de visualiser les processus et de faciliter la conception, alors que l’aspect mathématique assure à ce formalisme des fondements rigoureux. L’utilisation des RdPs et particulièrement les RdPs colorés (Réseaux de Petri de haut niveau) pour modéliser les communications simultanées des agents avec plusieurs correspondants pour traiter des problèmes réels et des processus de décision fait l’objet de plusieurs travaux tels que [SHM98] et [BKPB00]. En outre, ce formalisme a fait preuve d’une puissance pour modéliser les aspects asynchronisation (messages) et parallélisme caractérisant le comportement des agents dans un système multi-agent.

Dans notre contexte, nous avons opté pour les RdPs colorés pour avoir un formalisme bien défini qui permet de donner une sémantique plus précise au comportement des agents de nos modèles ContEco et COSAH.

Dans ce qui suit, nous présentons les réseaux de Petri, étendus aux réseaux colorés, et décrivons un outil logiciel que nous avons utilisé pour simuler des RdPs colorés : CPN-tools.

3.3 Les réseaux de Petri

3.3.1 Définition d’un réseau de Petri

Concrètement un réseau de Petri est un graphe biparti comprenant deux sortes de noeuds : les places et les transitions. Les arcs de ce graphe relient les transitions aux places ou les places aux transitions. Les places contiennent des jetons ou marques qui se déplacent de place en place en franchissant les transitions suivant les règles de franchissement. On dit qu’une transition T est franchie, si toutes les places P d’entrée pour les quelles il existe un arc orienté de P vers T , possèdent une marque et parmi toutes ces places validées, une au hasard sera effectivement déclenchée, franchir une transition revient donc à enlever une marque sur chaque place d’entrée et à en ajouter une sur chaque place de sortie (voir figure 3.2).

De manière formelle, un RdP est alors un 5-uplet $\langle P, T, Pre, Post, M_0 \rangle$ [Val] : P est un ensemble fini de places, T est un ensemble fini de transitions,

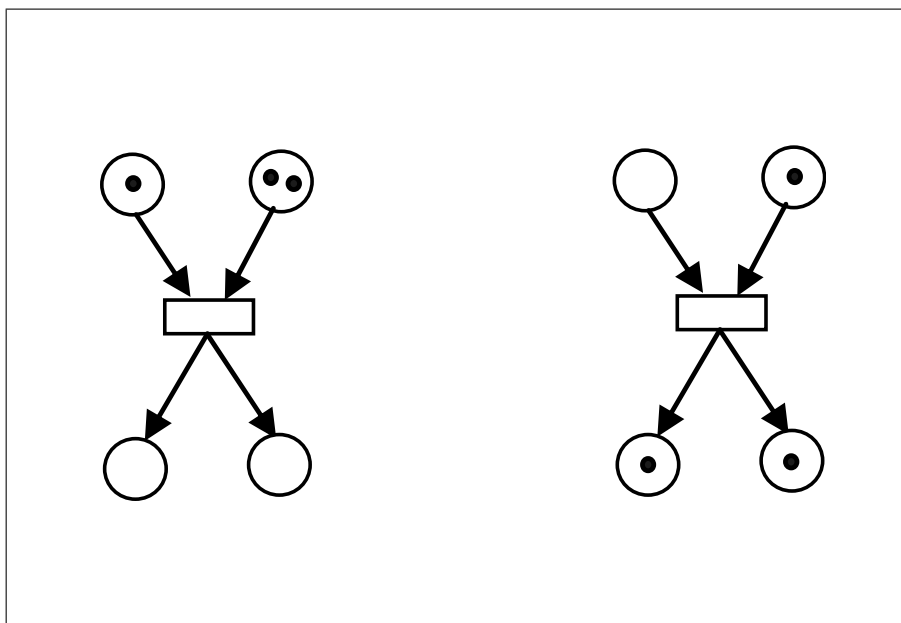


FIG. 3.2 – Tir d'une transition sensibilisée

$Pre : P * T \rightarrow N$: application incidence avant, $Post : P * T \rightarrow N$: application incidence arrière et $M_0 : P \rightarrow N$: application marquage initial (marques ou jetons).

Sensibilisation d'une transition : Une transition est sensibilisée par un marquage M si et seulement si :

$$\forall P \in P, M(P) \geq Pre(p, t)$$

Tir d'une transition : une transition sensibilisée \rightarrow tirée \rightarrow changement de marquage selon :

$$\forall P \in P, M'(P) = M(P) + Post(p, t) - Pre(p, t)$$

3.3.2 Réseaux de Petri colorés et la plate-forme CPN-Tools

Lorsque le nombre d'entités du système à modéliser est important, la taille du réseau de Petri devient rapidement énorme ; et si les entités présentent des comportements similaires. L'utilisation des réseaux de pétri colorés qui permettent en fait de condenser le modèle initial. Les réseaux de Petri colorés sont des réseaux

de Petri dans les quels les jetons portent des couleurs, sachant qu'une couleur est une information supplémentaire rattachée à un jeton et qui permet de distinguer les jetons entre eux et peut être de type quelconque [Jen97]. Ainsi les arcs ne sont pas seulement étiquetés par le nombre de jetons mais aussi par leurs couleurs. Le franchissement d'une transition est alors conditionné par la présence dans les places en entrée du nombre de jetons nécessaires, qui en plus satisfont les couleurs qui étiquettent les arcs.

Après le franchissement d'une transition, les jetons qui étiquettent les arcs d'entrée sont retirés des places en entrée tandis que ceux qui étiquettent les arcs de sortie sont ajoutés aux places en sortie de cette transition. Les réseaux colorés n'apportent pas de puissance de description supplémentaire par rapport aux réseaux de Petri, ils permettent juste une condensation de l'information. A tout réseau de Petri coloré marqué correspond un réseau de Petri qui lui est isomorphe.

Pour implementer des modèles RdP colorés, nous retrouvons l'environnement logiciel CPN-Tools [CPN] qui est un outil de d'édition, simulation et analyse d'espaces d'états. L'utilisation de la coloration permet d'enrichir les réseaux de Petri simples, en donnant plus de sens aux jetons. CPN-Tools permet également de créer des modèles hiérarchisés. Ce qui signifie ici que certaines transitions peuvent masquer un réseau de Petri.

Ayant défini les SMA comme étant une méthode que nous adoptons pour modéliser, simuler et résoudre le problème de stockage de conteneurs, et les réseaux de Petri colorés comme étant une méthode formelle pour décrire la dynamique globale de nos modèles ContEco et COSAH; nous allons définir dans la section suivante et dernière de ce chapitre le deuxième concept clé de notre démarche de résolution : l'optimisation heuristique que nous adoptons pour optimiser le problème sous-jacent.

3.4 Optimisation heuristique

3.4.1 Que faire face à un problème NP-dur ?

Jusqu'à maintenant, de nombreux problèmes ne peuvent pas être résolus de manière exacte en un temps de calcul raisonnable. Il en va ainsi de divers pro-

blèmes connus dans le milieu d'optimisation combinatoire comme étant très difficiles à savoir les problèmes d'ordonnancement, les problèmes de planification, les problèmes de tournées, le problème de sac à dos, le problème de coloriage de graphes, etc. Ayant à résoudre un problème d'optimisation combinatoire du type " trouver une solution qui optimise une fonction objectif sous un nombre de contraintes définies " et qui est prouvé (ou bien le problème de décision associé) NP-dur il est préférable d'abandonner l'idée d'avoir un algorithme polynômial qui donne à coup sûr la solution optimale, à moins de prétendre prouver que $P = NP$. Autrement dit, quoique sûres, ces algorithmes dits exacts soient à éviter vu leur complexité temporelle exponentielle. Deux attitudes peuvent avoir lieu : abandonner l'idée d'avoir un algorithme polynômial ou abandonner l'idée d'avoir un algorithme qui donne toujours la solution optimale. Dans le premier cas, nous pouvons essayer d'obtenir un algorithme exact non polynômial et ne l'appliquer que sur des données relativement petites. Pour certains problèmes, comme le problème du sac à dos, on a des algorithmes pseudo-polynômiaux qui peuvent rester raisonnablement efficaces sous certaines conditions. Nous pouvons aussi avoir des algorithmes qui donnent de bons résultats sur les données courantes qui ont une structure particulière. Nous pourrions sinon essayer des méthodes de type séparation, évaluation *Branch and Bound* qui évitent de parcourir l'espace de toutes les solutions en élaguant : par exemple, nous pourrions éviter de développer une solution partielle si nous sommes sûrs qu'elle sera moins bonne que la meilleure solution déjà trouvée. Dans le deuxième cas, nous abandonnons donc la contrainte " trouver la solution optimale " et chercher un algorithme polynômial qui trouve une solution, quelque fois optimale souvent bonne. Dans ce cas, nous utilisons donc un algorithme approché : il fournit toujours une solution mais pas forcément une solution optimale en un temps raisonnable. Un algorithme approché peut être basé sur un critère glouton propre au problème : on parle d'heuristique gloutonne *Greedy heuristic*, ou l'instantiation d'une métaheuristique (exp. algorithmes génétiques, recuit simulé, recherche tabou). Un algorithme approché peut être déterministe, c'est à dire, pour une entrée donnée, il donnera toujours la même solution (exp. heuristiques gloutonnes, recherche locale, recherche tabou) ou non déterministe (exp. recuit simulé, algorithme génétique).

Un algorithme approché (algorithme de recherche informée) met en oeuvre

des heuristiques. Une heuristique (du grec *heuriskein* : trouver, découvrir) peut avoir deux sens [PEA90] :

- moyen (pas sûr) de choisir entre plusieurs alternatives,
- sous certaines conditions, on peut être sûr de l’alternative choisie.

Ce sont des méthodes de résolution approchée, souvent plus simples et faciles à mettre en oeuvre que les méthodes exactes et sont nécessaires pour la résolution de problèmes NP-difficiles de grande taille ou comportant un grand nombre de contraintes. Les heuristiques sont basées sur des règles empiriques tirées de l’expérience. De nombreux problèmes se résolvent grâce à des heuristiques : le problème des n -reines, le jeu de Taquin, le problème du voyageur de commerce, etc. Selon le type de problème, on privilégiera une heuristique ou un type d’heuristique par rapport aux autres.

Les heuristiques peuvent se classer en trois catégories :

- Les heuristiques constructives gloutonnes se contentent de construire pas à pas une seule solution. Elles se caractérisent par une grande rapidité mais leur performance est souvent décevante.
- Les heuristiques de recherche locale (ou de recherche d’un optimum local) travaillent quant à elles sur une solution qu’elles tentent d’améliorer itérativement. Lors d’une itération, la solution courante est légèrement modifiée afin d’obtenir une solution voisine. Ces algorithmes obtiennent en général des résultats meilleurs que les procédures constructives, mais n’ont pas toujours une grande capacité à explorer des régions très différentes de l’espace des solutions.
- Les heuristiques évolutionnaires agissent sur une population d’individus (des solutions ou des morceaux de solutions) qui coopèrent et s’adaptent individuellement. Elles ont la plupart du temps un fort potentiel pour trouver des solutions très différentes lors de leur application, mais manquent souvent d’agressivité, car malgré la diversité des solutions rencontrées, celles-ci ne sont pas toujours de grande qualité. De plus, elles nécessitent souvent un ajustement long et fastidieux de nombreux paramètres avant d’obtenir des résultats intéressants.

Dans ce qui suit, nous allons décrire ces heuristiques selon deux classes : les heuristiques constructives, et les métaheuristiques incluant les heuristiques de re-

cherche locale et les heuristiques évolutionnaires. C'est la classification qui est souvent adoptée dans la plupart des travaux de description des méthodes approchées.

3.4.2 Classes d'heuristiques

3.4.2.1 Heuristiques constructives gloutonnes *Greedy Heuristics*

Le fonctionnement d'une heuristique constructive gloutonne est similaire à celui d'un algorithme glouton [Edm71] exact. La différence réside dans le fait qu'on n'impose plus que la solution obtenue soit optimale. Il existe de nombreuses heuristiques gloutonnes dédiées à des problèmes classiques : le problème du monde des cubes, le problème de coloriage de graphes, le problème d'affectation, le problème de sac à dos, etc., dont certaines donnent de bons résultats et sont très utilisées. Cette classe d'heuristiques construit une solution progressivement sans remettre en cause les choix effectués à chaque itération. Ces choix sont les plus intéressants relativement à un critère donné appelé critère glouton.

3.4.2.2 Les métaheuristiques

Les métaheuristiques sont des méthodes " génériques " pour rechercher une solution approchée aux problèmes d'optimisation. Elles sont souvent basées sur des analogies avec des phénomènes physiques (recuit simulé) ou naturels (algorithmes génétiques, colonies de fourmis, essaims, etc.). Nous pouvons les distinguer en deux classes de méthodes : les méthodes par voisinage, et les algorithmes génétiques :

Méthodes par voisinage Nous présentons brièvement trois méthodes basées sur la notion de voisinage. L'idée sous-jacente à ces trois méthodes est simple : on explore l'espace des solutions en partant d'une solution et en se " déplaçant " de proche en proche, les voisins d'une solution sont des solutions proches de celle-ci, c'est à dire qu'on peut obtenir par exemple en transformant à peine la solution initiale. Pour appliquer les méthodes ci-dessous, il faut donc avoir au préalable défini une notion de voisinage d'une solution. Bien sûr, pour le même problème, plusieurs notions de voisinage peuvent être définies.

Recherche locale ou Recherche d'un optimum local *Hill-Climbing*

On part d'une solution si possible " bonne " (par exemple donnée par une heuristique gloutonne) et on balaie l'ensemble des voisins de cette solution : s'il n'existe pas de voisin meilleur que notre solution, on a trouvé un optimum local et on arrête. Sinon, on choisit le meilleur des voisins et on recommence. Une autre implémentation consiste non pas à passer au meilleur des voisins à chaque étape mais au premier meilleur voisin trouvé. La convergence vers un optimum local pouvant être très lente, on peut éventuellement fixer un nombre de boucles maximum, si on veut limiter le temps d'exécution. Cette méthode a l'inconvénient de " rester bloquée " dans un optimum local : une fois un optimum local trouvé, on s'arrête, même si ce n'est pas l'optimum global. Selon le " paysage " des solutions, l'optimum local peut être très bon ou très mauvais par rapport à l'optimum global. Si la solution de départ est donnée par une heuristique déterministe, l'algorithme sera déterministe. Si elle est tirée au hasard, on a un algorithme non déterministe et donc plusieurs exécutions différentes sur la même instance pourront donner des solutions différentes et de qualités différentes. Si les voisins sont très nombreux, on a de fortes chances de trouver l'optimum global mais visiter un voisinage peut être long : on visitera une grande partie de l'espace des solutions. Si le voisinage est très restreint, on risque fort de rester bloqué dans un optimum local de " mauvaise qualité " : le choix de la notion de voisinage est un compromis entre efficacité et qualité. Les deux méthodes qui suivent peuvent être vues comme des méthodes de recherche locale avancées.

La recherche Tabou *Tabu Search* Le principe de la recherche tabou a été défini par Glover en 1986 [Glo86]. Cette méthode repose sur une absence du hasard et sur une liste de taille fixe qui mémorise les dernières solutions visitées. A chaque étape, on recherche le meilleur voisin non tabou (qui n'appartient pas à cette liste) : un voisin est à priori tabou si on a exploré cette solution durant les N précédentes itérations. La maintenance d'une liste " taboue " de voisins étant souvent trop coûteuse, on se borne souvent à stocker les mouvements (ou transformations) effectués. A chaque itération, on choisit le meilleur voisin (ou un meilleur voisin selon le cas) correspondant à un mouvement non tabou ; on effectue ce mouvement, on le place dans la liste (file) taboue et on élimine le

plus ancien mouvement de cette liste, si la file est pleine. De plus, si la solution actuelle est la meilleure trouvée depuis le début, on la stocke. On s'arrête soit après un nombre fixé d'itérations, soit après un nombre fixé d'étapes n'ayant pas amélioré la solution. Une fois la notion de voisin fixée, cette solution pourra donc être paramétrée par la longueur de la liste taboue. La méthode " taboue " peut donc être vue comme une généralisation de la recherche d'optimum local (si $N=0$, on retombe dans le cas du hill-climbing), l'idée est de permettre de s'échapper d'un optimum local grâce à la liste taboue.

Le recuit simulé *Simulated Annealing* Inventée par Kirkpatrick en 1983 [KGV83], cette méthode est inspirée de la physique : le recuit est utilisé pour obtenir des états de faible énergie il est utilisé par exemple dans la fabrication du verre. On part d'une solution quelconque (à priori, il n'est pas forcément intéressant de partir d'une " bonne " solution). A chaque étape, on tire au hasard une solution voisine. Si elle est meilleure on l'adopte, sinon on calcule l'augmentation de coût Δ et on va adopter cette solution avec une probabilité liée à cette augmentation : plus le coût augmente, plus la probabilité de la garder sera faible. Mais cette probabilité dépend aussi du temps écoulé depuis le lancement de l'algorithme : au début, on accepte facilement un changement qui produit une solution plus coûteuse ; mais à la fin on l'accepte très difficilement. Plus précisément on calcule $accept = E^{-\Delta/T}$, T étant une quantité que nous appellerons la température. On tire au hasard un réel p entre 0 et 1 : si p est inférieur à $accept$, on adopte la nouvelle solution, sinon on garde l'ancienne. Il faut donc choisir une température initiale, et une façon de diminuer la température au cours de l'exécution pour assurer l'arrêt. La température initiale doit être assez élevée pour que $accept$ soit assez grande au départ même si l'augmentation de coût est grande. C'est un paramètre fixé par l'utilisateur. Pour diminuer T , on peut par exemple multiplier T par un réel α inférieur à 1 à chaque fois. Plus ce réel est proche de 1 plus le refroidissement est lent. Ce coefficient de refroidissement est lui aussi un paramètre fixé lors de l'utilisation. On peut aussi raffiner : on effectue plusieurs cycles de recuit simulé, en repartant à chaque fois de la solution trouvée au cycle précédent. Une difficulté de cette heuristique réside dans le réglage du refroidissement ; celui-ci pourra être choisi différemment selon l'instance du problème considérée.

Les algorithmes génétiques *Genetic Algorithms* Les algorithmes génétiques, créés par J. Holland en 1975 [Hol75], sont basés sur deux principes de génétique : la survie des individus les mieux adaptés et la recombinaison génétique. Contrairement aux algorithmes précédents qui essayaient d'améliorer un unique individu-solution, les algorithmes génétiques font évoluer une population de solutions. Le principe de base est de simuler l'évolution d'une population de solutions avec les règles citées ci-dessus en vue d'obtenir une solution ou un ensemble de solutions les plus adaptées. Selon [Gol94], le principe est le suivant : A chaque génération, un nouvel ensemble de créatures artificielles est créé en utilisant des parties des meilleures solutions précédentes avec éventuellement des parties innovatrices.

L'arrêt se fera par exemple quand la population ne s'améliore plus de manière significative ou après un nombre donné de générations. Pour la mise en oeuvre, il faut donc :

- Choisir le codage (souvent binaire) d'une solution.
- Fixer la taille d'une population.
- Définir la fonction d'adaptation *fitness*.
- Définir les opérateurs de sélection, le principe étant : meilleure est notre adaptation, plus on a de chances d'être sélectionné.
- Définir les opérateurs de croisement.
- Définir éventuellement les opérateurs de mutation.

L'un des problèmes de la méthode est encore la convergence prématurée vers une population " de mauvaise qualité ".

D'autres méthodes basées sur des " populations " ont été mises au point, comme les algorithmes de colonies de fourmis.

L'utilisation et le choix d'une métaheuristique (gloutonne, de recherche locale ou évolutionnaire) doit se faire en tenant compte de l'importance du paramétrage ; de la qualité du voisinage de la solution initiale, de la taille de ce voisinage ainsi que de la vitesse de calcul de la fonction d'évaluation.

Dans notre cadre, nous avons choisi de travailler avec les heuristiques constructives gloutonnes, vu leur simplicité et facilité à mettre en oeuvre ainsi que leur rapidité de retrouver une solution réalisable. Ajoutons que notre but est de construire pas à pas une solution qui tient compte des contraintes spatio-temporelles

dynamiques et de l'environnement imprévisible caractérisant notre problème. D'où la nécessité de fixer des critères de choix qui seront injectés au niveau du comportement de chaque agent du modèle proposé COSAH chaque fois qu'il s'agit d'une décision à prendre.

3.5 Conclusion

Ce troisième chapitre a été l'occasion de justifier et présenter les méthodes et approches dont il sera fait usage dans la suite de cette thèse. Au fur et à mesure de cette présentation, nous avons précisé à quels aspects nous souhaitons nous attacher plus particulièrement et à quels concepts nous faisons appel dans les deux modèles que nous proposons à savoir ContEco et COSAH.

D'abord les systèmes multi-agent pour modéliser, simuler et résoudre le problème du stockage de conteneurs. Nous avons ainsi cherché à définir ce que nous entendons par un agent, un système multi-agent, la communication, la coopération, la négociation, l'écorésolution et la formalisation d'un système multi-agent.

Ensuite les réseaux de Petri pour formaliser le modèle multi-agent de base ContEco. Nous avons ainsi cherché à définir ce que c'est un réseau de Petri et un réseau de Petri coloré.

Et enfin, l'optimisation heuristique pour optimiser le problème étudié, autrement dit pour améliorer la qualité de la solution obtenue en procédant uniquement à une résolution distribuée basée sur les systèmes multi-agent. Nous avons ainsi cherché à définir une heuristique et les différentes classes d'heuristiques, en particulier les heuristiques constructives, gloutonnes adoptées dans le modèle COSAH

Une fois argumentés et définis, nous partons de ces propos pour passer au chapitre suivant où nous allons décrire de manière exhaustive nos deux contributions : le modèle de base ContEco et le modèle étendu COSAH. Nous détaillons l'architecture, la dynamique globale et la dynamique détaillée de chacun.

Chapitre 4

Le modèle COSAH : Un modèle d'Optimisation Heuristique Distribuée du PSC

4.1 Introduction

L'objectif que nous nous étions fixé et annoncé au début de cette thèse est la résolution et l'optimisation du Problème de Stockage de Conteneurs-PSC dans un port Fluvial ou maritime. Dans ce chapitre, nous présentons d'abord notre première contribution à travers laquelle nous avons cherché à exploiter la forte analogie entre une version relaxée du PSC Problème de Stockage de Conteneurs-PSCR *Relaxed Container Stacking Problem-RCSP* et le problème classique du monde des cubes, et ce en concevant un modèle multi-agent de base désigné par ContEco pour *Container stacking Eco-solving*. Ensuite, nous présentons notre deuxième contribution : un modèle multi-agent étendu reposant sur une méthode d'optimisation heuristique distribuée [KKGY07a] d'une version étendue du PSC. Ce modèle est appelé COSAH pour *COntainer Stacking based on multi-Agent approach and Heuristic method*, il sert les deux concepts : Agent et Heuristique, Agent pour la résolution, et Agent et Heuristique pour l'optimisation. COSAH est constitué d'un noyau et d'un module complémentaire, désignés respectivement par DistNIS pour *Distributed Non Informed Search* et ISM pour *Informed Search*

Module. DistNIS met en œuvre un algorithme distribué de recherche non informée simulant et résolvant l'activité de stockage de conteneurs tout en respectant ses contraintes spatio-temporelles. En y injectant le module ISM, DistNIS est transformé en un algorithme de recherche informée désigné par DistIS pour *Distributed Informed Search* qui utilise des heuristiques constructives gloutonnes. Du fait, le but essentiel de COSAH est de fournir une solution qui satisfait les contraintes spatiales et temporelles du problème et qui minimise le nombre de mouvements parasites. Ainsi, ce modèle pourra réduire ainsi la complexité spatio-temporelle en élaguant, de manière distribuée, l'espace de recherche inutile. Ces deux contributions, les versions du problème auxquelles elles sont dédiées ainsi que les relations entre elles, sont représentées par le schéma conceptuel de la figure 4.1.

4.2 Modèle de base : ContEco

Cette section est consacrée en un premier temps à la description détaillée du modèle de base ContEco [KKGY06a, KKGY06b], et en un deuxième temps à la formalisation de ce modèle en faisant appel à la méthode formelle de réseaux de

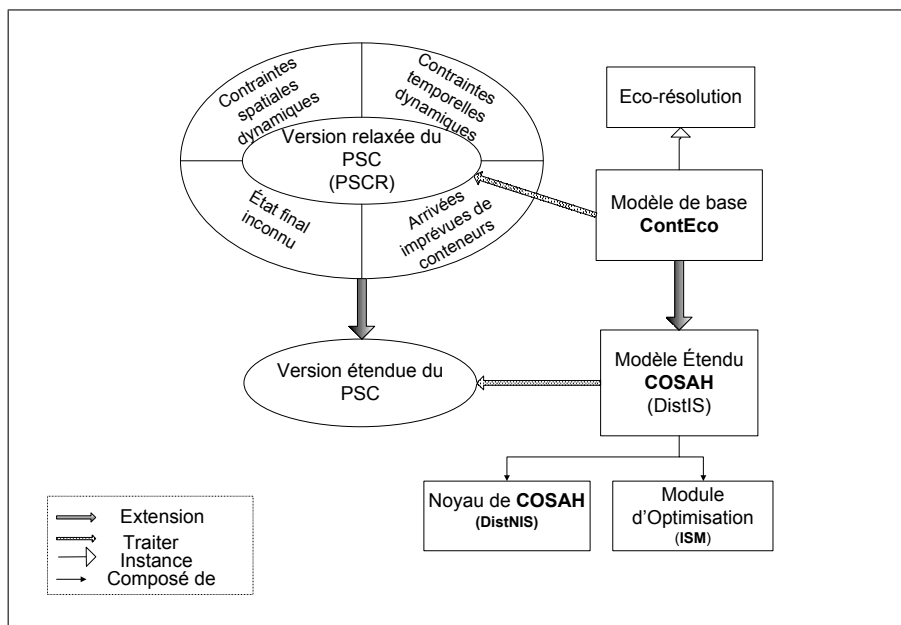


FIG. 4.1 – Schéma Conceptuel de l'approche proposée

Petri colorés.

4.2.1 Description de la version relaxée du PSC : PSCR

Nous traitons une version relaxée du problème de stockage de conteneurs appelée Problème de Stockage de Conteneurs-PSCR *Relaxed Container Stacking Problem-RCSP* qui simplifie la structure du problème originel. Nous nous restreignons à étudier le problème de stockage de conteneurs initialement rangés sur le quai d'un port de terminaux à conteneurs selon une configuration initiale, puis à la suite de départs de certains conteneurs, la question est comment retrouver une séquence faisable d'opérations d'empilements et de dépilements qui puisse nous mener vers une configuration finale bien déterminée connue à l'avance, tout en respectant les contraintes d'espace et de temps. Dans cette version relaxée, nous n'allons pas considérer l'arrivée de conteneurs. Dans la suite nous présentons formellement les différents éléments de cette version afin de la mieux appréhender.

4.2.1.1 Paramètres

Les paramètres de travail que nous nous sommes fixés sont les suivants :

- Une zone de stockage S composée de M emplacements.
- N Conteneurs ISO stockés en piles.
- H_{max} : la hauteur maximale des piles.
- $Config_{init}$: un arrangement initial quelconque des N conteneurs dans les M emplacements.
- $Config_{fin}$: un arrangement final (connu à l'avance) de N' conteneurs dans les M Emplacements devant être atteint à partir de $Config_{init}$.

4.2.1.2 Hypothèses

Nous adoptons les hypothèses suivantes :

- Chaque conteneur a un port de destination, une charge, une tare, une dimension (1 EVP, 2 EVP etc.), une date de départ souhaitée connue à l'avance aussi.
- Les conteneurs de dimensions différentes sont stockés sur deux zones de stockage différentes.

- Une arrivée et un départ ne peuvent pas coïncider.
- Nous supposons qu'il n'y a pas d'arrivées prévues de conteneurs.
- Un mouvement consiste à saisir un conteneur, le déplacer et le déposer dans un autre emplacement.
- Les navires dont la date de départ est connue d'avance sont appelés A, B, C, D, E, F selon l'ordre chronologique de leurs départs.
- Les navires dont le départ est certain mais dont la date associée n'est pas connue avec précision, sont appelés Y. Nous sommes sûres uniquement que cette date est postérieure à la dernière date considérée (la date de la barge F).
- Les navires dont la date de départ est inconnue sont appelées : Z.
- Nous associons à chaque conteneur la barge par laquelle il va partir (A, B, C, D, E, F, et Z) ce qui détermine implicitement sa date de départ t . Par exemple pour deux conteneurs différents C et C', $t(C) < t(C')$ si C va partir sur le navire A et C' va partir sur le navire B, C, D, ...ou Y.
- Les conteneurs sont catégorisés par référence aux navires sur lesquelles ils vont être chargés. Par exemple un conteneur est de catégorie A s'il va être chargé sur le navire A
- Les conteneurs sont prioritaires les uns aux autres selon leur catégorie (les conteneurs de catégorie A sont les plus prioritaires).

4.2.1.3 Objectifs

Nous cherchons à établir une séquence faisable de mouvements (dépilement/empilement) respectant les contraintes spatio-temporelles. Cette séquence permettra d'atteindre un arrangement final connu à l'avance à partir d'un arrangement initial donné (cf. Figure 4.2) suite au départ des conteneurs prioritaires A, et au réarrangement des conteneurs déplacés.

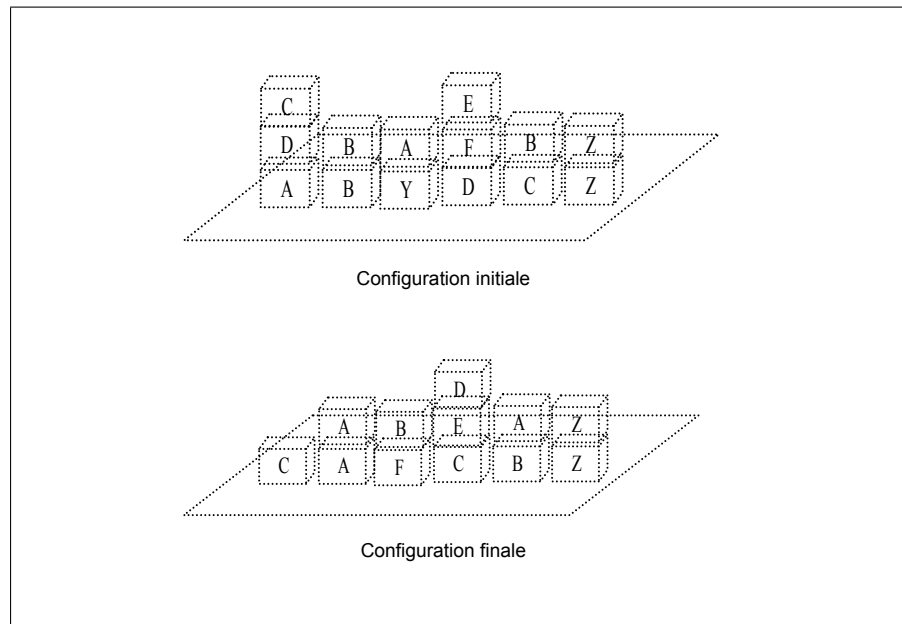


FIG. 4.2 – Configuration initiale et configuration finale à atteindre

4.2.2 Description de ContEco

Comme nous le constatons, le PSC en sa version relaxée décrite ci-dessus peut être considéré comme étant un éco-problème à traiter. Dès lors, nous avons développé le modèle multi-agent ContEco qui simule et résout RCSP en s'inspirant du principe de l'éco-résolution appliquée au problème du monde des cubes et pas mal d'autres problèmes [Dro00]. Dans les sections suivantes, nous détaillons ContEco en présentant les définitions proposées, son architecture et sa dynamique globale.

4.2.2.1 Définitions

- Un agent Conteneur AC_i est un agent associé à un conteneur i .
- L'agent Conteneur AC_0 est un agent Conteneur " virtuel " associé au sol.
- Un AC ne peut être placé sur un autre que s'il est devenu libre.
- Chaque AC_i possède un AC_But selon la configuration finale $Config_{fin}$.
- Un AC est satisfait s'il est placé sur un AC_But .
- L'état d'un AC est soit satisfait soit insatisfait.
- Un AC est libre s'il n'est pas recouvert par des ACs (placés au-dessus).

4.2.2.2 Architecture

Les agents mis en jeu dans notre modèle multi-agent sont l'agent Conteneur et l'agent Interface. L'agent Conteneur est défini par ses accointances formées par les autres agents Conteneur et l'agent Interface, ses connaissances statiques : identificateur, port_Destination, charge, tare, AC_But et dimension, et ses connaissances dynamiques : catégorie, $AC_dessous$, AC_dessus et son état. Son comportement sera défini dans la section suivante. Un agent Conteneur AC peut être considéré comme un ensemble de composants qui lui permettent de percevoir, décider et agir. Comme le montre la figure 4.3 un agent Conteneur contient essentiellement un module de communication qui lui offre la possibilité d'échanger des messages avec ses accointances et un module de décision qui dépend du comportement interne de cet agent.

4.2.2.3 Dynamique globale

Partant d'un arrangement initial, notre modèle doit permettre d'abord le départ des conteneurs les plus prioritaires A, ensuite le réarrangement des autres

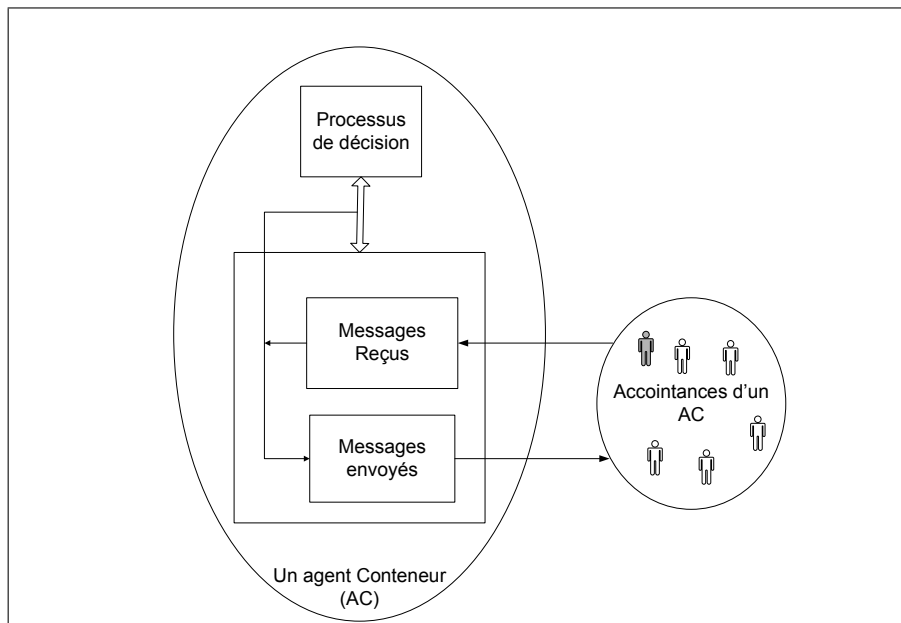


FIG. 4.3 – Composants d'un agent Conteneur et ses interactions

conteneurs selon un arrangement final bien déterminé connu à l'avance, et enfin la mise à jour ce dernier (décaler les catégories des conteneurs). Afin d'atteindre cet objectif global, les agents Conteneur coopèrent et négocient en menant une conversation particulière pour que chacun atteigne son objectif local individuel celui d'être placé sur un de ses agents Conteneur but. Cette conversation obéit au modèle d'éco-résolution. Elle est assurée par l'échange direct des messages suivants :

- " Tu_me_Gènes " (G) : AC_i veut se libérer d'un AC_j au dessus.
- " Chercher_Place " (CP) : AC_i est en état de fuite, il recherche un emplacement vide approprié.
- " Je_suis_Libre " (LI) : un AC_i informe un AC_j qu'il est devenu libre.
- " Tu_es_Libre " (L) : un AC_i indique à un AC_j qu'il n'est plus au dessus, donc AC_j est devenu libre.
- " Sur_Toi " (ST) : un AC_i informe un AC_j qu'il est placé au-dessus.

L'élaboration du modèle de base ContEco nous a permis d'agencer notre problème par rapport aux travaux existants à propos de la résolution distribuée de problèmes NP-difficiles, particulièrement via l'éco-résolution. Une étude expérimentale préliminaire [KKGY06a] nous a permis aussi de tester la faisabilité de nos réflexions et d'appuyer nos propositions.

Afin de décortiquer et simuler la dynamique globale multi-agent du modèle ContEco, nous avons utilisé le formalisme réseaux de Petri colorés explicitant la conversation entre deux agents distincts. Les modèles RdPs associés sont simulés par l'outil CPN-Tools.

4.2.2.4 Modèles RdPs

Nous ne pouvons pas nous limiter au formalisme des réseaux de Petri classiques, car nos agents Conteneur vont prendre leur décision en fonction de critères bien définis. Il va falloir donner un peu plus de sens à nos jetons, en distinguant des jetons " classiques " caractéristiques d'un état binaire, et des jetons qui peuvent contenir des informations, comme par exemple le type du message échangé et l'état de l'agent Conteneur qui varient lors des communications entre les différents agents Conteneur.

Le modèle RdP coloré principal associé au problème sous-jacent est fait de

modules correspondant au comportement interne d'un agent Conteneur. Ce modèle est constitué de deux modules de base : module Autonomie (figure 4.4) et module Réception (figure 4.5). Les deux sections suivantes vont détailler ces deux modules.

Coloration des jetons Nous avons défini des jetons de couleurs différentes qui contiennent diverses informations. La couleur agent Conteneur, notée AC contient les informations relatives à un agent Conteneur : son identificateur id_{cc} , l'identificateur de l'agent Conteneur au-dessous id_{c-1} , l'identificateur de l'agent Conteneur en-dessus id_{c+1} , l'identificateur de son agent Conteneur but id_{cb} et son état $etat_{cc}$ (satisfait/ non satisfait). Cette couleur est définie par un tuple : $(id_{cc}, id_{c-1}, id_{c+1}, id_{cb}, etat_{cc})$ comme par exemple $(1, 10, 2, 0, "sat")$.

La couleur Message, notée MES , contient les informations relatives à un message particulier échangé entre les agents Conteneur : son expéditeur, son destinataire et son type (" G " / " CP " / " ST " / " LI " / " L "). Cette couleur se définit par un triplet de couleurs $(Dest, Exped, Type)$ exp. $(i, j, "G")$.

Ces jetons pourront être mis à jour au cours de la communication des agents Conteneur. La couleur simple binaire, que l'on retrouve dans les réseaux de Petri ordinaires est notée e .

Module d'Autonomie Ce module décrit le comportement cognitif d'un agent Conteneur. Tant qu'il ne reçoit pas de messages, il essaie de satisfaire son objectif local. Dans le cas où il est satisfait, il reste en attente d'un message. S'il ne l'est pas, il essaie de fuir et de chercher un autre emplacement, deux sous cas se présentent : s'il est libre alors il envoie aux autres agents Conteneur un message " CP " et reste en attente d'un message " LI ", une fois qu'il l'a reçu, il met à jour son AC-dessous et envoie le message " L " à son ancien AC_dessous et un message " ST " à son nouveau AC_dessous. Dans le cas où il est gêné par un agent Conteneur dessus, il envoie à ce dernier un message " G ". Et on relance le même processus (figure 4.4).

Module de Réception Ce module décrit le comportement réactif d'un agent Conteneur suite à la réception de chaque type de messages : " G ", " CP ", " ST "

", " L ", ou " LI " (figure 4.5).

- A la réception d'un message " G " (figure 4.6) : deux cas se présentent : si l'agent Conteneur générateur est libre alors il part et recherche un autre emplacement jusqu'à la réception d'un message " LI ". S'il n'est pas libre, il

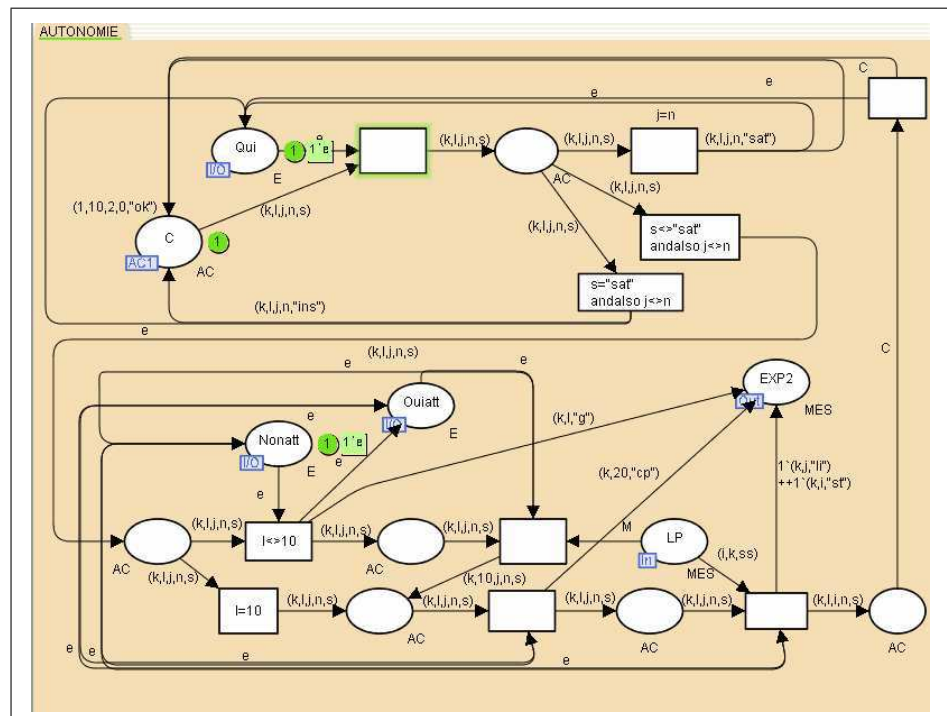


FIG. 4.4 – Module d'autonomie d'un agent Conteneur

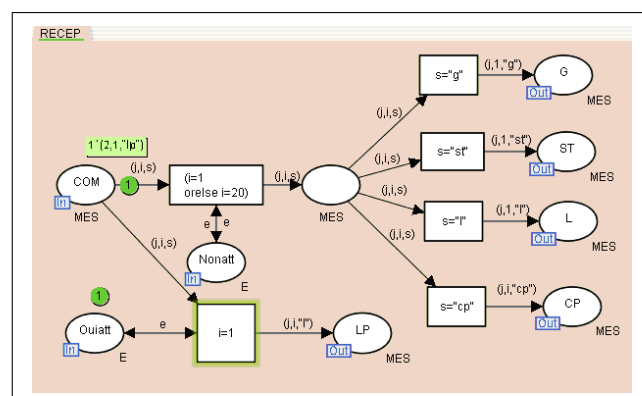


FIG. 4.5 – Module de réception d'un agent Conteneur

envoie un message " G " à son AC_dessus jusqu'à la réception d'un message " L " .

- A la réception d'un message " CP " (figure 4.7), un agent Conteneur vérifie s'il est libre, dans ce cas il renvoie un message " LI " . Dans le cas contraire, il ignore ce message.
- A la réception d'un message " ST " (figure 4.8), un agent Conteneur met à jour son AC_dessus.

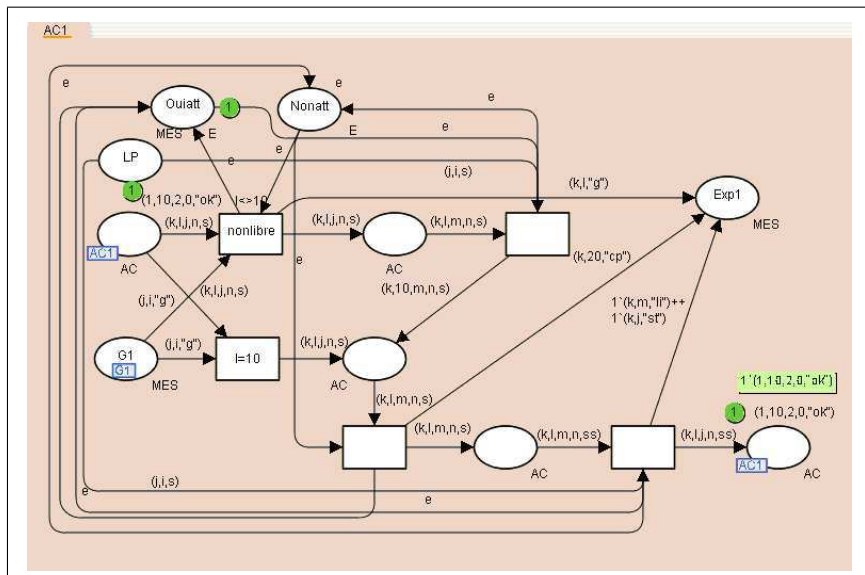


FIG. 4.6 – Message " Tu_me_gênes (G) "

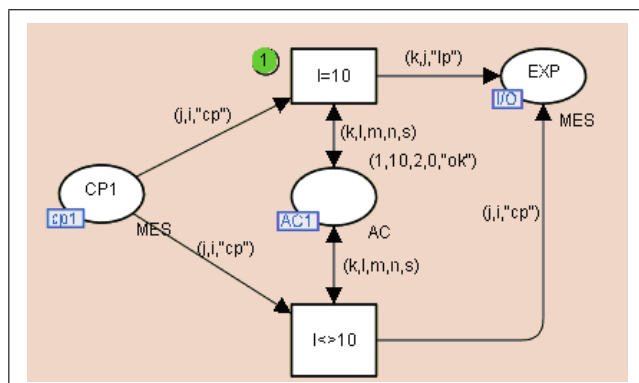


FIG. 4.7 – Message " Chercher_Place (CP) "

- A la réception d'un message " L " (figure 4.9), un agent Conteneur met à jour son AC_dessus à " libre ".

4.3 Le modèle COSAH

Dans cette section, nous procédons à une étape préalable de présentation, qui permet de ressortir les caractéristiques essentielles du PSC (en une version étendue par rapport à PSCR) que nous tenons en compte et de poser les définitions et les notations adoptées, suit une description exhaustive du modèle COSAH. Cette description se fait en deux temps : dans un premier temps, nous décortiquons le noyau de COSAH : DistNIS, et en deuxième temps, nous détaillons le module complémentaire ISM dévoilant la méthode heuristique que nous avons adoptée

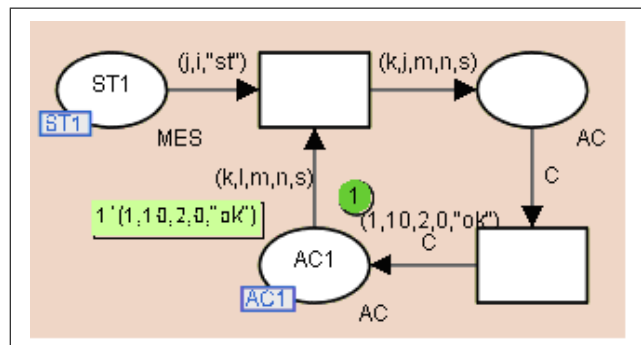


FIG. 4.8 – Message " Sur_Toï (ST) "

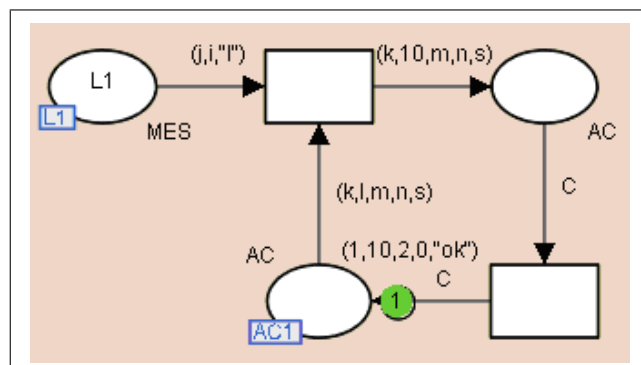


FIG. 4.9 – Message " Tu_es_Libre (L) "

pour l'optimisation du PSC.

4.3.1 Description du problème étudié

Comme nous l'avons déjà invoqué, notre souci est d'arranger les conteneurs en import ou en export dans des emplacements adéquats tenant compte des contraintes spatio-temporelles et minimisant le nombre de mouvements improductifs lors de départs de quelques uns pour être chargés dans un navire, camion ou train. Par exemple, un camion arrive de manière imprévue et demande un conteneur, un des problèmes difficiles est comment retrouver facilement ce conteneur ? La situation idéale est qu'il soit placé au sommet d'une pile dans la zone stockage, ce qui est très peu évident. Notre apport dans ce travail est la manière de considérer le problème étudié pour pouvoir le résoudre tenant compte de ses aspects dynamique et complexité. Nous allons chercher à répondre la question suivante : après chaque départ de quelques conteneurs, comment pouvons nous réarranger les conteneurs perturbés par ces départs de la manière la plus adéquate ? Notre idée de base est de comment gérer efficacement les prochains départs (atteindre facilement les conteneurs pour être chargés sur navires ou camions) aussi bien que les prochaines arrivées (déterminer des emplacements appropriés pour les conteneurs de manière à faciliter leur départs ultérieurement).

4.3.1.1 Contraintes

Dans cette section, nous proposons une formulation mathématique des contraintes statiques du problème étudié afin de lui donner un aspect formel.

$$\sum_{p=0}^{np-1} X_p \leq NP \quad (4.1)$$

$$H_p \leq H_{max}, \forall p \in \{0, \dots, NP\} \quad (4.2)$$

$$Y_{ij}DT_i \leq Y_{ij}DT_j, \forall i, j \in 1, \dots, N, i \neq j \quad (4.3)$$

$$X_{ij}Y_i > 0, \forall i, j \in 1, \dots, N, i \neq j \quad (4.4)$$

où :

$$\begin{aligned}
 X_p &= \begin{cases} 1, & \text{Si la pile } p \text{ contient au moins un conteneur;} \\ 0, & \text{Sinon.} \end{cases} \\
 X_{ij} &= \begin{cases} 1, & \text{Si le conteneur } j \text{ peut se placer sur le conteneur } i; \\ 0, & \text{Sinon.} \end{cases} \\
 Y_i &= \begin{cases} 1, & \text{Si le conteneur } i \text{ a au moins une place au dessus;} \\ 0, & \text{Sinon.} \end{cases} \\
 Y_{ij} &= \begin{cases} 1, & \text{Si le conteneur } i \text{ est de la catégorie inférieure à celle de } j; \\ -1, & \text{Sinon.} \end{cases}
 \end{aligned}$$

- La contrainte 4.1 assure que le nombre de piles exploitées ne doit pas dépasser le nombre de piles disponibles.
- La contrainte 4.2 exprime qu'on ne doit pas dépasser la hauteur maximale des piles.
- La contrainte 4.4 assure qu'un conteneur ayant une catégorie inférieure à celle d'un autre, doit partir avant.
- La dernière contrainte 4.3 exprime que chaque conteneur ne peut être placé sur un autre que s'il y a au moins une place vide immédiatement dessus.

4.3.1.2 Hypothèses

Nous considérons les hypothèses suivantes :

- Une zone de stockage contenant éventuellement N conteneurs placés dans M emplacements disponibles, et structurés en piles distinctes ayant une hauteur maximale.
- Chaque conteneur possède : un port de provenance, un port de destination, une charge, une tare, une dimension (1 EVP, 2 EVP, etc.), une date d'arrivée inconnue à l'avance et une date de départ souhaitée connue à l'avance.
- Les conteneurs considérés sont en import ou en export.
- Une arrivée et un départ ne peuvent pas coïncider.
- Les conteneurs de dimensions différentes sont stockés dans deux zones de stockage différentes.
- Les distances entre les différentes piles de la zone de stockage sont prises en

compte, tandis que la distance de la zone de stockage vers le quai ne l'est pas.

- Le temps mis par un opérateur pour déplacer un conteneur d'un emplacement à un autre est identique quelque soit l'emplacement courant de ce conteneur.
- Un mouvement consiste à saisir un conteneur, le déplacer et le déposer dans un autre emplacement.
- Les navires, camions ou trains dont la date départ est connue d'avance sont appelées A, B, C, D, E et F selon l'ordre chronologique de leurs départs.
- Les navires, camions ou trains dont la date de départ n'est pas connue avec précision, sont appelées Y. Nous sommes sûrs uniquement que cette date est postérieure à la dernière date considérée (la date de départ F).
- Les navires, camions ou trains dont la date de départ est inconnue sont appelées : Z.
- Nous associons à chaque conteneur en export (en import) le navire (camion ou train) par lequel il va partir (A, B, C, D, E, F, ..., Y & Z), ce qui correspond implicitement sa date de départ dénotée t . Par exemple pour deux conteneurs différents C et C', $t(C) < t(C')$ si C va partir sur la barge A et C' va partir sur la barge B, C, D, ..., ou Y etc.
- Les conteneurs en export sont catégorisés par référence aux navires (camions ou trains) par lesquelles ils vont partir. Par exemple un conteneur est de catégorie A s'il va partir sur le navire A.
- La priorité entre les conteneurs est définie selon leurs catégories.

4.3.2 Description du modèle COSAH

4.3.2.1 DistNIS

Afin de simuler et résoudre le problème étudié, nous proposons un modèle multi-agent, désigné par DistNIS qui constitue le noyau du modèle proposé COSAH, et qui a pour objectif la satisfaction des contraintes de hauteur maximale, du nombre des piles et de date de départ des conteneurs. Il s'agit d'assurer le départ des conteneurs prêts à être chargés, de déterminer aléatoirement des emplacements pour les conteneurs perturbés qui vont partir ultérieurement ainsi que

pour les conteneurs déchargés entre deux départs. DistNIS ne cherche pas à optimiser le nombre de mouvements parasites pouvant avoir lieu au cours des activités chargement et déchargement, nous nous restreignons à respecter uniquement les contraintes spatio-temporelles. Pour ce, DistNIS se base sur un algorithme distribué de recherche non informée, précisément sur un algorithme distribué de stockage aléatoire.

Notations

- P_{max} : le nombre maximum de piles dans la zone de stockage.
- H_{max} : la hauteur maximale des NP piles.
- N : le nombre total d'agents Conteneur.
- H_p : la hauteur d'une pile p courante.
- $Config_{init}$: un arrangement initial des N agents Conteneur dans les M emplacements disponibles.
- i et j : désigne deux agents Conteneur donnés.
- x_i : désigne le numéro de la pile à laquelle appartient l'agent Conteneur i , $x_i \in \{0, \dots, NP - 1\}$.
- y_i : désigne le niveau de l'agent Conteneur i dans la pile x_i , $y_i \in \{0, \dots, H_{max}-1\}$.
- (x_i, y_i) : couple qui désigne les coordonnées ou encore l'emplacement d'un agent Conteneur i dans la zone de stockage S .
- $EmpInit_i$: l'emplacement initial d'un agent Conteneur i .
- Cat_i : désigne la catégorie de l'agent Conteneur i , $Cat_i \in \{A, B, C, D, E, F, \dots, Y, ou Z\}$.
- $ACSous_i$: désigne l'agent Conteneur placé en dessous de l'agent Conteneur i .
- $ACSus_i$: désigne l'agent Conteneur placé au dessus de l'agent Conteneur i .
- $NivRech_i$: le niveau de recherche d'un agent Conteneur i .
- $mACont$: désigne la matrice des différents agents Conteneur : leurs coordonnées et leur catégories. Elle décrit la disposition des agents Conteneur dans la zone du Stockage S .
- $lCherPlace_i$: liste des agents Conteneur cherchant une place sur l'agent Conteneur i .
- $lAContLibre_i$: liste des agents Conteneur libres pour recevoir l'agent Conte-

neur i au dessus.

- $EmpSol_i$: c'est l'emplacement solution courante d'un agent Conteneur i .
- $Etat_i$: désigne l'état d'un agent Conteneur i . Cet état peut être satisfait ou insatisfait.

Définitions

- AC : c'est un agent Conteneur associé à un conteneur donné.
- L'identificateur d'un agent Conteneur i est sous la forme : " $AC_$ " + $Cat_i x_i y_i$ (exp. AC_B02).
- La tête d'une pile est un agent Conteneur ayant pour niveau égal à la hauteur de cette pile.
- Un AC est dit " libre " s'il n'est pas recouvert par des AC s et si son niveau ne dépasse pas la hauteur maximale. Autrement dit, s'il dispose d'au moins un emplacement vide dessus.
- Un AC dérange ou gêne l' AC en-dessous immédiatement si ce dernier est de catégorie A ou s'il dérange à son tour un autre AC .
- Un AC est satisfait s'il occupe un emplacement respectant les contraintes spatio-temporelles.
- Un AC est insatisfait s'il est encore en train de rechercher un emplacement respectant les contraintes spatio-temporelles.
- L'état d'un AC est soit satisfait ou insatisfait.
- L' $i^{ème}$ niveau de recherche regroupe tous les agents Conteneur dérangeant immédiatement l' $i^{ème}$ occurrence de l'agent Conteneur de catégorie A.
- Un niveau de recherche est un numéro séquentiel : c'est l'occurrence d'un AC de catégorie A rencontré dans la configuration parcourant les piles du haut vers le bas à condition qu'un autre AC de catégorie A ne soit pas au-dessus.
- Un AC appartient à l' $i^{ème}$ niveau de recherche s'il dérange immédiatement l' $i^{ème}$ occurrence d'un AC de catégorie A.
- Un AC est soit parti (pour être chargé dans un navire, train ou camion), soit arrivé (déchargé d'un navire, camion ou train).
- Un AC est dit perturbé s'il dérange un autre AC , d'une autre manière son niveau de recherche est strictement positif. Cet agent doit être déplacé.

- Un plan de déplacements ou mouvements est une séquence d’actions constituée de couples de type : (soulever-déposer) ou (dépiler-empiler).
- Une solution individuelle d’un agent Conteneur i notée $EmpSol_i$, correspond à un emplacement retenu. Autrement dit, elle correspond à un plan partiel de déplacements qui lui permet d’atteindre cet emplacement se limitant à la satisfaction des contraintes de hauteur de piles, du nombre de piles et de date de départ.
- Une solution finale globale est la fusion des différentes solutions individuelles. Autrement dit, il s’agit d’un plan global de déplacements des agents Conteneur. Ce plan émerge de la fusion des plans partiels, certainement disjoints, des agents Conteneur. Subséquemment, cette solution se restreint à la satisfaction des contraintes spatio-temporelles.
- La situation de blocage non solvable correspond au cas où il n’existe aucune solution à part la mise en disposition d’une zone tampon supplémentaire.

Architecture Comme nous l’avons déjà mentionné, le modèle DistNIS se base sur un algorithme distribué de recherche non informée. Dans cette section, nous allons décrire les différents agents de DistNIS à savoir : l’agent Conteneur (représentant un conteneur) et l’agent Interface. Chacun de ces agents est défini par ses accointances (les agents avec lesquels il peut communiquer), ses connaissances (statiques et dynamiques), ainsi que ses méthodes décrivant son comportement.

Agent Conteneur A chaque conteneur traité, nous associons un agent Conteneur défini par :

- Ses accointances formées par les autres agents Conteneur et l’agent Interface.
- Ses connaissances statiques : son identificateur, son emplacement initial $EmpInit$, son port provenance, son port destination, sa tare, sa charge, sa dimension (1 ou 2EVP).
- Ses connaissances dynamiques : sa catégorie, $EmpSol$, $ACSous$, $ACSus$, $lCherPlace$, $lAContLibre$ et son état.

Agent Interface Cet agent communique avec la totalité des agents Conteneur. D'abord il les crée, puis il déclenche le processus de résolution relatif à un seul départ, assure son déroulement, détecte sa terminaison (qui est équivalente à la satisfaction de chaque agent Conteneur). Ensuite, avant le prochain départ, il génère des arrivées ayant lieu éventuellement entre ces deux départs successifs. Aussitôt les conteneurs arrivés sont placés, il déclenche l'exécution du prochain départ (ou encore cycle). Enfin, il fournit le résultat final formé par la solution globale finale (le coût de cette solution en terme du nombre de mouvements parasites).

Dynamique globale Dans cette section, nous décrivons la dynamique globale du modèle DisNIS : les étapes principales du processus de résolution et les messages les plus importants échangés entre les différents agents Conteneur. Après leur création et initialisation (connaissances statiques et dynamiques, accointances, matrice des catégories), les agents Conteneur communiquent et interagissent dans le but de permettre à chaque agent de retrouver un emplacement ou encore construire une solution individuelle qui satisfait uniquement les contraintes spatio-temporelles sans se soucier des mouvements parasites pouvant avoir lieu ultérieurement. De ce fait, une solution globale finale de notre problème est construite de manière distribuée en fusionnant les solutions individuelles des différents agents Conteneur. Une solution individuelle n'est pas atteinte indépendamment des autres agents, plutôt c'est le résultat de leur interaction. Mais, il faut signaler qu'au cours de cette interaction coopérative dans son ensemble, les agents Conteneur se comportent de manière " naïve " et " égoïste " à la fois puisque chacun d'eux cherche à être placé même en dérangeant un autre agent Conteneur en -dessous et même s'il va être déplacé après. Cette interaction, formant le processus de résolution de base, est assurée via un échange direct de messages entre un agent Conteneur i et un agent Conteneur j :

- 'Géné' : i veut se libérer de j qui est au dessus.
- 'ChercherPlace' : i recherche un emplacement disponible s'il dérange j .
- 'JeSuisLibre' : i est libre et il peut accueillir j au-dessus, il lui diffuse ce message.

- 'TuEsLibre' : i retrouve un nouvel emplacement, alors il informe l'agent Conteneur en-dessous j qu'il est devenu libre.
- 'Désolé' : i informe j qu'il ne peut pas l'accepter dessus. i est désolé s'il a déjà un agent Conteneur au dessus ou si son niveau coïncide avec la hauteur maximale de la pile.
- 'Migrer' : i demande à j s'il peut être placé dessus.
- 'Confirmer' : i accepte définitivement j dessus.
- 'Rejeter' : i rejette la demande de migration de j .

Dynamique détaillée Rappelons que le processus de résolution que nous proposons vise à ce que les agents Conteneur perturbés ou nouvellement arrivés retrouvent des emplacements appropriés suite aux départs d'agents Conteneur prioritaires A. Dans cette section, nous allons décrire de manière exhaustive ce processus en regardant de plus près le comportement des différents agents Conteneur.

Initialisation des agents Conteneur A la première date de départ, nous devons assurer le départ des agents Conteneur de catégorie A et le réarrangement des agents Conteneur perturbés. Aux dates de départ suivantes, nous devons assurer en plus l'arrangement des agents Conteneur arrivant entre deux départs successifs. Avant d'entamer la description de la démarche poursuivie pour la résolution de notre problème, nous devons présenter un nouveau paramètre que nous avons défini : niveau de recherche relatif à un agent Conteneur. Ce paramètre permet de répartir l'ensemble de nos agents Conteneur en sous-ensembles : un sous ensemble E_i est associé à l' $i^{\text{ème}}$ niveau de recherche s'il contient les agents Conteneur (de catégories différentes de A) empilés immédiatement sur l' $i^{\text{ème}}$ occurrence d'un agent Conteneur de catégorie A en considérant la pile du haut vers le bas. La $(i+1)^{\text{ème}}$ occurrence d'un agent Conteneur A correspond à l'agent Conteneur A qui est séparé du dernier agent Conteneur A de l' $i^{\text{ème}}$ occurrence par des agents Conteneur de catégories différentes de A (les agents Conteneur A empilés les uns sur les autres de manière successive et non intercalés par des agents Conteneur de catégories différentes appartiennent à une même occurrence). Deux agents conteneur appartenant à deux niveaux de recherche différents sont inter-

calés par un ou plusieurs agents Conteneur A successifs. Ce nouveau paramètre nous a été d'une grande utilité dans le contrôle de plusieurs états tout au long de notre processus de résolution. A l'étape courante, ce paramètre nous permet d'identifier facilement les agents Conteneur perturbés, ce sont ceux dont le niveau de recherche est non nul. En effet, le processus de résolution commence par les repérer et les initialiser à 'Insatisfaits'. Ensuite deux comportements sont déclenchés simultanément : Comportement de départ pour les agents Conteneur A et un comportement réactif pour les agents Conteneur perturbés. Ce comportement global est schématisé par l'algorithme 4.1.

```

Procédure Globale ()
Début
1-Si  $NivRech_i = 0$  alors
2-  $Etat_i \leftarrow$  'satisfait'
3-Sinon
4-  $Etat_i \leftarrow$  'insatisfait'
5- Si  $Cat_i = 'A'$  alors
6-   ComportDep ()
7- ComportReact()
Fin

```

Algorithme 4.1: Comportement global des agents Conteneur

Comportement de départ A chaque départ, les agents Conteneur A (s'il y en a éventuellement) sont dépilés pour être chargés sur le navire, camion ou train associé. Deux cas se présentent : certains agents Conteneur A qui sont directement accessibles et qui ne sont pas gênés par des agents Conteneur ayant autres catégories, alors ils libèrent leurs emplacements et envoient aux agents Conteneur en dessous des messages TuEsLibre() (Algorithme 4.2, lignes 1-4). D'autres agents Conteneur A ne peuvent pas partir puisqu'ils sont gênés d'un ou plusieurs agents Conteneur de catégorie supérieure et donc moins prioritaires. Ces derniers doivent être déplacés afin de libérer les agents Conteneur A suite à la réception des messages Gêné() (Algorithme 4.2, lignes 8-9). Par conséquent, des mouvements parasites supplémentaires sont procréés. Afin de bien mener ce

départ, nous devons respecter d'abord la contrainte fonctionnelle suivante : deux agents Conteneur A d'occurrences différentes ne peuvent pas partir simultanément. Cette contrainte est satisfaite en faisant appel au paramètre de niveau de recherche : les agents Conteneur A d'une même occurrence ne peuvent pas partir sans avoir déplacé et réarrangé les agents Conteneur de catégorie supérieure appartenant au niveau de recherche associé.

```

Procédure ComportDep()
Début
1-Si  $Cat_{ACSus_i} = 'R'$  alors
2-  $mACont(x_i, y_i) \leftarrow ' '$ 
3- envoyer TuEsLibres () à  $ACSous_i$ 
4-  $nbrAContA \leftarrow nbrAContA - 1$ 
5- Si  $nbrAContA = 0$  alors
6-     envoyer TraiterArriv () à l'agent Interface
7- Tuer( $i$ )
8-Sinon
9- envoyer Gêné() à  $ACSus_i$ 
Fin

```

Algorithme 4.2: Comportement de départ des agents Conteneur A

Comportement réactif A la suite de départs des agents Conteneur A, les agents Conteneur perturbés d'un même niveau de recherche exécutent simultanément un comportement réactif déclenché suite à la réception des différents messages tel que schématisé dans l'algorithme 4.3.

Ce comportement est déclenché par la réception du message Gêné () reçu des agents Conteneur A à leur départ (Algorithme 4.2). Un agent Conteneur recevant ce message, soit qu'il est de catégorie A ou autre, va essayer de libérer son emplacement, pour partir ou rechercher un nouvel emplacement. S'il n'est pas à son tour libre alors il envoie à l'agent Conteneur au-dessus le message Gêné() (Algorithme 4.4). Les agents Conteneur perturbés diffusent aux autres agents Conteneur (de n'importe quelle autre catégorie) des messages RechercherPlace() tel que indiqué dans l'algorithme 4.5 (Lignes 2-4).

```

Procédure ComportReact()
Début
1- Si msg <> 'Null' alors
2-   Si msg = 'Gêné' alors
3-     JeGêne ()
4-   Si msg = 'ChercherPlace' alors
5-     ChercherPlacePour (j)
6-   Si msg = 'JeSuisLibre' alors
7-     IlEstLibre (j)
8-   Si msg = 'Désolé' alors
9-     IlEstDésolé (j)
10-  Si msg = 'Migrer' alors
11-    SurMoi (j)
12-  Si msg= 'Confirmer' alors
13-    IlConfirme (j,Etatj)
14-  Si msg= 'Rejeter' alors
15-    IlRejete ()
16-  Si msg='MettreAJour' alors
17-    MaJCoord (msg)
18-  Si msg='TuEsLibres' alors
19-    JeSuisLibre (j)
20-  Si msg= 'Quitter'alors
21-    Tuer (i)
22-Sinon
23-  Etati ← 'attente'
Fin

```

Algorithme 4.3: Messages échangés entre les agents Conteneur

Les agents Conteneur, satisfaits ou pas, recevant les demandes de recherche d'emplacements, envoient des réponses positives de type JeSuisLibre() ou négatives de type Désolé() selon qu'ils sont libres ou pas et que la hauteur maximale des piles auxquelles ils appartiennent est déjà atteinte ou pas (Algorithme 4.6, Lignes 1, 2 et 6). Dans ce traitement schématisé par l'algorithme 4.6, nous tenons compte des contraintes spatiales du problème.

A la réception des réponses JeSuisLibre (), chaque agent Conteneur, s'il est encore insatisfait sauvegarde cette réponse dans une la liste $lAContLibre_i$, et

```

Procédure JeGêne ()
Début
1-Si  $CatACSus_i \neq 'R'$  alors
2-   envoyer Gêne () à  $ACSus_i$ 
3-Sinon
4-   chercherPlace ()
Fin

```

Algorithme 4.4: Comportement des agents Conteneur recevant le message gène()

```

Procédure ChercherPlace ()
Début
1-Pour  $n \in lACont$ 
2-   Si ( $(n \neq i)$  et  $(n \neq ACSous_i)$ ) alors
3-     envoyer ChercherPlace() à  $n$ 
4-      $nbrRechPlaceDem \leftarrow nbrRechPlaceDem + 1$ 
Fin

```

Algorithme 4.5: Comportement des agents Conteneur perturbés

```

Procédure ChercherPlacePour ()
Début
1- Si ( $CatACSus_i = 'R'$ ) et  $(y_i < H_{max})$  alors
2-   envoyer JeSuisLibre ( $y_i$ ) à  $j$ 
3- Sinon
4-   Si  $(j \in lCherPlace_i)$  alors
5-      $lCherPlace_i \leftarrow lCherPlace_i \cup \{j\}$ 
6-     envoyer Désolé () à  $j$ 
Fin

```

Algorithme 4.6: Comportement des agents Conteneur recevant le message ChercherPlace()

répond à une proposition (selon la politique FIFO) en envoyant le message Migrer() et reste en attente d'une confirmation définitive ou d'un rejet définitif (Algorithme 4.7, lignes 2-5). A la réception d'un message Désolé () (Algorithme 4.8, lignes 7-12), il envoie de nouveau le message Migrer () à une proposition suivante stockée dans $lAContLibre_i$. Ceci n'empêche que l'agent Conteneur expéditeur du message Désolé() stocke ces agents Conteneur chercheurs de place dans une liste $lCherPlace_i$. Si un agent Conteneur ne reçoit que des messages Désolé(), alors nous avons une situation de blocage non solvable c'est-à-dire soit que toutes piles sont saturées ou contenant des agents Conteneur perturbés. Dans ce dernier cas, nous alternons entre des messages de types : ChercherPlace()-JeSuisLibre()-Migrer()-Gêné() jusqu'à l'infini, nous ne pouvons résoudre cette situation qu'en ayant une zone de stockage supplémentaire ou encore zone tampon. Notons qu'une fois devenu libre ou satisfait, un agent Conteneur envoie de nouveau des messages JeSuisLibre() aux agents Conteneur chercheurs de place stockés déjà dans $lCherPlace_i$.

```

Procédure IlEstLibre (j)
Début
1-  $nbrLibreRep \leftarrow nbrLibreRep + 1$ 
2- Si ( $Etat_i = 'instatisfait'$ ) et ( $Cat_{ACSus_i} = 'R'$ ) alors
3-   Si ( $nbrLibreRep = 1$ ) et  $\neg(attente)$  alors
4-      $attente \leftarrow Vrai$ 
5-     Envoyer Migrer() à j
6-   Sinon
7-     Si  $j \notin lAContLibre_i$  alors
8-        $lAContLibre_i \leftarrow lAContLibre_i \cup \{j\}$ 
Fin

```

Algorithme 4.7: Comportement des agents Conteneur recevant le message JeSuisLibre()

A la réception du message Migrer() (voir algorithme 4.9) d'un ou plusieurs agents Conteneur en attente de confirmation ou de rejet, un agent Conteneur n'accepte au-dessus qu'un seul agent Conteneur qui correspond à l'expéditeur du premier message Migrer (), il lui envoie alors le message Confirmer() et envoie aux

```

Procédure IlEstDesole (j)
  Début
  1-  $nbrDesoleRep \leftarrow nbrDesoleRep + 1$ 
  2- Si ( $nbrDesoleRep = nbrRechPlaceDem$ ) alors
  3-    $nbrDesoleRep \leftarrow 0$ 
  4-    $nbrRechPlaceDem \leftarrow 0$ 
  5-   chercherPlace()
  6- Sinon
  7-   Si ( $nbrDesoleRep + nbrLibreRep \geq nbrRechPlaceDem$ )
  8-     et ( $Etat_i = 'insatisfait'$ ) et ( $Cat_{ACSus_i} = 'R'$ ) alors
  9-        $EmpSol \leftarrow NouvEmpAlea(lAContLibre_i)$ 
  10-      Si  $EmpSol \neq NULL$  alors
  11-         $attente \leftarrow 'Vrai'$ 
  12-        Envoyer Migrer () à  $EmpSol$ 
  Fin

```

Algorithme 4.8: Comportement des agents Conteneur recevant le message Désolé()

autres le message Rejet (). Cette confirmation est envoyée même s'il est insatisfait. Ce qui pourrait certainement générer un mouvement inévitable puisque l'agent Conteneur destinataire doit se déplacer après.

Un agent Conteneur, étant en attente d'une réponse définitive, reçoit soit une confirmation définitive : dans ce cas (voir algorithme 4.10) il libère son emplacement courant, envoie le message TuEsLibre() à l'agent Conteneur en dessous, et occupe son nouvel emplacement solution. Mais il ne devient satisfait que si le nouvel agent Conteneur en dessous est lui-même satisfait (Algorithme 4.10, lignes 1-3). En conséquence, le nombre de mouvements est incrémenté, voir qu'il va être encore incrémenté si l'agent Conteneur en dessous est insatisfait ; soit un rejet

(Algorithme 4.11) : dans ce cas, l'agent Conteneur en question recherche la proposition suivante stockée dans $lAContLibre_i$ et relance le même processus d'envoi de messages ChercherPlace() -Migrer.

Un agent Conteneur, devenu libre (Algorithme 4.12, lignes 1-4) : soit qu'il est de catégorie A alors il quitte à son tour son emplacement et part. Soit qu'il est de catégorie différente et insatisfait, alors il recherche à son tour un nouvel

```

Procédure SurMoi (j)
  Début
  1- nbrSurMoi ← nbrSurMoi + 1
  2- Si nbrSurMoi = 1 alors
  3-   envoyer Confirmer(Etati) à j
  4-   envoyer MettreAJour (xi, yj) à j
  5-   Si (Etati = 'insatisfait') alors
  6-     JeGene ()
  7- Sinon
  8-   Envoyer Rejeter() à j
  Fin

```

Algorithme 4.9: Comportement des agents Conteneur recevant le message Migrer ()

```

Procédure IlConfirme (j, Etatj)
  Début
  1- Si (Etati = 'insatisfait') alors
  2-   Si (Etatj = 'satisfait') alors
  3-     Etati = 'satisfait'
  4-     nbrMvt ← nbrMvt + 1
  5-     Envoyer TuEsLibres() à ACSousi
  6-     CatACSusi ← 'R'
  Fin

```

Algorithme 4.10: Comportement des agents Conteneur recevant le message Confirmer ()

emplacement. Et enfin, soit qu'il est de catégorie différente et satisfait, il envoie dans ce cas aux agents Conteneur stockés déjà dans sa liste $lCherPlace_i$ et ayant reçu préalablement le message Désolé (), le message JeSuisLibre () (Algorithme 4.12, lignes 12-15).

Le comportement de départ et le comportement réactif, ainsi détaillés, sont exécutés par niveau de recherche à chaque départ, sauf qu'à partir du deuxième

```

Procédure IlRejete (j)
Début
1- Si (IndCour < |lAContLibrei|) et
   (Etati = 'insatisfait') et (attente) et (CatACSusi = 'R') alors
2-   EmpSol ← NouvEmpAlea (lAContLibrei)
3-   attente ← 'Vrai'
4-   Envoyer Migrer () à EmpSol
5- Sinon
6-   attente ← 'faux'
Fin

```

Algorithme 4.11: Comportement des agents Conteneur recevant le message Rejeter ()

```

Procédure JeSuisLibre (j)
Début
1- Si Cati = 'A' alors
2-   Envoyer TuEsLibres() à ACSousi
3-   mACont(xi, yi) ← ''
4-   nbrAContA ← nbrAContA - 1
5-   Si nbrAContA = 0 alors
6-     envoyer TraiterArriv () à l'agent Interface
7-   Tuer (i)
8- Sinon
9-   CatACSusi ← 'R'
10-  Si Etati = 'insatisfait' alors
11-    ChercherPlace()
12-  Sinon
13-    Si |lCherPlacei| > 0 alors
14-      Pour n ∈ lCherPlacei faire
15-        envoyer JeSuisLibre (yi) à n
16-      RéinitList()
Fin

```

Algorithme 4.12: Comportement des agents Conteneur recevant le message TuEsLibre ()

départ, nous intégrons des arrivées d'agents Conteneur tel que nous allons le montrer dans la section suivante. Ces comportements, exécutés simultanément par les agents Conteneur de notre modèle, permettent de simuler un comportement aléatoire de l'activité de stockage des conteneurs sur un quai, et de fournir la solution globale. Cette dernière correspond à une séquence de déplacements associés aux différentes solutions individuelles qui satisfait les contraintes spatio-temporelles et les règles fonctionnelles du problème sans se soucier du nombre de mouvements engendrés.

Traitement des arrivées Les arrivées des agents Conteneur ayant lieu entre deux départs successifs sont traitées de la manière suivante : Une fois tous les agents Conteneur A sont partis au premier départ, l'agent interface génère des agents Conteneur arrivés à chaque départ (Algorithme 4.2, lignes 4-7 et algorithme 4.12, lignes 5-8). Ces agents, initialement insatisfaits, sont injectés/empilés de manière aléatoire dans la zone de stockage courante pour être traités par la suite identiquement à des agents Conteneur perturbés. A partir du moment où le dernier agent Conteneur arrivé est injecté (voir algorithme 4.13), nous relançons le processus de résolution décrit précédemment.

Situation de blocage Comme nous l'avons déjà mentionné, nous pouvons rencontrer une situation de blocage non solvable au cours de l'exécution du modèle DistNIS. Dans ce cas, nous aurons recours nécessairement à une zone tampon supplémentaire où nous placerons les agents Conteneur insatisfaits.

Ce cas de blocage survient si et seulement si :

$$\exists i : N_{p_{i1}} > \sum_{j=1, i \neq j}^{NP} N_{v_j} + N_{A_{j1}} X_{ji} \text{ où } i, j \in \{1, \dots, NP\}, i \neq j$$

```

Procédure MajCoord(msg)
Début
1- Si ( $\neg AContArriv$ ) alors
2-    $mACont(x_i, y_i) \leftarrow ''$ 
3-    $x_i \leftarrow x_j$ 
4-    $y_i \leftarrow y_j + 1$ 
5-    $mACont(x_i, y_i) \leftarrow Cat_i$ 
6-   MaJCat ()
7- Si  $AContArriv$  alors
8-    $nbrACArrNPlac \leftarrow nbrACArrNPlac - 1$ 
9-   Si  $nbrACArrNPlac = 0$  alors
10-    envoyer DepTousA () à l'agent Interface
11-Si ( $|lCherPlace_i| > 0$ ) et ( $Cat_{ACSus_i} = 'R'$ ) et ( $y_i < H_{max}$ ) alors
12-  Pour  $n \in lCherPlace_i$  faire
13-    envoyer JeSuisLibre ( $y_i$ ) à  $n$ 
14-Si ( $Etat_i = 'satisfait'$ ) alors
15-  RéinitList()
Fin

```

Algorithme 4.13: Traitement des arrivées à l' $i^{\text{ème}}$ départ

où

N_{pi1} : le nombre d'agents Conteneur perturbés de la pile i du premier niveau de recherche

N_{vj} : le nombre de places vides dans la pile j

N_{Aj1} : le nombre d'agents Conteneur de catégorie A de la première occurrence dans la pile j

$$X^{ji} = \begin{cases} 1, & \text{Si un agent Conteneur A de la pile } j \text{ part avant celui de la pile } i; \\ 0, & \text{Sinon.} \end{cases}$$

Cette condition peut être démontrée par récurrence comme suit :

Preuve :

Pour alléger et faciliter la lisibilité de notre condition, soient :

$N_v = \sum_{j=1, i \neq j}^{NP} N_{vj}$: le nombre de places vides dans toutes les piles différentes de i ,

et $N_A = N_{A_{j1}} X_{ji}$: le nombre d'agents Conteneur A qui partent.

Nous pouvons vérifier qu'il existe une pile tel que : Si $N_{p_{i1}} = 1$ et $N_v + N_A$ est égal à 0, alors le processus de résolution se bloque.

Supposons qu'il existe une pile où $N_{p_{i1}} = n$, nous avons une situation de blocage ssi $n > N_v + N_A$.

Montrons alors qu'il existe une pile où $N_{d_{i1}} = n + 1$, nous avons une situation de blocage ssi $n + 1 > N_v + N_A$.

Nous avons par hypothèse $n > N_v + N_A$ Or : $n + 1 > n > N_v + N_A$ d'où : $n + 1 > N_v + N_A$

Par conséquent, il existe une pile tel que : Si $n > N_v + N_A \Leftrightarrow$ une situation de blocage.

Nous pouvons aussi prouver cette condition en démontrant son dual :

$\forall i : N_{p_{i1}} \leq N_v + N_A \Leftrightarrow$ Il existe au moins une solution.

Preuve :

Pour une pile contenant un seul agent Conteneur perturbé :

Si $N_{p_{i1}} = 1 \leq N_v + N_A \Leftrightarrow$ pas de blocage puisque l'agent Conteneur peut trouver un emplacement parmi les places vides ou parmi les places qui ont été occupées par des agents Conteneur A.

Supposons que Si $N_{p_{i1}} = n \leq N_v + N_A$ alors il existe une solution (1).

Montrons qu'il existe aussi une solution pour $N_{p_{i1}} = n + 1$.

Par hypothèse, nous avons $n \leq N_v + N_A \Leftrightarrow n \leq N_v + N_A + 1 \Leftrightarrow$ Il existe une solution puisque n agents Conteneur parmi les $n + 1$ agents Conteneur perturbés peuvent retrouver au moins n emplacements selon (1), et par conséquent libérer un ou plusieurs A en dessous. L'agent Conteneur perturbé non encore satisfait peut occuper l'emplacement de l'un des agents Conteneur A déjà partis.

Ainsi, les $n + 1$ agents Conteneur sont satisfaits. D'où :

$\forall i : N_{p_{i1}} \leq N_v + N_A \Leftrightarrow$ Il \exists au moins une solution.

La description du noyau DistNIS de COSAH permet d'appréhender le principe de base du modèle COSAH, de mettre en avant la résolution distribuée du PSC via les systèmes multi-agent, et de simuler l'activité de stockage de conteneurs dans la un terminal à conteneurs. Désormais, le lecteur pourra librement se servir de ce noyau comme composant de base d'un autre modèle qu'il souhaite

développer. Il pourra intégrer autant de modules complémentaires qu'il vaudra afin d'atteindre la deuxième finalité de COSAH qu'est l'optimisation du PSC. Ceci pourra être achevé en introduisant des méthodes exactes ou approchées au sein du comportement des agents de DistNIS. Dans le modèle COSAH, nous avons développé un module complémentaire nommé ISM basé sur une méthode heuristique, qui fera l'objet de la suite de ce chapitre.

4.3.2.2 DistIS

Un module complémentaire appelé ISM, mettant en œuvre des heuristiques constructives gloutonnes, est désormais injecté dans DistNIS de manière à émerger un algorithme distribué de recherche informée nommé DistIS (*Distributed Informed Search algorithm*). DistIS constitue le cœur de COSAH et en fait un modèle d'optimisation heuristique distribuée. Il a pour but d'optimiser l'activité de stockage de conteneurs par le biais de ISM, et ce en minimisant le nombre de mouvements parasites pouvant avoir lieu lors des opérations de chargement ou de déchargement des conteneurs, tenant compte des contraintes d'espace et de temps. L'idée principale de DistIS est de déterminer de manière guidée de nouveaux emplacements aussi bien pour les conteneurs perturbés après le départ des conteneurs A, que pour les conteneurs nouvellement arrivés. Certes, il y a des points communs entre DistIS et DistNIS, mais nous allons nous concentrer sur les notions et concepts spécifiques à DistIS. Ça ne serait ce que ceux introduits par le module complémentaire ISM. Notons que quelques éléments de la structure de la section précédente peuvent être repris.

Notations Dans ce modèle, nous avons adopté la plupart des notations déjà invoquées dans le modèle DistNIS (voir 4.3.2.1) : P_{max} , H_{max} , N , $Config_{init}$, H_p , i et j , x_i , y_i , (x_i, y_i) , Cat_i , $ACSous_i$, $ACSus_i$, $NivRech_i$, $mACont$, $lCherPlace_i$, $EmpSol_i$ et $Etat_i$. Nous avons rajouté quelques notations spécifiques au modèle DistIS :

- $lAContPref_i$: liste des agents Conteneur que l'agent Conteneur i préfère de se mettre au-dessus selon des critères bien définis, d'une autre manière c'est la liste des emplacements disponibles préférés par i .

- $lAContNonPref_i$: liste des agents Conteneur non préférés par l'agent Conteneur i .
- $Etat_i$: désigne l'état d'un agent Conteneur i . Cet état peut être insatisfait en tolérance, insatisfait, ou satisfait.
- $Dist_{ij}$: c'est la différence entre les catégories des agents Conteneur i et j définie comme suit :

$$Dist_{ij} = \begin{cases} Cat_j - Cat_i & \text{Si } Cat_j \geq Cat_i; \\ -(Cat_j - Cat_i) & \text{Sinon.} \end{cases}$$

Définitions Dans la section suivante nous donnons les définitions des concepts proposés dans DistIS ajoutées à celles données dans la section (4.3.2.1).

- La préférence d'un agent Conteneur pour un ou plusieurs agents Conteneur exprime le fait que cet agent préfère se placer au-dessus de l'un d'entre eux. Elle permet de déterminer les nouveaux emplacements qu'il préfère y migrer ou occuper.
- La tolérance est une procédure déclenchée si une situation de blocage solvable est détectée au cours du processus de résolution afin assurer la convergence du modèle.
- La situation de blocage solvable correspond au cas où toutes les têtes des piles sont perturbées et il existe au moins une solution pour le déblocage.
- Un agent conteneur est dit tolérant s'il est insatisfait mais il tolère comme même à un agent Conteneur de se placer au-dessus de manière provisoire, dans le cas d'état de blocage solvable de la résolution.
- Un AC est satisfait s'il occupe définitivement un emplacement respectant les contraintes spatio-temporelles et minimisant le nombre de mouvements qu'il peut effectuer.
- Un AC est insatisfait s'il est en train de chercher un nouvel emplacement respectant les contraintes spatio-temporelles et minimisant le nombre de mouvements qu'il peut effectuer. Dans une situation de blocage solvable, cet agent est placé temporairement au-dessus d'un autre agent Conteneur qui est lui-même insatisfait. En effet, pour distinguer ce cas, nous introduisons un troisième état possible d'un agent Conteneur : insatisfait en tolérance.

- L'état d'un *AC* peut être : satisfait, insatisfait en tolérance ou insatisfait.
- Une solution individuelle d'un agent Conteneur *i* notée *EmpSol_i*, correspond à un emplacement retenu. Autrement dit, elle correspond à un plan partiel de déplacements qui lui permet d'atteindre cet emplacement tout en satisfaisant les contraintes de hauteur de piles, du nombre de piles et de date de départ et d'autre part ; et minimisant ses mouvements dans le futur.
- Une solution finale globale est la fusion des différentes solutions individuelles. Autrement dit, il s'agit d'un plan global de déplacements émergeant de la fusion des plans partiels des agents Conteneur. Subséquemment, cette solution satisfait les contraintes spatio-temporelles et minimise le nombre total de mouvements effectués.

Architecture Le modèle DistIS met en jeu aussi l'agent Interface et l'agent Conteneur, l'agent Interface garde les mêmes structures et rôle que ceux définis dans DistNIS. L'agent Conteneur conserve les mêmes accointances et connaissances statiques, mais possède de nouvelles connaissances dynamiques et change en majorité de comportement. Les connaissances dynamiques sont : sa catégorie *Cat*, son emplacement initial *EmpSol*, l'agent Conteneur dessous immédiatement *ACSous*, l'agent Conteneur dessus immédiatement *ACSus*, *lCherPlace*, *lAContPref_i*, *lAContNonPref_i* et son état.

Dans le modèle DistIS, l'architecture interne d'un agent Conteneur est illustrée par la figure 4.10. Dans cette figure, nous insistons sur le module de prise de décision qui reflète un comportement de l'agent Conteneur différent de celui vu jusqu'à maintenant, et ce en intégrant un module d'optimisation qui met en jeu des heuristiques constructives gloutonnes chaque fois qu'il s'agit d'une décision à prendre.

Dans ce qui suit, nous nous étalons sur ce module et précisément sur la méthode heuristique adoptée dans COSAH : quelles sont les heuristiques mises en place ? Comment et quand sont-elles déclenchées ? comment fonctionnent-elles ? Et quelle est leur utilité ?

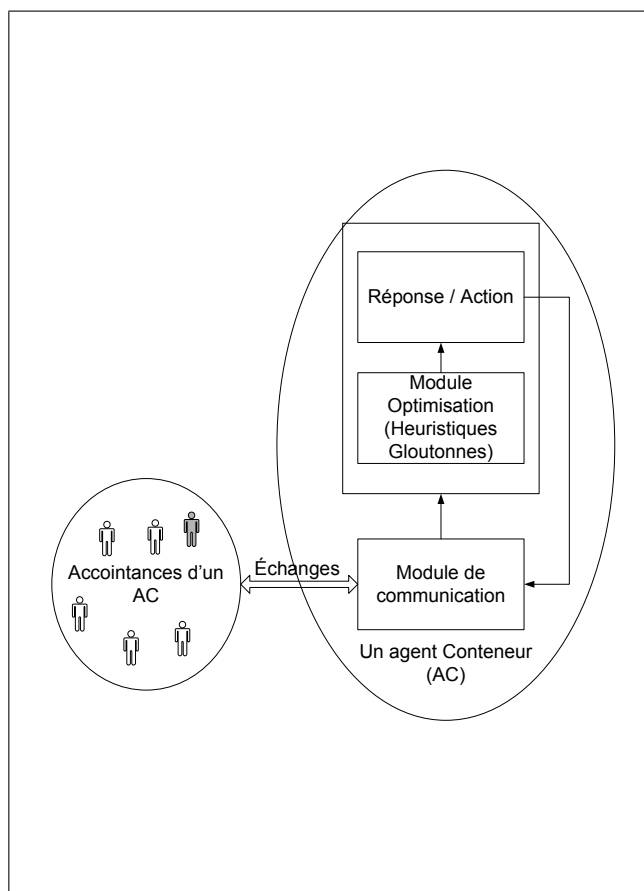


FIG. 4.10 – Composants du module de prise de décision d'un agent Conteneur

Dynamique globale Rappelons que le modèle DistIS a pour but essentiel celui de déterminer une solution globale qui minimise le nombre de mouvements parasites total et qui satisfait les contraintes déjà évoquées. Après leur création et initialisation, les différents agents Conteneur s'exécutent simultanément déclenchant leurs propres algorithmes afin de donner naissance à cette solution. Chaque agent Conteneur participe de sa part à la constitution de cette solution globale, et ce en recherchant une solution individuelle qui minimise ses propres mouvements inutiles tout en respectant les contraintes définies dans son environnement, et en coopérant avec les autres agents Conteneur. Cette coopération est assurée à travers un protocole de négociation muni d'un certain nombre de messages échangés entre ces agents. A chaque date de départ, un cycle de traitements

simultanés est lancé par chaque agent conteneur. Soit que ce dernier est concerné par ce départ, alors il exécute un comportement de départ, soit qu'il est perturbé ou nouvellement arrivé, alors il déclenche un processus de recherche d'une solution individuelle qui puisse minimiser ses déplacements. Dans ce deuxième cas, chaque agent conteneur commence par diffuser un message de recherche d'un nouvel emplacement aux autres agents Conteneur. Au fur et à mesure de la réception de leurs différentes propositions, il exécute un algorithme de recherche informée de la solution individuelle la plus satisfaisante, faisant appel à des heuristiques constructives gloutonnes. Une fois devenu satisfait, il ne se retire pas du processus de résolution mais il essaiera de satisfaire les demandes de recherche d'emplacements qu'il a reçues des autres agents Conteneur non encore satisfaits. Si tous les agents atteignent des solutions individuelles, la solution globale est constituée et un autre cycle de ce même processus est relancé au prochain départ. Dans le cas d'une situation de blocage solvable, nous déclenchons une procédure de tolérance bien définie. Pour la description de ce raisonnement, nous pouvons nous référer aux algorithmes de la section précédente, quoiqu'il faut signaler que les procédures associées sont traitées différemment dans DistIS, et plus précisément au niveau de : l'algorithme 4.7 (Lignes 7,8), l'algorithme 4.8 (ligne 9), et l'algorithme 4.11 (ligne 2). En effet, pour décider de son nouvel emplacement, un agent Conteneur ne fait plus son choix dans la même liste $LAContLibre_i$ de manière aléatoire, mais dans deux listes distinctes $LAContPref_i$ et $LAContNonPref_i$ de manière guidée (voir algorithme 4.14).

4.3.2.3 Dynamique détaillée

Heuristiques adoptées Dans notre contexte, nous ayons recours aux heuristiques constructives gloutonnes afin d'optimiser la résolution du problème sous-jacent. Rappelons d'abord que DistNIS avait pour seul objectif de satisfaire les contraintes spatio-temporelles de notre problème et par conséquent de simuler et de résoudre aléatoirement l'activité de stockage des conteneurs prêts à être chargés, les conteneurs perturbés qui vont partir ultérieurement et les conteneurs arrivant entre deux départs. Pour ce, DistNIS se base sur un algorithme distribué de recherche non informée. Cependant, DistIS ne se suffit pas à la satisfaction des contraintes du problème étudié, mais permet en plus d'optimiser sa fonction

objectif (minimiser le nombre de mouvements parasites pouvant se produire lors des opérations de départ ou d'arrivée de conteneurs en import ou en export). Pour ce, nous mettons en œuvre un algorithme de recherche informée. Pareillement à DistNIS, une solution globale à notre problème est construite de manière distribuée en " fusionnant " les solutions individuelles des différents agents Conteneur, et une solution individuelle n'est pas atteinte indépendamment des autres agents, plutôt c'est le produit de leur coopération. La différence dans DistIS est le fait d'appuyer cette coopération, il ne s'agit plus d'une simple communication entre les agents et d'une détermination aléatoire de la solution individuelle pour un agent. Désormais, chaque agent cherche une solution individuelle qui minimise ses déplacements. Nous pouvons alors nous retrouver dans des situations de conflits lorsque plus qu'un agent Conteneur essaye de se placer dans un même emplacement libre. Parmi les solutions adoptées (voir chapitre 3) pour la résolution de conflits, nous faisons appel à la négociation : les agents Conteneur en conflit entrent dans une série de tractations, d'échanges de messages et de compromis de manière à parvenir à un accord, c'est-à-dire une solution qui satisfasse toutes les parties ; et à la coordination par planification distribuée entre les différents agents Conteneur : lors de l'exécution du processus de recherche de solutions individuelles, les points de conflits sont reconnus et gérés au fur et à mesure et ce en imposant la possibilité de replanifier, ou relancer la recherche de solution individuelle dans notre cas, de manière dynamique. Pour cela, les agents Conteneur doivent être dotés d'une capacité à communiquer leurs intérêts et leurs contraintes individuelles et d'une certaine intelligence qui puisse guider leur comportement. C'est ce que nous intentons d'accomplir dans DistIS. Dorénavant, les agents Conteneur coopèrent via un protocole de négociation avancée, et un agent Conteneur se comporte de manière guidée. Ceci est concrétisé par le biais d'heuristiques constructives gloutonnes spécifiques à notre problème, que nous introduisons dans le comportement interne d'un agent et entre agents afin d'émuler l'intelligence humaine dans la détermination de choix et décisions. A une date de départ t , les agents conteneur perturbés recherchent de nouveaux emplacements afin de libérer les agents Conteneur A concernés par ce départ. Retrouver un nouvel emplacement correspond à la détermination d'une solution individuelle qui respecte les contraintes de nombre de piles et de hauteur maximale et qui mi-

nimise le nombre de mouvements pouvant avoir lieu aux prochains départs et arrivées. La détermination d'une solution individuelle fait appel à un algorithme de recherche informée basée sur des heuristiques constructives gloutonnes (partant d'un emplacement initial, un agent conteneur essaye de retrouver la meilleure solution possible). L'idéal pour chaque agent Conteneur est de ne gêner aucun autre agent Conteneur pour ne pas être obligé à se déplacer encore et donc générer un mouvement parasite de plus, c'est-à-dire chaque agent conteneur a intérêt à être empilé sur un agent Conteneur qui va partir à la même date ou bien à une date ultérieure. D'une autre manière, chaque agent conteneur a intérêt à se placer sur un agent Conteneur de même catégorie ou de catégorie supérieure, et chaque agent Conteneur a un intérêt mutuel celui de ne pas être gêné par un agent Conteneur de catégorie supérieure. Ceci dit, nous respectons toujours la hauteur maximale H_{max} et le nombre de piles disponibles NP . Notons que cette situation idéale est difficilement atteignable tout le temps vu la présence d'intérêts partagés et donc conflictuels. Dès lors, nous sommes amenés à renforcer l'intelligence de nos agents Conteneur afin de leur permettre de conduire un comportement souple, coopératif et rationnel en présence de contraintes et paramètres changeants et imprévisibles (arrivées imprévues, départs estimés). Pour obtenir ces résultats, nous avons fait appel à des heuristiques spécifiques définies selon des critères de préférence d'une solution à une autre. Ces heuristiques sont mises en jeu lors de la recherche d'une solution individuelle. A la réception des messages JeSuisLibre (), chaque agent Conteneur classe les différents agents expéditeurs dans deux listes triées : une liste des agents Conteneur préférés c'est-à-dire ceux qu'il préfère se placer au-dessus, et une deuxième liste des agents Conteneur non préférés. Puis, il envoie au meilleur agent Conteneur le message SurToi (). Cet agent correspond soit à l'agent le plus préféré dans $lAContPref_i$ ou à l'agent le moins non préféré dans $lAContNonPref_i$. Ensuite, il attend soit une confirmation soit un rejet identiquement à ce qui se passe dans le modèle DistNIS, sauf qu'à la réception du message rejeter, l'agent Conteneur ne recherche plus une solution alternative aléatoirement mais choisit la solution immédiatement moins préférée ou plus non préférée. Cet échange de messages se poursuit jusqu'à ce que chaque agent Conteneur retrouve une solution qui minimise le plus le nombre de mouvements qu'il peut effectuer jusqu'à son départ.

Procédure d'évaluation d'une solution La solution globale de notre modèle COSAH est un ensemble de mouvements non conflictuels. Chaque mouvement effectué par un agent Conteneur représente une solution individuelle locale $EmpSol_i$ retenue après une négociation tout au long du processus de recherche. La procédure d'évaluation d'une solution reçue d'un agent Conteneur j est effectuée en parallèle au niveau de chaque agent Conteneur i en calculant un certain coût exprimé par une distance entre i et j dénotée par $Dist_{ij}$. Selon cette distance, l'agent Conteneur i constitue sa liste d'agents Conteneur préférés et sa liste d'agents non préférés. La préférence d'un agent Conteneur est définie comme suit :

$$Pref_{ij} = \begin{cases} 1 & \text{Si } Dist_{ij} \geq 0; \\ 0 & \text{Sinon.} \end{cases}$$

Ainsi, nous pouvons définir formellement les listes $lAContPref_i$ et $lAConNonPref_i$ comme étant deux listes d'agents Conteneur j ($j \neq i$) tel que $Pref_{ij} = 1$ et $Pref_{ij} = 0$, respectivement.

Le coût d'une solution est égal au coût de déplacement d'un agent Conteneur i à un emplacement au-dessus d'un agent Conteneur j . Ce coût est exprimé en terme de nombre de mouvements, il est calculé selon la règle suivante :

$$C_{ij} = \begin{cases} 1 & \text{Si } Dist_{ji} > 0; \\ 0 & \text{Sinon.} \end{cases}$$

Heuristiques de choix d'une solution Tout au long de son processus de recherche, chaque agent Conteneur choisit à chaque fois l'agent Conteneur sur lequel il va se placer éventuellement dans son voisinage formé par les deux listes $lAContPref_i$ et $lAConNonPref_i$. Nous avons défini une relation d'ordre totale entre les différents agents Conteneur des deux listes qui permet de juger la préférence d'un agent Conteneur à un autre (ou encore d'une solution individuelle à une autre). Cette relation est utilisée, d'une part, pour trier les deux listes de chaque agent Conteneur, et d'autre part pour aider cet agent Conteneur perturbé ou nouvellement arrivé à choisir un emplacement où il peut être empilé de la manière la plus optimale (minimisant ses mouvements dans le futur). Ainsi, nous pouvons juger un agent Conteneur j est plus préféré qu'un autre agent Conteneur

k par un agent Conteneur i selon les critères de préférence ou les heuristiques gloutonnes suivantes :

- Critère de distance : Si j et k appartiennent tous les deux à $lAContPref_i$, alors i préfère se placer au dessus de celui dont la date de départ est la plus proche postérieure à la sienne c'est-à-dire le premier partant après lui (favorisant celui qui a la même date de départ que la sienne). S'ils appartiennent tous les deux à $lAContNonPref_i$, alors i choisit celui dont la date de départ est la plus proche antérieure à la sienne. Enfin, s'ils appartiennent à deux listes différentes, i choisit celui qui appartient à $lAContPref_i$. i n'a pas intérêt à être affecté à un emplacement pour être déplacé ultérieurement, et j ou k n'ont pas intérêt à être gênés par un agent Conteneur qui sera nécessairement déplacé au départ de l'un d'eux. Bel et bien, ce critère permet de minimiser le nombre de mouvements générés tout en respectant simultanément les intérêts d'un agent Conteneur chercheur de place et d'un agent Conteneur déjà satisfait.
- Critère de hauteur : i choisit parmi j et k celui situé à un niveau plus élevé c'est-à-dire appartenant à la pile la plus chargée. Ce critère permet de maximiser l'exploitation des piles disponibles. Au lieu d'avoir des piles mi-pleines, nous aurons à long terme des piles saturées et des piles vides ce qui peut éventuellement optimiser le placement des conteneurs nouvellement arrivés.
- Heuristique de pile destination : i choisit parmi j et k celui situé dans la pile la plus proche c'est-à-dire situé à une distance minimale. Ce critère permet de minimiser la distance parcourue par les équipements de manutention pour déplacer des conteneurs.
- Critère d'ordre lexicographique des piles : ce critère est utile dans le cas où j et k seraient ex æquo suivant les critères précédents. Les agents Conteneur perturbés et les agents Conteneur nouvellement arrivés sont empilés au-dessus des agents Conteneur qui vérifient ces critères selon un ordre de priorité décroissant : d'abord le critère de distance, ensuite le critère de hauteur de pile, et enfin l'ordre lexicographique.
- L'agent Conteneur j est le plus préféré par l'agent Conteneur i s'il appartient à $lAContPref_i$ et :

$$Pref_{ij}^+ = Min_{j=1..n_1}(Dist_{ij}) \text{ où } j \neq i, n_1 \leq N$$

- Un agent Conteneur j est le moins non préféré par un agent Conteneur i s'il appartient à $LAContNonPref_i$ et :

$$Pref_{ij}^- = Max_{j=1..n_1}(-Dist_{ij}) \text{ où } j \neq i, n_1 \leq N$$

Algorithme d'optimisation A la réception des messages : JeSuisLibre (), JeSuisDésolé () ou Rejeter (), un agent Conteneur recherche un nouvel emplacement faisant appel à une fonction meilleurNouvEmp() qui implémente les heuristiques gloutonnes déjà évoquées (voir Algorithme 4.14)

Traitement des arrivées Les arrivées des agents Conteneur ayant lieu entre deux départs successifs sont traitées de la manière suivante : Une fois tous les agents Conteneur A sont partis au premier départ, l'agent interface génère de manière aléatoire des agents Conteneur arrivés à chaque départ et envoie en outre le message Gêné () à chacun. La diffusion de ce message permet à ces nouveaux agents Conteneur, initialement insatisfaits, d'être empilés identiquement à des agents Conteneur perturbés selon la méthode heuristique distribuée proposée. Ceci garantit la cohérence du modèle COSAH et contribue à la minimisation des mouvements pouvant avoir lieu au prochain départ. A partir du moment où le dernier agent Conteneur arrivé est empilé, nous relançons le départ suivant selon le même processus de résolution décrit précédemment.

Situation de blocage solvable (tolérance) En plus de la situation de blocage non solvable rencontrée dans DistNIS et explicitée dans la section 4.3.2.1, une situation de blocage solvable peut provenir dans l'exécution de DistIS lorsque toutes les têtes des piles (contenant des A) sont perturbées et les autres piles (ne contenant pas des A) sont saturées. Autrement dit, si un agent conteneur reçoit uniquement des messages JeSuisDésolé () ou bien si le nombre de chercheurs de places est égal au nombre de piles qui sont non saturées et dont les têtes sont toutes perturbées. Cela est dû aux intérêts conflictuels des différents agents Conteneur ayant chacun pour but essentiel de minimiser le nombre de ses déplacements. Dans ce cas, nous déclenchons une procédure de tolérance : un agent Conteneur, appelé agent Conteneur tolérant, même s'il est insatisfait et recherche

```

Fonction meilleurNouvEmp (i) → EmpSol
Début
1- EmpSol ← null
2- nivMax ←  $-\infty$ 
3- distMin ←  $+\infty$ 
4- Si ( $|lAContPref_i| > 0$ ) alors
5-   min ← 1111 ;
6-   Pour i de 0 à  $|lAContPref_i| - 1$  Faire
7-     EmpLibre ← i
8-     Si ( $|lAContPref_i|=1$ ) alors
9-       Vider (lAContPref_i)
10-      Si  $\neg$  (EmpLibre.nom = EmpLibre.dessus) alors
11-        retourner (EmpLibre)
12-      Si (min  $\geq$  EmpLibre.dist) alors
13-        Si (min = EmpLibre.dist) alors
14-          Si (nivMax < EmpLibre.niveau) alors
15-            nivMax ← EmpLibre.niveau
16-            EmpSol ← EmpLibre
17-            min ← EmpLibre.dist
18-          Sinon Si (nivMax = EmpLibre.niveau) alors
19-            Si (distMin  $\geq$  EmpLibre.distPile) alors
20-              distMin ← EmpLibre.distPile
21-              nivmax ← EmpLibre.niveau
22-              min ← EmpLibre.dist
23-              EmpSol ← EmpLibre
24-          Sinon
25-            min ← EmpLibre.dist
26-            nivmax ← EmpLibre.niveau
27-            distMin ← EmpLibre.distPile
28-            EmpSol ← EmpLibre
29-        Si (EmpSol  $\neq$  null) alors
30-          Supprimer (lAContPref_i , EmpSol)
31-        Retourner (EmpSol)
32-      Si ( $|lAContNonPref_i| > 0$ )
33-        max ←  $-\infty$ 
34-        nivMax ←  $-\infty$ 
35-        meilleurNouvEmp (lAContNonPref_i )
36-      Sinon
37-        retourner (null)
Fin

```

Algorithme 4.14: Algorithme d'optimisation heuristique

un nouvel emplacement, accepte temporairement des agents Conteneur au-dessus dans le but de libérer des emplacements occupés par des A et puis nous reprenons le processus d'exécution sans tolérance en déclenchant de nouveau après un certain Δ la procédure je gêne(). Les agents Conteneur tolérants sont déterminés ou encore choisis selon une heuristique de tolérance qui consiste à appliquer dans un ordre de priorité décroissante ces deux critères :

- Critère de catégorie : un agent Conteneur i est tolérant par rapport à un autre agent Conteneur j s'il est le premier partant c'est-à-dire si Cat_i est inférieure à Cat_j . Ce critère permet de minimiser le nombre de mouvements pouvant avoir lieu ultérieurement. La question qui se pose : pourquoi nous n'avons pas choisi le contraire c'est-à-dire le dernier partant. En effet, nous pensons qu'en empilant les conteneurs les moins prioritaires sur les plus prioritaires, nous pouvons aboutir à une configuration ordonnée qui facilitera leur départ, une fois les A sont partis libérant ainsi des emplacements et déclenchant de nouveau la procédure je gêne () au niveau du conteneur le plus prioritaire.
- Critère de hauteur : si i et j ont la même catégorie, alors l'agent Conteneur situé au niveau le moins élevé est l'agent Conteneur tolérant, c'est-à-dire celui qui appartient à la pile la moins élevée. Ce critère permet de libérer plus d'emplacements.
- Critère d'ordre lexicographique des piles : ce critère est utilisé dans le cas où j et k seraient ex æquo suivant les deux critères précédents.

4.4 Conclusion

Dans ce chapitre, nous avons décrit un modèle multi-agent de base baptisé ContEco que nous avons conçu pour la résolution d'une version relaxée du problème de stockage de conteneurs PSCR, en se référant à la méthode d'éco-résolution appliquée au problème classique de planification : problème du monde des cubes. Ensuite, nous avons exprimé ce modèle par une méthode formelle : les réseaux de Petri colorés, et ce uniquement en vue de donner une sémantique aux différents comportements internes des agents.

La démarche de formalisation nous a aidé à la clarification du modèle ContEco

et a montré une facilité d'appréhender les aspects d'asynchronisation et de parallélisme lors des communications entre les agents. Elle représente une " maquette " de base reflétant nos réflexions et appuyant leur faisabilité et leur développement. Cette démarche permet également au lecteur de ce manuscrit de se familiariser plus avec les concepts multi-agent et de le préparer à notre deuxième et principale contribution : le modèle COSAH, un modèle d'optimisation heuristique distribuée pour la résolution d'une version étendue du problème du stockage de conteneurs tenant compte des départs et des arrivées de conteneurs en import ou en export, ainsi que des contraintes dynamiques d'espace et de temps (espace de stockage variable selon le départ ou l'arrivée de conteneurs, fenêtres temporelles glissantes).

En effet, dans une deuxième partie de ce chapitre, nous avons détaillé ce modèle : nous avons commencé par une description de son noyau DistNIS fondé sur un algorithme distribué de recherche non informée ou encore sur un algorithme distribué de stockage aléatoire. Ensuite, nous avons misé sur le module complémentaire ISM basé sur un algorithme de recherche informée qui injecte dans DistNIS une méthode heuristique exhaustivement expliquée. Le noyau de COSAH permet d'une part, la simulation de l'activité de stockage de conteneurs dans la réalité de plusieurs ports fluviaux ou maritimes. Et d'autre part, la résolution de manière aléatoire du problème de chargement et déchargement des conteneurs en import ou en export à la fois, et ce en respectant les contraintes d'espace et de temps. Quant au module ISM, il met en jeu des heuristiques constructives gloutonnes dans le but d'optimiser le PSC en minimisant le nombre de mouvements parasites des conteneurs que ce soit pour la période courante séparant deux départs successifs ou pour les périodes suivantes.

Afin d'évaluer le modèle COSAH et précisément les heuristiques adoptées, nous proposons dans le chapitre suivant une étude empirique comparative selon divers paramètres utilisant en premier lieu des jeux de tests générés aléatoirement, et en deuxième lieu des données réelles relatives à un cas d'étude : le port maritime de Radès en Tunisie.

Chapitre 5

Validation Expérimentale

5.1 Introduction

Afin de pouvoir tester la validité de notre modèle COSAH, nous avons décidé de l'implémenter. Cette implémentation s'est déroulée en deux temps : dans un premier temps, nous avons implémenté le noyau de base de COSAH : DistNIS en créant des agents d'un seul type (agent Conteneur) en les dotant de toutes les fonctionnalités et connaissances indispensables à leur comportement tenant compte des contraintes d'espace et de temps. Nous avons voulu simuler et résoudre l'activité de stockage de conteneurs à l'image de port de Radès que nous avons pris comme exemple dans ce chapitre. Puis, dans un deuxième temps, nous avons implémenté COSAH en rajoutant le module complémentaire ISM. Notre objectif est d'optimiser le PSC et de faire du modèle COSAH une solution envisageable dans la réalité de plusieurs ports fluviaux ou maritimes. Pour ce faire, nous avons mis au point un prototype mettant en œuvre le modèle COSAH et un modèle centralisé équivalent désigné par *COSH* pour *Container Stacking via Heuristic method*. Ce dernier repose sur un algorithme centralisé basé sur une recherche informée dénommé CentIS (*Centralised Informed Search*) qui, pareillement à COSAH, se compose d'un noyau dénommé CentNIS (*Centralised Non Informed Search*) que nous avons étendu en rajoutant le module ISM *Informed Search Module*. Notons que nous avons adopté les mêmes heuristiques constructives gloutonnes définies dans DistIS qui sont appliquées aux départs des

conteneurs de catégorie A [KKGY07b]. Pour mettre en œuvre le modèle COSAH nous avons utilisé la plateforme multi-agent JADE [CSE99]. Ce chapitre s'articule autour de trois sections : la première présente succinctement notre choix de la plateforme multi-agent JADE, la deuxième expose les résultats comparatifs de nos expérimentations qui sont faites sur des instances générées aléatoirement, la troisième s'attarde sur un cas d'étude : le port de Radès, où nous utilisons des jeux de tests réels.

5.2 Implémentation

Les algorithmes que nous avons développés ont été implémentés et exécutés sur un PC portable Centrino 2GHz doté de 1,00 Go de RAM. Nous pensons toutefois qu'il serait souhaitable de l'exécuter sur une machine multiprocesseur afin de mieux exploiter les avantages d'une modélisation multi-agent, et ce essentiellement en terme d'asynchronisme et de parallélisme. Pour développer des agents, il est pratique de choisir une plateforme. A l'heure actuelle, plusieurs outils ont été développés pour la réalisation d'agents et plusieurs études comparatives ont été menées comme [GD02]. Si ces plateformes inondent en quantité, elles ne répondent pas toutes aux critères que nous nous sommes fixés. Le paragraphe suivant justifie notre choix de la plateforme JADE pour le développement de nos agents Conteneur.

5.2.1 Critères de choix

Ces critères sont naturellement en relation avec nos attentes d'un système multi-agent et des caractéristiques des agents. Ainsi, la plate forme devra répondre aux critères suivants :

- Les agents doivent pouvoir communiquer entre eux. Cette communication n'est pas un simple échange de messages mais un dialogue . Il ne s'agira pas pour un agent de répondre mécaniquement à un message mais de pouvoir analyser plusieurs possibilités de réponses avant de réagir.
- La plateforme doit être répandue, elle doit avoir répondu à plusieurs projets de développement de systèmes multi-agent.

- Le langage de programmation sous jacent et la nature des messages échangés est aussi un point important. Un langage plus répandu favorisera une meilleure compréhension des codes sources.

En définitif, JADE nous semble la plateforme qui se rapproche plus de nos critères. C'est une plateforme de création d'agents qui prend en compte les spécifications de la FIPA pour l'interopérabilité des systèmes multi-agent. Le concept d'agent est vu par JADE comme un processus autonome et indépendant qui a une identité, et qui requiert la communication (collaboration, compétition, coopération, etc.) avec les autres agents dans le but de remplir ses missions.

5.2.2 Présentation de JADE

La plateforme *JADE* (*Java Agent DEvelopment Framework*) est un environnement de développement d'agents implanté totalement dans le langage JAVA. Il facilite la mise en place d'un système multi agent répondant aux spécifications de *FIPA* (*Foundation for Intelligent Physical Agent*) à travers un ensemble d'outils. La plateforme *JADE* inclut tous les composants obligatoires qui contrôlent un système multi-agent. Ces composants sont l'*ACC* (*Agent Communication Channel*), l'*AMS* (*Agent Management System*) et le *DF* (*Directory Facilitator*). Toute la communication entre agents est exécutée par messages *FIPA/ACL*. La plateforme d'agent peut être répartie sur plusieurs serveurs. Une seule application Java, et donc une seule machine virtuelle Java *Java Virtual Machine-JVM*, est exécutée sur chaque serveur. Chaque *JVM* est un conteneur d'agents qui fournit un environnement complet pour l'exécution d'agent et permet à plusieurs agents de s'exécuter en parallèle sur le même serveur. L'architecture de communication offre la transmission de messages flexibles et efficaces. *JADE* crée et contrôle une file d'attente des messages entrants pour chaque agent. Le modèle global de communication entre ces agents *FIPA* a été mis en application. Ses composants ont été distingués clairement et ont été entièrement intégrés : protocoles d'interaction, *ACL*, langues, schémas de codage, protocoles de transport, etc. A chaque agent on associe un processus léger *thread* mais les agents ont souvent besoin d'exécuter des tâches en parallèle. L'environnement de développement pourvoit aussi des comportements tout " prêt à exécution " pour les tâches les plus souvent

sollicitées dans la programmation orientée agent.

5.3 Expérimentations

Nous avons présenté l'outil qui va nous permettre d'implémenter et d'évaluer le modèle COSAH. Il nous reste donc à présenter et analyser les résultats numériques inhérents. Il nous faut cependant décrire les instances qui composent notre jeu de tests. En effet, nous avons utilisé, en premier lieu, des instances générées aléatoirement, et en deuxième lieu un jeu de tests réels relatifs au port de Radès en Tunisie.

5.3.1 Instances

Nous nous sommes basés sur des instances générées aléatoirement, chaque instance représente une configuration ou encore un arrangement initial de conteneurs. Ces instances possèdent les caractéristiques suivantes : nombre de conteneurs, nombre de piles de conteneurs, la hauteur maximale, et les catégories des conteneurs ou encore leur dates de départ estimées. Notons que la hauteur maximale est générée aléatoirement dans l'intervalle $\{1, \dots, 6\}$, puisque la hauteur maximale dans les plus grands ports mondiaux ne dépasse pas six. L'horizon de notre travail s'étale sur six départs, d'où la catégorie des conteneurs varie dans l'intervalle $\{A, \dots, F\}$. Cependant, nous pouvons intégrer des conteneurs ayant des dates de départs ultérieures.

5.3.2 Résultats numériques

Dans cette section, nous présentons les résultats numériques des expérimentations de COSAH. Ces expérimentations se subdivisent en deux classes : les premières sont inter-modèles comparant COSAH à COSH, et les deuxièmes sont intra-modèle comparant COSAH au nombre maximal de mouvements pouvant être atteint, et illustrant la sensibilité du modèle COSAH à quelques paramètres du problème : le nombre de mouvements maximal et la hauteur moyenne des piles. Par la suite, ces résultats sont détaillés à l'aide de graphiques commentés.

Notons que ces résultats numériques correspondent à la moyenne de 10 exécutions pour chaque instance.

5.3.2.1 Comparaisons inter-modèles

Les graphiques suivants illustrent une étude comparative de COSAH par rapport à COSH afin de montrer, d'une part, l'impact de la distribution, et d'autre part, l'effet de l'utilisation de la méthode Heuristique pour la résolution et l'optimisation du problème étudié : le stockage de conteneurs dans un port-PSC.

Distribution VS Centralisation La figure 5.1 montre une comparaison entre DistNIS le noyau de base de COSAH et CentNIS celui de COSH, tous deux basés sur une méthode de recherche non informée. Ce graphique montre que la résolution distribuée et centralisée du PSC en procédant à une méthode de recherche non informée simulent toutes les deux de la même manière une activité de stockage aléatoire des conteneurs dans un port, et génèrent des solutions voisines pour la plupart des instances. Dans sa globalité, ces résultats proches nous

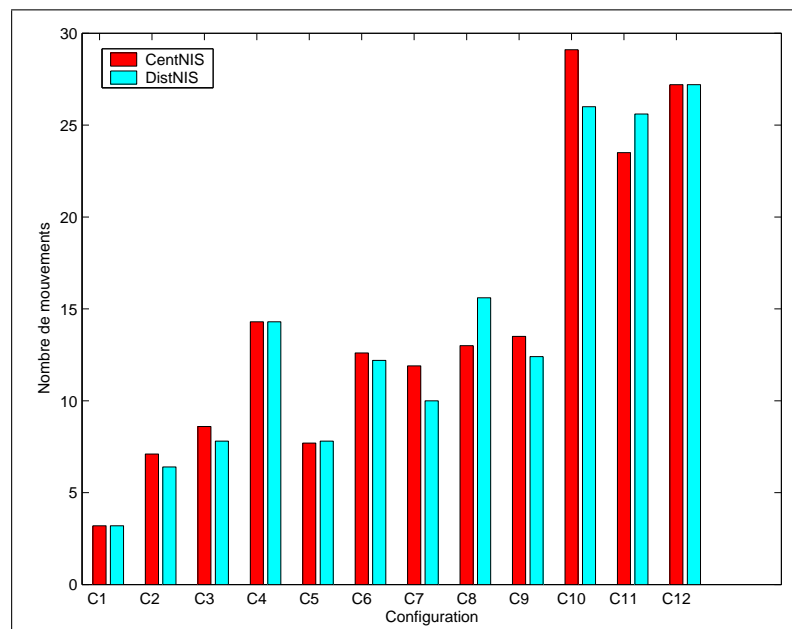


FIG. 5.1 – Comparaison DistNIS par rapport à CentNIS

conduisent à ne plus faire de distinction entre CentNIS et DistNIS et à ne parler que d'un noyau simulant un processus aléatoire de stockage de conteneurs. Mais, comme nous pouvons le constater, 50% de l'ensemble des configurations considérées montre que CentNIS est bien moins performant que DistNIS. Nous pouvons voir que DistNIS se comporte bien car au bout d'un certain temps, d'une exécution à une autre, nos agents Conteneur empruntent les mêmes chemins et poursuivent un même schéma conversationnel, ce qui mène à des résultats oscillant dans un même intervalle étroit de valeurs.

La figure 5.2 montre une comparaison entre COSAH et COSH, d'autre manière entre DistIS et la version centralisée CentIS, toutes deux basées sur une méthode de recherche informée qui est la méthode heuristique. Ce graphique montre une forte variation entre DistIS et CentIS, nous pouvons voir que COSAH est beaucoup plus performant que COSH sur 75% des configurations considérées. Pour les autres configurations (C1, C7, C10), les résultats sont confondues. Par rapport aux résultats illustrés dans la figure 5.1, où la distribution seule n'est pas d'un effet considérable sur la qualité de la solution, nous pouvons percevoir clai-

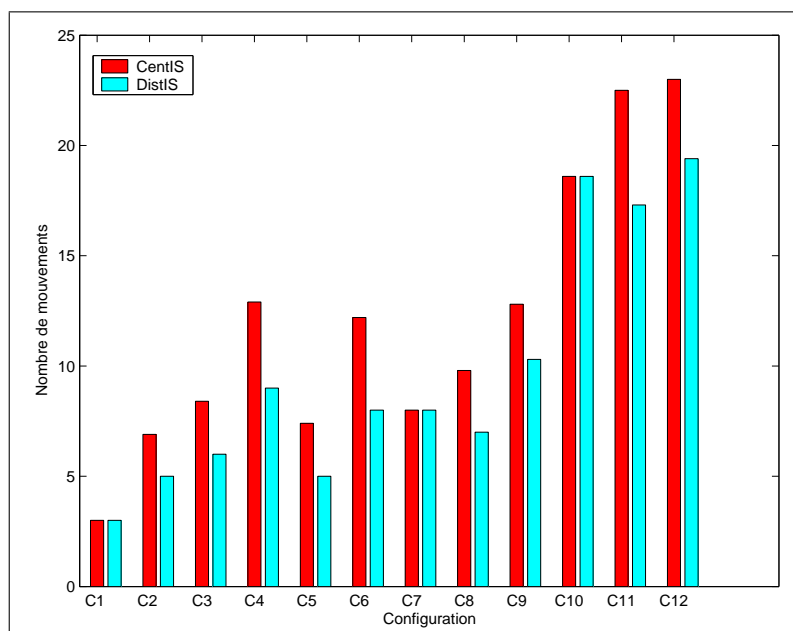


FIG. 5.2 – Comparaison de DistIS par rapport à CentIS

rement dans cette figure l'impact du couplage de la distribution et du guidage ou encore la combinaison des deux concepts Agent et Heuristique sur la qualité des solutions produites. La performance de COSAH montre bien l'efficacité d'une méthode d'optimisation heuristique distribuée par rapport aux méthodes heuristiques centralisées classiques.

Avec Heuristiques VS Sans Heuristiques La figure 5.3 montre une comparaison entre CentIS et CentNIS. Cette comparaison illustre bien l'efficacité de l'utilisation d'une méthode heuristique dans la minimisation du nombre de mouvements pour le problème du stockage des conteneurs. D'un point de vue général, ces résultats prouvent encore la performance des heuristiques constructives gloutonnes spécifiques dans la résolution des problèmes d'optimisation combinatoire tel est le cas du problème sous-jacent.

La figure 5.4 montre une comparaison DistNIS (*Distributed Non Informed Search*) et DistIS (*Distributed Informed Search*). Cette comparaison montre que l'introduction d'un degré d'intelligence via un algorithme de recherche informée

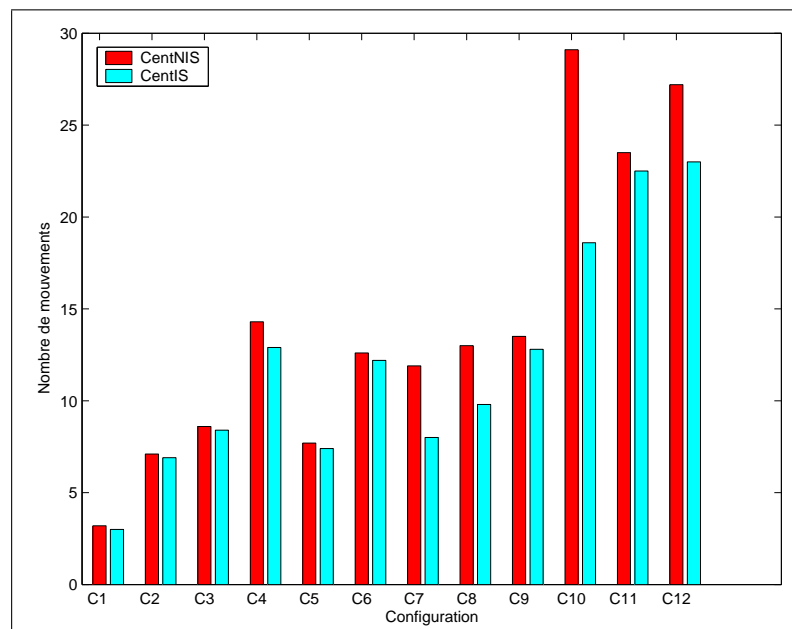


FIG. 5.3 – Comparaison de CentIS par rapport à CentNIS

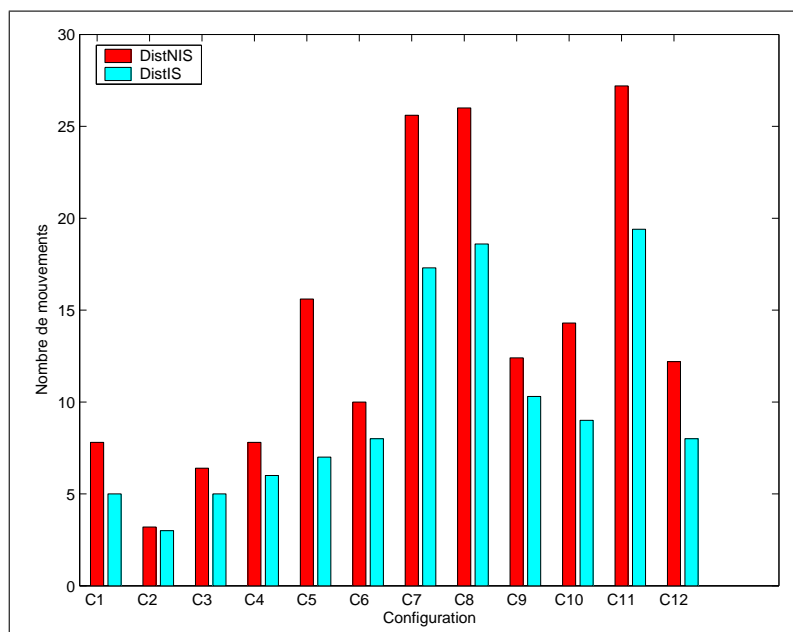


FIG. 5.4 – Comparaison de DistIS par rapport à DistNIS

dans le comportement des agents Conteneur impliqués dans COSAH conduit à une réduction significative du nombre total des mouvements parasites. Cette étude comparative inter-modèles montre que l'introduction d'un degré d'intelligence dans le comportement des agent Conteneurs impliqués dans COSAH en faisant appel aux heuristiques constructives gloutonnes, conduit à des résultats nettement meilleurs que ceux du modèle distribué avec une recherche non informée, qui à son tour améliore mais sensiblement la version centralisée avec une recherche non informée.

5.3.2.2 Comparaisons intra-modèle

L'étude comparative inter-modèles décrite dans la section précédente a montré que notre modèle COSAH est particulièrement adapté pour la résolution du problème étudié PSC. En conséquence, nous nous concentrons dans cette section sur une étude comparative illustrant la sensibilité de COSAH à différents paramètres : la borne maximale du nombre de mouvements et la hauteur moyenne des piles.

Comparaison de COSAH par rapport à la borne maximale La courbe de la figure 5.5 représente la variation du rapport entre le nombre de mouvements produit par COSAH et la borne maximale pouvant être atteinte, en fonction du nombre de conteneurs. Nous pouvons voir que ce rapport noté $NbrMvt_{COSAH} / NbrMvt_{max}$ ne dépasse pas 55%, ce qui appuie l'efficacité de notre solution proposée. La borne maximale correspond au nombre total de mouvements pouvant être atteint au départ courant ainsi qu'aux prochains départs en supposant qu'il n'y aura pas d'arrivées et qu'un conteneur qui gêne un autre conteneur dans le départ courant continue à le gêner jusqu'à son départ.

La borne maximale dénotée $NbrMvt_{max}$ est calculée en utilisant la formule mathématique 5.1, étant donnée une configuration initiale $Config_{init}$:

$$\sum_{p=0}^{NP-1} \sum_{i=0}^{h_p-1} \sum_{j=i+1}^k x_{ij} c_{ij} \quad (5.1)$$

Où :

NP : le nombre de piles.

h_p : la hauteur d'une pile p ($h_p \leq H_{max}$).

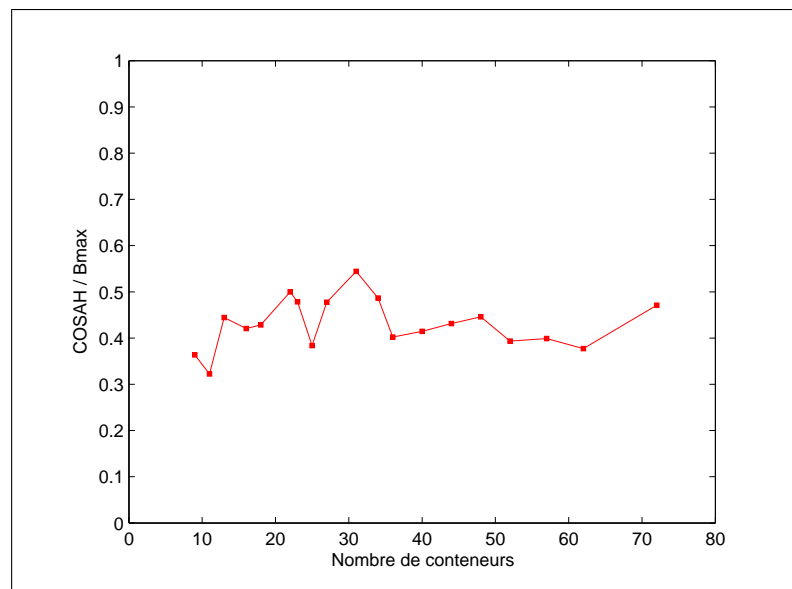


FIG. 5.5 – Mesure de performance de COSAH par rapport à la borne maximale

$$c_{ij} = cat_j - cat_i$$

$$x_{ij} = \begin{cases} 1 & \text{Si le conteneur } j \text{ est de catégorie strictement supérieure à celle de } i; \\ 0 & \text{Sinon.} \end{cases}$$

Étant donné une pile p et un conteneur i (c'est-à-dire situé à l' $i^{\text{ème}}$ niveau), nous appliquons la formule 5.1 à partir du conteneur au-dessus de i ($j = i + 1$) jusqu'à ce qu'un conteneur k de catégorie inférieure à celle i soit rencontré (k doit être traité aussi). Dans ce cas, nous poursuivons le calcul du nombre de mouvements en réappliquant la même formule mais à partir du conteneur k .

Comparaison de COSAH et de COSH en fonction de la hauteur moyenne

La figure 5.7 montre la variation du nombre de mouvements en fonction de la hauteur moyenne (la hauteur maximale est fixée à 6 et le nombre de piles à 10). Ce paramètre est retenu après avoir testé la sensibilité du nombre de mouvements à plusieurs paramètres (nombre de conteneurs, taux d'occupation, nombre de catégories, nombre de conteneurs de chaque catégorie, nombre initial de conteneurs de catégorie A, nombre de conteneurs prioritaires à chaque départ) et leur

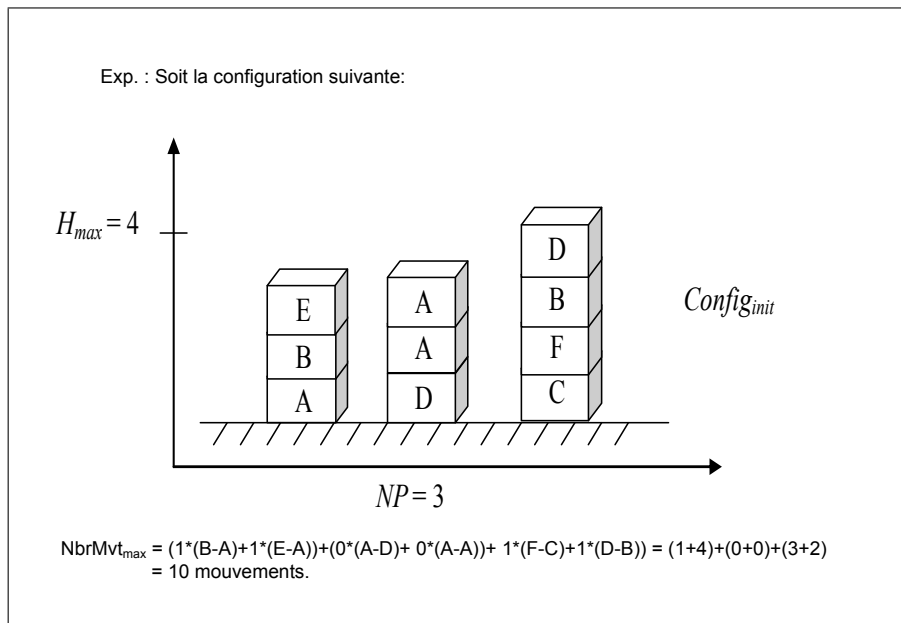


FIG. 5.6 – Exemple de calcul de la borne maximale du nombre de mouvements

combinaisons pouvant caractériser et déterminer une configuration donnée. Nous constatons que le nombre de mouvements est généralement proportionnel à la hauteur moyenne. L'existence d'un nombre initial élevé de conteneurs de catégorie A ou bien de places vides expliquent la diminution du nombre de mouvements pour les instances 6 et 8 ($\#(32, 3.2)$ et $\#(44, 4.4)$), et les valeurs voisines du nombre de mouvements pour DistNIS et DistIS pour les trois premières instances ($\#(6, 0.6)$, $\#(11, 1.1)$, $\#(16, 1.6)$) et la huitième instance ($\#(44, 4.4)$).

5.4 Cas d'étude : Port de Radès

Cette section est consacrée à une application qui met en oeuvre le modèle COSAH et son utilisation pour la simulation et la résolution du problème de stockage de conteneurs PSC dans le port maritime Tunisien : Port de Radès. Nous commençons d'abord par présenter succinctement le port de Radès et ses activités. Ensuite, nous montrons des expérimentations qui ont permis d'approuver les performances attendues de notre solution dans la réalité.

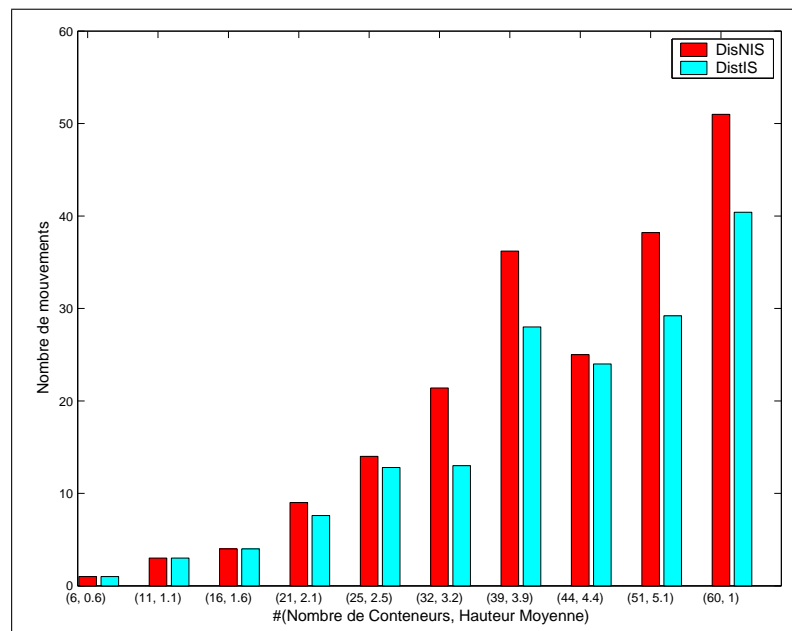


FIG. 5.7 – Sensibilité de DistIS et de DistNIS à la hauteur moyenne

5.4.1 Présentation du port

En dépit de l'instabilité de la situation économique mondiale, l'économie Tunisienne a continué à se développer surtout dans les secteurs du transport, de l'industrie et du tourisme. Le commerce extérieur a connu, pour sa part, une nette évolution des importations et des exportations. Toutes ces circonstances ont donné à l'économie nationale une dynamique lui permettant de s'intégrer dans l'économie mondiale, ainsi que la capacité de s'adapter aux mutations nationales et internationales. Le volume global des échanges dans tous les ports de commerce tunisiens a atteint 28,5 millions de tonnes d'après des études statistiques de l'an 2005 [STA05]. Ces millions de tonnes sont répartis en des marchandises diverses non unitisées embarquées et débarquées et des marchandises unitisées. Ces dernières représentent 56,67 % du volume global des marchandises en 2005 [STA05] répartis entre conteneurs et unités roulantes (semi-remorques, remorques et voitures). En 2005, le trafic des conteneurs transitant par les ports de commerce tunisiens a atteint 223 mille unités (environ 340 mille EVP) contre 211 mille unités en 2004 (environ 310 mille EVP) enregistrant une hausse de 10%. Aussi le taux de conteneurisation des marchandises dans l'ensemble des ports tunisiens a atteint 41% en 2005 contre 39% en 2004. Il importe de signaler, que cette activité est centralisée principalement au port de Radès et aux ports de la goulette et de Sfax. Le port de Radès, quant à lui, possède un taux de conteneurisation qui s'est élevé à 64% et manipule 92% du volume global de conteneurs (environ 316 mille EVP en 2005) [STA05, OMM05]. Cette hausse est concrétisée grâce à des programmes et des projets d'investissement mis en place au niveau de l'Office de la Marine Marchande et des Ports-OMMP et de la Société Tunisienne d'Acconage et de Manutention-STAM. Ces deux établissements ont procédé à la consolidation d'équipements par l'acquisition de nouveaux matériels de manutention (cavaliers gerbeurs, chariots élévateurs, grues pneumatiques, *Roll Trailer*, *Ro-Ro Truck*, bennes $5m^3$, enrouleurs électriques et des mini-chargeuses). Le port de Radès est composé d'une zone d'import et une zone d'export. Chaque zone est en elle même composée de deux sous-zones : une destinée au stockage des conteneurs pleins et une autre dédiée au stockage des conteneurs vides. La logistique utilisée actuellement au port de Radès pour les opérations de chargement / dé-

chargement (ou embarquement / débarquement) des conteneurs en export ou en import consiste à affecter des engins (*Reach-Stacker*, élévateurs, *Ro-Ro Trucks*, des *Ro-Ro Trailers* et des cavaliers gerbeurs). Les *Ro-Ro Trucks* avec les *Ro-Ro Trailers* sont utilisés pour assurer le transfert des conteneurs transportés vers les zones de stockage. Les cavaliers gerbeurs avec les *Reach-Stackers* et les chariots élévateurs pour assurer le stockage (gerbage ou enlèvement) des conteneurs. Le port de Radès dispose de 7 postes à quais, 4 grues pneumatiques, 22 cavaliers gerbeurs, 10 *Reach-Stacker* et 34 *Roll-Truck*. Au milieu de tout ça, nous nous intéressons à l'activité de stockage de conteneurs (débarquement/embarquement) détenue par la STAM, devenue en 2005 le concessionnaire du port de Radès. Dans ce port, l'opération de stockage de conteneurs dépend du fait qu'ils soient en import ou en export. On peut avoir soit des conteneurs renfermant une marchandise dangereuse, donc de plus haute priorité qui sont cheminés directement vers leur destination, soit des conteneurs vides qui sont toujours destinés à l'export, soit des conteneurs frigorifiques qui doivent être rapidement affectés à des prises de frigo. Ils sont mis à quais en observant une procédure d'autorisation tenant compte de l'avancement des opérations commerciales du navire, de l'espace disponible, etc. Ils sont stockés dans un même lot dans la première zone de stockage qui se présente. L'opération d'embarquement s'effectue par le rapprochement des conteneurs mis à quai par des élévateurs équipés avec des *spreaders* frontaux. En ce qui concerne les conteneurs en export, on connaît la date de leur départ puisque la date d'arrivée du navire associé est connue à l'avance. Afin de faciliter leur départ, les conteneurs ayant la même date de départ sont stockés dans des zones de stockage, dédiée chacune à un navire bien déterminé. Cette stratégie de zoning permet effectivement de minimiser le nombre de mouvements au départ ou à l'arrivée de conteneurs en export, mais ça conduit à une mal exploitation de l'espace de stockage disponible. Quant aux conteneurs en import, ceux ci arrivent par navire, puis sont déchargés sur le quai, ensuite transférés vers une zone de stockage. Une fois les formalités douanières sont achevées, ces conteneurs peuvent être soit enlevés par camion ou train dans le cas d'un contrôle positif, ou bien retournés à la zone de stockage dans le cas contraire. Il est à noter que les dates de départs de ces conteneurs sont inconnues, puisque les camions (clients associés) arrivent un à un de manière inattendue. Le déchargement des navires se fait à

l'inverse de leur plan de chargement, on utilise l'heuristique de la destination la plus proche du navire.

Le stockage des conteneurs en import se fait de manière aléatoire par les cavaliers gerbeurs. Cette stratégie accroît indubitablement le nombre de mouvements attendus. Afin de réduire ce nombre, une méthode d'exploitation est nouvellement instaurée au port de Radès qui se base sur des investissements en équipements et qui consiste principalement à affecter deux cavaliers gerbeurs dans chaque zone de stockage pour assurer les opérations de débarquement et d'embarquement. Cette méthode permet d'améliorer le rendement de ces opérations commerciales. Mais elle présente l'inconvénient d'être dépendante au nombre d'engins disponibles. En ce qui concerne le repérage des conteneurs, il est effectué soit par des pointeurs soit via un système de gestion informatisé basé sur une saisie manuelle de 4 coordonnées : numéro de la zone de stockage, numéro de la travée ou colonne (pile), numéro de la ligne (rangée), et le niveau de gerbage (hauteur). Suite à cette étude de l'existant du port de Radès, nous avons décelé des points défailants qui se résument essentiellement en :

- Une politique de placement consiste à adopter une hauteur maximale égale à 2.5 dite encore politique 3/2 (c'est-à-dire chaque pile de hauteur maximale égale à 3 doit être suivie par une pile adjacente de hauteur maximale égale à deux). De ce fait, si on veut enlever un conteneur d'une pile donnée, on aura au moins un emplacement disponible dans une pile de hauteur 2. Ceci permet de garder une marge de manœuvre et de faciliter la manutention des conteneurs déplacés ou nouvellement arrivés, sans avoir un souci de minimisation du nombre de mouvements. En outre, cette politique pourra entraîner une mal exploitation de la zone de stockage vu que le taux d'occupation maximal vaut environ 80%.
- Une solution très limitée pour la planification de l'activité de stockage de conteneurs qui se fait de manière aléatoire.
- Une solution primitive (semi-automatique) pour le repérage et pointage des conteneurs.

A notre égard, ces quatre points représentent des pâmoisons dans le système de gestion du port de Radès actuel et constituent un terrain favorable pour mettre en œuvre et tester notre modèle COSAH tenant compte de la plupart des paramètres

réels du port (nombre de conteneurs, fonctionnement des engins disponibles, résistance du sol, retour de conteneurs, plusieurs arrivées simultanées, nature de conteneurs et hauteur des piles).

La section suivante présente les expérimentations testant la validité de COSAH selon un jeu de tests extrait de la réalité du port de Radès.

5.4.2 Expérimentations

Après l'étude de l'existant au port de Radès, nous avons constaté que notre algorithme DistNIS représentant le noyau de notre modèle proposé COSAH reflète bien sa réalité et décrit la politique actuelle adoptée pour le stockage des conteneurs. Compte tenu des comparaisons faites entre DistNIS et DistIS, des difficultés que peut induire cette politique et de quelques paramètres réels du port, nous avons décidé de tester la performance de COSAH par rapport à la politique actuelle en terme de nombre de mouvements selon un jeu d'essai constitué de données réelles. Ce jeu d'essai comporte 9 instances, définie chacune par un nombre de conteneurs et un taux d'occupation. La hauteur maximale vaut 2.5 tel que nous l'avons indiqué, et le nombre de piles est égal à 10 dans une zone de stockage.

Comme le montre la figure 5.8, COSAH minimise bien le nombre de mouvements par rapport à la politique déployée actuellement. Ces résultats, fortement encourageants, montre que COSAH est une solution envisageable pour optimiser la réalité de l'activité de stockage au port de Radès. Notons que la non proportionnalité entre le nombre de mouvements et le nombre de conteneurs ou bien le taux d'occupation est due à d'autres caractéristiques des instances considérées comme le nombre de catégories des conteneurs, le nombre de conteneurs de chaque catégorie, le nombre de places vides, le nombre initial de conteneurs de catégorie A. La variation de ces caractéristiques peuvent aussi expliquer les valeurs proches du nombre de mouvements pour COSAH et la politique actuelle du port pour les instances ($\#(10, 0.23)$ et $\#(12, 0.33)$).

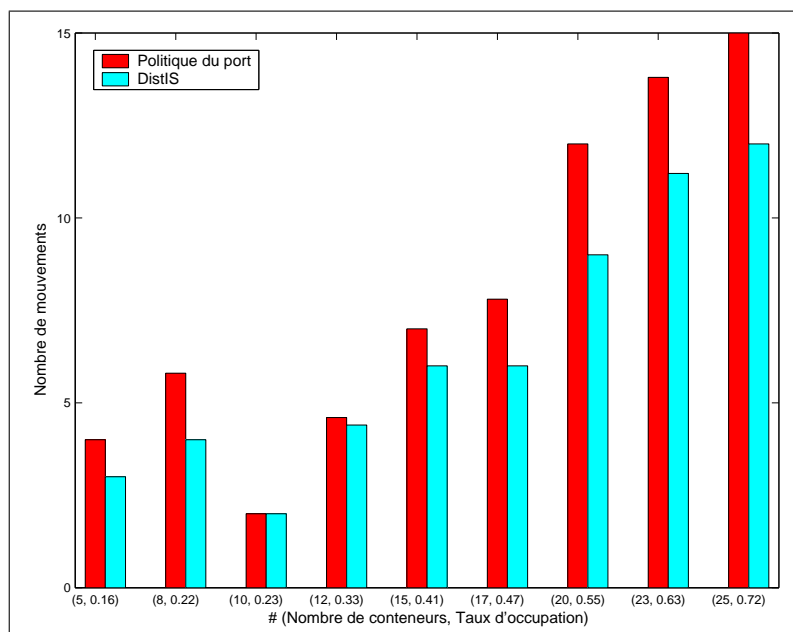


FIG. 5.8 – Mesure de performance de COSAH par rapport à la politique actuelle du port

5.5 Conclusion

Nous avons présenté d'abord notre plate-forme multi-agent qui nous a permis une évaluation comparative de notre solution basée sur une méthode heuristique distribuée et présentée par le modèle COSAH. Par la suite, nous avons implémenté ce modèle ainsi que la version centralisée associée COSH, et élaboré une étude comparative inter-modèles et intra-COSAH. Enfin, nous avons appliqué COSAH à un cas d'étude réel tunisien : le port de Radès. Nous avons montré d'une part, que le noyau DistNIS de notre modèle proposé COSAH simule bien la réalité du port de Radès. Et d'autre part, que l'algorithme DistIS (DistNIS et ISM) sur lequel s'appuie COSAH résout et optimise l'activité de stockage de conteneurs. Les résultats expérimentaux ont permis d'entrevoir l'efficacité de COSAH en particulier et d'une méthode d'optimisation heuristique distribuée en général, pour résoudre un problème particulier d'optimisation combinatoire : le PSC.

Conclusion Générale

Contributions

A l'origine de cette thèse est un intérêt pour les problèmes de transport maritime ou fluvial conteneurisé, liés à la gestion des quais dans des terminaux à conteneurs. La réduction des coûts associés à ce transport représente une préoccupation de plusieurs dirigeants d'entreprises et fait l'objet de plusieurs travaux de recherche. Nous nous intéressons au problème de stockage de conteneurs PSC qui constitue un problème complexe et ouvert s'inscrivant dans le cadre des problèmes d'optimisation combinatoire. Il s'agit de réduire les temps morts en minimisant le nombre de mouvements pouvant se produire lors de l'opération de chargement tout en respectant des contraintes d'espace et de temps. Il nous est alors nécessaire de proposer un modèle qui résout et optimise le placement de conteneurs à leur départs et arrivées. Notre travail a donné d'ores et déjà lieu à des contributions tant au niveau théorique que pratique.

Sur le plan théorique :

Notre première contribution était de proposer un modèle de base d'éco-résolution dénommé ContEco pour la simulation et la résolution d'une version relaxée du problème soulevé : RCSP. Au cours de cela, nous avons utilisé la méthode formelle réseaux de Petri colorés pour formaliser la conversation entre deux agents distincts. les modèles RdPc obtenus, simulés via l'outil CPN-tools, nous étaient utiles dans le sens où ils nous ont permis de capturer la dynamique de ce modèle de base.

Notre deuxième et principale contribution dans cette thèse était de développer un nouveau modèle paramétrable basé sur une méthode d'optimisation heuristique distribuée, appliqué au PSC. Ce modèle est appelé : COSAH (Container

Stacking via multi-Agent approach and Heuristic method).

Le modèle COSAH est constitué d'un noyau DistNIS basé sur un algorithme distribué de recherche non informée qui simule bien la réalité de quelques ports, et d'un module complémentaire ISM mettant en œuvre une méthode heuristique.

La dynamique de COSAH est la suivante : des agents Conteneur associés aux conteneurs manipulés coopèrent, coordonnent et négocient entre eux au sein d'un processus d'optimisation régi par des heuristiques. COSAH a pour but de fournir un plan de placement ou déplacement des conteneurs lors de départs de certains d'entre eux ou l'arrivée d'autres. Ce plan doit minimiser le nombre de mouvements pouvant avoir lieu, tenant compte des contraintes d'espace et de temps. Les performances de COSAH sont ensuite évaluées, dans un premier temps, sur des instances générées aléatoirement. Et, dans un deuxième temps, sur un jeu d'essai comportant 9 instances extraites de la réalité du port maritime tunisien : Le port de Radès. Nous avons procédé à des comparaisons inter-modèles et intra-modèle. Nous avons tiré des résultats satisfaisants montrant l'efficacité de la distribution via les systèmes multi-agent dans la simulation et résolution de problème complexe et ouvert aussi que le notre, et appuyant l'efficience des heuristiques dans l'optimisation des problèmes fortement combinatoires complexes et dynamiques. De plus, ces résultats ont permis d'entrevoir l'efficacité de notre modèle COSAH en particulier, et d'une méthode d'optimisation heuristique distribuée alliant les deux concepts : Agent et Heuristique, en général. En résumé, cette étude expérimentale nous a montré que le modèle COSAH répond bien à deux objectifs essentiels que nous nous sommes fixés :

- L'optimisation du placement des conteneurs dans la zone de stockage (statique ou dynamique selon les départs ou les arrivées de navires / camions ou de conteneurs en import ou en export) en minimisant le nombre de mouvements improductifs respectant un nombre de contraintes spatio-temporelles et tenant compte de plusieurs paramètres pouvant distinguer les conteneurs et définir une certaine priorité entre eux (destination, taille, date de départ, horizon des arrivées de camions, trains, barges ou navires associés).
- La Réorganisation du quai : Fournir une aide à la décision quant à l'extension du port, l'exploitation maximale du volume utile de stockage, choix du matériel, etc.

Sur le plan pratique, nous avons appliqué COSAH à un cas d'étude réel, celui du Port de Radès. Ce qui nous a permis d'attester que le noyau DistNIS reflète bien la réalité de ce port, et que COSAH s'avère une solution appropriée surtout dans le cas des conteneurs en import. Cette solution présente l'avantage d'être extensible à des contextes d'usage à la fois variables (port maritime ou fluvial, arrivées ou départs de conteneurs, conteneurs en export ou en import), et imprévisibles (retards, conteneurs vides, retour de conteneurs). Notre apport pratique se traduit aussi par la mise au point d'un prototype qui implémente notre modèle COSAH de la manière la plus autonome possible représentant l'évolution la plus optimale de l'activité de stockage de conteneurs dans un port fluvial ou maritime. Nous nous basons pour cela sur une plate-forme multi-agent existante (JADE).

Perspectives

A la relecture de ce travail, nous ne pouvons nous empêcher de livrer au lecteur quelques perspectives de recherche que voici :

- Étendre COSAH de manière à pouvoir supporter d'autres paramètres et traiter des imprévus (conteneurs vides, retard de conteneurs).
- Appliquer des heuristiques ou des métaheuristiques sur l'ensemble des solutions obtenues associées à une même instance.
- Lancer des méthodes d'optimisation telles que les heuristiques de recherche locale sur les différents plans partiels de déplacement des agents Conteneurs.
- Adopter le modèle COSAH pour la résolution et l'optimisation du problème d'arrimage de conteneurs dans un navire *Stowage planning problem*.
- Généralisation du modèle COSAH pour la famille des problèmes issus du monde des cubes.

Bibliographie

- [AMS02] D. Ambrosino, M. E. Marina, and A. Sciomachen. Decision rules for the yard storage management. Technical report, University of Genova, 2002.
- [APS00] M. Avriel, M. Penn, and N. Shpirer. Container ship stowage problem : complexity and connection to the coloring of circle graphs. *Discrete Applied Mathematics*, 103 :271–279, 2000.
- [BD97] O. Boissier and Y. Demazeau. Une architecture multi-agents pour des systèmes de vision ouverts et décentralisés. *Technique et Science Informatiques*, 16(18) :1039–1062, 1997. Numéro spécial sur l’Intelligence Artificielle Distribuée et des Systèmes Multi-Agents.
- [BFG05] I. Boudali, W. Fki, and K. Ghedira. An interactive distributed approach for the vrp with time windows. *International Journal of Simulation*, 6(10) :48–59, 2005.
- [BG88] A.H. Bond and L. Gasser. *Readings in Distributed Artificial Intelligence*. San Mateo CA : Morgan-Kaufmann Publishers, 1988.
- [BGAB83] L. Bodin, B. Golden, A. Assad, and M. Ball. Routing and scheduling of vehicles and crews. *Computers & Operations Research*, 10 :63–211, 1983.
- [Bis03] E.K. Bish. A multiple-crane-constrained scheduling problem in a container terminal. *European Journal of Operational Research*, 144 :83–107, 2003.

- [BKPB00] I. Bakam, F. Kordon, C. Le Page, and F. Bousquet. Formalisation de modèles multi-agents par les réseaux de petri : Application à l'étude d'un système de gestion de la chasse à l'est-cameroun. In *5ème colloque Africain sur la Recherche en Informatique*, Madagascar, 2000.
- [BRSV00] J. Böse, T. Reiners, D. Steenken, and S. Voss. Vehicle dispatching at seaport container terminals using evolutionary algorithms. In *Proceedings of the 33rd Hawaii International Conference on Systems Sciences*, 2000.
- [BZC99] L. Breton, J-D. Zucker, and E. Clément. Une approche multi-agents pour la résolution d'équations en physique des milieux granulaires. In Hermès, editor, *Actes des JFIADSMA '99*, pages 281–294, Paris, France, 1999.
- [CCH00] C-Y. Chen, S-L. Chao, and T-W. Hsieh. A time-space network model for the space resource allocation problem in container marine transportation. Technical report, National Cheng Kung University Taiwan, 2000.
- [CFLR04] P. Chen, Z. Fu, A. Lim, and B. Rodrigues. Port yard storage optimization. *IEEE Transactions on Automation Science and Engineering*, 1(1), July 2004.
- [CMS88] S. Cammarata, D. McArthur, and R. Steeb. *Readings in Distributed Artificial Intelligence*, chapter Strategies of cooperation in distributed problem solving, pages 102–105. Morgan Kaufmann, 1988.
- [CPN] CPN. Cpn-tools. Bibliographie en ligne, disponible à l'adresse <http://www.daimi.au.dk/CPNtools> 2003.
- [CRM88] Y.G. Chung, S.U. Randhawa, and E.D. McDowell. A simulation analysis for a transtainer-based container handling facility. *Computers & Industrial Engineering*, 14(2) :113–125, 1988.
- [CSE99] CSELT. Jade (java agent development framework). Bibliographie en ligne, Groupe de recherche de Gruppo Telecom, Italie, disponible à l'adresse <http://www.jade.tilab.com>, 1999.

- [CSN⁺03] Y. Cheng, H. Sen, K. Natarajan, C. Teo, and K. Tan. Dispatching automated guided vehicles in a container terminal. Technical report, National University of Singapore, 2003.
- [Dag89] C.F. Daganzo. The crane scheduling problem. *Transportation Research-B*, 23(3) :159–175, 1989.
- [dCD93] B. de Castilho and C.F. Daganzo. Handling strategies for import containers at marine terminals. *Transportation Research-B*, 27 :151–166, 1993.
- [DD91] A. Drogoul and C. Dubreuil. Eco-problem-solving model :results of the n-puzzle. In Elsevier North Holland, editor, *Decentralized A.I. 3, Actes du Workshop MAAMAW'91*, pages 283–295, Amsterdam, 1991.
- [Dem95] Y. Demazeau. From interactions to collective behaviour in agent-based systems. In *Proceedings of the First European conference on cognitive science*, pages 117–132, Saint Malo, France, 1995.
- [DEO01] M.B. Duinkerken, J.J.M. Evers, and J. A Ottjes. A simulation model for integrating quay transport and stacking policies automated containers terminals. In *Proceedings of the 15th European Simulation Multi-conference*, June 2001.
- [DLC89] Edmund H. Durfee, Victor R. Lesser, and Daniel D. Corkill. Trends in cooperative distributed problem solving. *IEEE Transactions on Knowledge and Data Engineering*, 1(1) :63–83, 1989.
- [DLL⁺98] E. K. Demir, T-Y. Leong, C-L. Li, J. W. C. Ng, and D. Simchi-Levi. Locating containers in a mega terminal. Technical report, Dept. Of IE & MS, Northwestern University/ PSA Corporation Limited/ John M. Olin School of Business, Washington University, 1998.
- [dM00] J.R. Van der Meer. *Operational control of internal transport*. PhD thesis, Erasmus University, Rotterdam, 2000.

- [DO98] Y. Demazeau and M. Occello. Software systems development as societies of agents. In *15th IFIP Conference (IT&KNOWS'98)*, pages 303–304, Vienna, September 1998.
- [Dro00] A. Drogoul. *Systèmes multi-agents situés*. Dossier d'habilitation à diriger des recherches, Université Pierre et Marie Curie, 2000.
- [DVNM99] R. Dekker, P. Voogd, L. Nagy, and P. Meersmans. Famasnewcon : long-term stacking experiments for the reference case. Technical report EI-9944/A, Econometric Institute/ Erasmus University Rotterdam, 1999.
- [DVvA06] R. Dekker, P. Voogd, and E. van Asperen. Advanced methods for container stacking. *OR Spectrum*, 28(4) :563–586, October 2006.
- [Edm71] J. Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1 :127–136, 1971.
- [EG05] M. Ennigrou and K. Ghedira. Flexible job shop scheduling with multi-agent system and tabu search. *Journal Européen des Systèmes Automatisés-JESA*, Juin 2005.
- [Equ] Equipement. Different types of handling equipment. Bibliographie en ligne, disponible à l'adresse <http://www.kalmarind.com>, Consulté le 2 Janvier 2006.
- [Fer89] J. Ferber. Computational reflection in class based object oriented languages. In *In Proceedings of the Conference on Object-Oriented Programming Systems, Languages and Applications*, pages 317–326, New Orleans, Louisiana, 1989.
- [Fer95] J. Ferber. *Les systèmes multi-agents : vers une intelligence collective*. 1995.
- [GC04] Y. Guan and R. K. Cheung. The berth allocation problem : models and solution methods. *OR Spectrum*, 26 :75–92, 2004.
- [GD02] T. Garneau and S. Delisle. Evaluation comparative d'outils et environnements. In *Journées Francophones d'Intelligence Artificielle et de Systèmes Multi-Agents (JFIADSMA'2002)*, Lille,

- France, 2002. Département de mathématiques et d'informatique, Université du Québec à Trois-Rivières.
- [Geo83] M.P. Georgeff. Communication and interaction in multi-agent planning. In *Proceedings of the Third National Conference on Artificial Intelligence (AAAI-83)*, Washington, D.C., 1983.
- [GG06] HO. Günther and Kim KH. Gupta. Container terminals and terminal operations. *OR Spectrum*, 28 :437–445, 2006.
- [Ghe99] K. Ghedira. Une approche distribuée d'optimisation sous contraintes. Présentation au Séminaire OASIS/LIP6, Mars 1999.
- [Glo86] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, 5 :533–549, 1986.
- [GN91] N. Gupta and D. S. Nau. Complexity results of blocks-world planning. In *Proceedings of AAAI-91*, 1991.
- [Gol94] D. Goldberg. *Algorithmes génétiques*. Adisson Wesley, France, adisson wesley edition, 1994.
- [GRZ98] L.M. Gambardella, A.E. Rizzoli, and M. Zaffalon. Simulation and planning of an intermodal container terminal. *Simulation*, 7(2) :107–116, 1998.
- [GS92] N. Gupta and D. S.Nau. On the complexity of blocks-world planning. *Artificial Intelligence*, 56(2-3) :223–254, August 1992.
- [GV91] K. Ghedira and G. Verfaillie. Approche multi-agents d'un problème de satisfaction de contraintes : le problème d'affectation. Rapport Final du CERT-DERI 1/3412-00/DERI, 1991.
- [GV92] K. Ghédira and G. Verfaillie. A multi-agents model for the resource allocation problem : A reactive approach. In *Proceedings of the 10th European Conference on Artificial Intelligence*, pages 252–254, Vienna, Austria, 1992.
- [Har02] S. Hartmann. Generating scenarios for simulation and optimization of container terminal logistics. Working paper 564, University of Kiel, 2002.

- [HG05] M. Hammami and K. Ghédira. Cosats : A new cooperation model between simulated annealing and tabu search for the k-graph partitioning problem. In *The Fourth IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology*, Muroran, Japan, May 2005.
- [Hol75] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.
- [HWD03] L. Henesey, F. Wernstedt, and P. Davidsson. Market-driven control in container terminal management. In *Proceedings of the 2nd International Conference on Computer Applications and Information Technology in the Maritime Industries*, pages 377–386, Hamburg, Germany, 2003.
- [INP01] A. Imai, E. Nishimura, and S. Papadimitriou. The dynamic berth allocation problem for a container port. *Transportation Research Part B*, 35 :401–417, 2001.
- [Jam05] J-P. Jamont. *DIAMOND : Une Approche pour la conception de Systèmes Multi-agents Embarqués*. PhD thesis, INPG, Septembre 2005.
- [Jen97] K. Jensen. A brief introduction to coloured petri nets. In E. Brinksma, editor, *Lecture Notes in Computer Science*, volume 1217, pages 203–208, Enschede, The Netherlands, 1997. Springer-Verlag.
- [KB98] K. H. Kim and J. W. Bae. Re-marshaling export containers in port container terminals. *Computers & Industrial Engineering*, 35(3-4) :655–658, 1998.
- [KD04] P.J. Kalczynski and L. Dabrowski. Multi-container loading with grouping goals : an agent -based approach. *DASD*, 2004.
- [KGV83] S. Kirkpatrick, C. D. Gelatt, and Jr. M. P. Vecchi. Optimisation by simulated annealing. *Science*, 220(4598) :671–680, 1983.
- [Kim97] K. H. Kim. Evaluation of the number of rehandles in container yards. *Computers & Industrial Engineering*, 32(4) :701–711, 1997.

- [KK98] K. H. Kim and H.B. Kim. The optimal determination of the space requirement and the number of transfer cranes for import containers. *Computers & Industrial Engineering*, 35(3-4) :427–430, 1998.
- [KK99] K. H. Kim and H.B. Kim. Segregating space allocation models for container inventories in port container terminals. *International Journal of Production Economics*, 59 :415–423, 1999.
- [KK02] K. H. Kim and H.B. Kim. The optimal sizing of the storage space and handling facilities for import containers. *Transportation Research Part B*, 36 :821–835, 2002.
- [KKGY06a] M. Kefi, O. Korbaa, K. Ghédira, and P. Yim. Formalisation d’un modèle multi-agent par les réseaux de petri : application au problème de stockage de conteneurs. In *roceedings de la 6ème Conférence Francophone de Modélisation, Optimisation et Simulation des Systèmes : Défis et Opportunités-MOSIM’06*, Rabat-Maroc, 3-5 April 2006.
- [KKGY06b] M. Kefi, O. Korbaa, K. Ghédira, and P. Yim. Formalising an agent-based container stacking model via petri nets. In *Proceedings Volumes of the 12th IFAC International Symposium on Information Control Problems in Manufacturing-INCOM’06*, St Etienne-France, 17-19 May 2006. Elsevier Ltd.
- [KKGY07a] M. Kefi, O. Korbaa, K. Ghédira, and P. Yim. *Agent and Multi-Agent Systems : Technologies and Applications*, volume 4496/2007 of *Lecture Notes in Computer Science*, chapter Container Handling using Multi-Agent Architecture, pages 685–693. 2007.
- [KKGY07b] M. Kefi, O. Korbaa, K. Ghédira, and P. Yim. Heuristic-based model for container stacking problem. In *19th International Conference on Production Research-ICPR’07*, Valparaiso, Chile, 29 July-2 August 2007.
- [KKHK04] K. H. Kim, K. W. Kim, H. Hwang, and C. S. Ko. Operator-scheduling using a constraint satisfaction technique in port

- container terminals. *Computers & Industrial Engineering*, 46 :373–381, 2004.
- [KM03] K. H. Kim and K. C. Moon. Berth scheduling by simulated annealing. *Transportation Research-B*, 37 :541–560, 2003.
- [Koz00] E. Kozan. Optimising container transfers at multimodal terminals. *Mathematical and Computer Modelling*, 31 :235–243, 2000.
- [KP99] E. Kozan and P. Preston. Genetic algorithms to schedule container transfers at multimodal terminals. *International Transactions in Operational Research*, 6 :311–329, 1999.
- [KP04] O. Korbaa and P. Yim. Container assignment to stock in a fluvial port. In *International Conference on Systems, Man and Cybernetics*, The Netherlands, 2004.
- [KPR00] K. H. Kim, Y. M. Park, and K. R. Ryu. Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, 124 :89–101, 2000.
- [Lim98] A. Lim. The berth planning problem. *Operations Research Letters*, 22 :105–110, 1998.
- [LJI02] C. I. Liu, H. Jula, and P. A. Ioannou. Design, simulation, and evaluation of automated container terminals. *IEEE Transactions on Intelligent Transportation Systems*, 3(1) :12–26, 2002.
- [LM01] P. Legato and R. M. Mazza. Berth planning and resources optimisation at a container terminal via discrete event simulation. *European Journal of Operational Research*, 133 :537–547, 2001.
- [Mar96] P. Marcenac. Emergence of behaviors in natural phenomena agent-simulation. *Complexity International Review*, 3, 1996.
- [McD85] E. McDowell. A study of maritime container handling. Publication ORESU-T-85-003, Oregon State University, Sea Grant College Program Ads 406, 1985.
- [MHGWCCP03] R. L. Moorthy, W. Hock-Guan, N. Wig-Cheong, and T. Chung-Piaw. Cyclic deadlock prediction and avoidance for zone

- controlled agv system. *International Journal of Production Economics*, 83 :309–324, 2003.
- [MJLZ⁺00] K.G. Murty, Y-W. Wan J. Liu, C. Zhang, M. C. L. Tsang, and R.Linn. Dss for operations in a container shipping terminal. In *The Proceedings of the First Gulf Conference on Decision Support System*, pages 189–208, State of Kuwait, November 2000. Council of Ministers General Secretariat.
- [MLWL05] K.G. Murty, J. Liu, Y-W. Wan, and R. Linn. A decision support system for operations in a container shipping terminal. *Decision Support Systems*, 39 :309–332, 2005.
- [Moo01] K. C. Moon. A mathematical model and a heuristic algorithm for berth planning. *Telecommunication Grooming*, 2(3), May/June 2001.
- [MW01a] P.J.M. Meersmans and A.P.M. Wagelmans. Dynamic scheduling of handling equipment at automated container terminals. Technical Report 33, Econometric Institute/Erasmus University Rotterdam, 2001.
- [MW01b] P.J.M. Meersmans and A.P.M. Wagelmans. Effective algorithms for integrated scheduling of handling equipment at automated container terminals. Technical Report 19, Econometric Institute/Erasmus University Rotterdam, 2001.
- [NS002] La révolution du conteneur. *ISEAMAR*, (49), Novembre 2002.
- [NS003] La conteneurisation fluviale. *ISEAMAR*, (51), Janvier 2003.
- [OMM05] OMMP. Rapport annuel 2005. Rapport annuel, Office de la Marine Marchande et des Ports, Radès, Tunisie, 2005.
- [PCB06] P. Plamondon, B. Chaibdraa, and A. Benaskeur. A multiagent task associated mdp (mtamdp) approach to resource allocation. In *Spring Symposium on Distributed Plan and Schedule Management, AAAI 2006*, March 2006.
- [PEA90] J. PEARL. *Heuristique-Stratégies de recherche intelligentes pour la résolution de problèmes par ordinateur*. 1990.

- [PK01] P. Preston and E. Kozan. An approach to determine storage locations of containers at seaport terminals. *Computers & Operations Research*, 28 :983–995, 2001.
- [PK03] Y.M. Park and K. H. Kim. A scheduling method for berth and quay cranes. *OR Spectrum*, 25 :1–23, 2003.
- [Por06] Hamburg Port. Container throughput in teu. Bibliographie en ligne, Hamburg Port Authority Rankings 2005, disponible à l’adresse <http://www.hafen-hamburg.de>, 2006.
- [RG95] A. S. Rao and M. P. Georgeff. Bdi-agents : from theory to practice. In *Proceedings of the First Intl. Conference on Multiagent Systems*, San Francisco, 1995.
- [RJCB00] M. Rebollo, V. Julian, C. Carrascosa, and V. Botti. A multi-agent system for the automation of a port container terminal. In *Autonomous Agents workshop on Agents in Industry*, Barcelona, Spain, 2000.
- [RT] H. Rashidi and E. P.K. Tsang. Container terminals : Scheduling decisions, their formulation and solutions. Submitted to *Journal of Scheduling*.
- [RW02] P.A. Roach and I.D. Wilson. A genetic algorithm approach to strategic stowage planning for container-ships. Working paper, University of Glamorgan, UK, 2002.
- [SD80] R.G. Smith and R. Davis. Frameworks for cooperation in distributed problem solving. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(1), 1980.
- [SH88] D. Sculli and C.F. Hui. Three dimensional stacking of containers. *Omega*, 16(6) :585–594, 1988.
- [SH94] A. El Fallah Seghrouchni and S. Haddad. Représentation et manipulation de plans à l’aide de réseaux de petri. In *Deuxièmes Journées Francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, Voiron, 1994.
- [SH96] A. El-Fallah Seghrouchni and S. Haddad. A coordination algorithm for multi-agent planning. In *Proceedings of International*

- Conference on Multi-agent Systems(ICMAS'96)*, Kyoto,Japan, 1996. AAAI Press.
- [SHM98] A. El Fallah Seghrouchni, S. Haddad, and H. Mazouzi. Observation répartie et analyse des interactions dans un système multi-agents. In *Deuxièmes Journées Francophones Intelligence Artificielle Distribuée et Systèmes Multi-Agents*, Nancy, 1998.
- [Smi80] R. G. Smith. The contract net protocol : High-level communication and control in a distributed problem solver. *IEEE Transactions on computers*, 29(12) :1104–1113, 1980.
- [ST95] J. Slaney and S. Thiébaux. Tractable approximation to optimal planning in the blocks world. Technical Report 896, IRISA, 1995.
- [STA05] STAM. Rapport annuel 2005. Rapport annuel, Société Tunisienne d'Acconage et de Manutention, Radès,Tunisie, 2005.
- [Ste99] D. Steenken. Latest operational experience with satellite based container positioning systems. In *Proceedings IIR-Conference on Container Handling, Automation and Technology*, London, 1999.
- [SVS04] D. Steenken, S. VoSS, and R. Stahlbock. Container terminal operation and operations research- a classification and literature review. *OR Spectrum*, 26 :30–32, 2004.
- [SWZ01] D. Steenken, T. Winter, and U.T. Zimmermann. *Online optimization of large scale systems*, chapter Stowage and transport optimization in ship planning, pages 731–745. Springer, Berlin, 2001.
- [TH02] T. Thurston and H. Hu. Distributed agent architecture for port automation. In *Proceedings of the 26th Annual International Computer Software and Applications Conference*, Aout 2002.
- [TICD93] M. Taleb-Ibrahimi, B. De Castilho, and C.F. Daganzo. Storage space vs handling work in container terminals. *Transportation Research B*, 27 :13–32, 1993.

- [Tur] Turnover. Development of world container turnover. Bibliographie en ligne, disponible à l'adresse <http://www.hafenhamburg.de/en/>, Consulté le 2 Juin 2006.
- [TV02] P. Toth and D. Vigo. *The vehicle routing problem*. Society for Industrial & Applied Mathematics. SIAM, Philadelphia, 2002.
- [Val] R. Valette. *Les Réseaux de Petri*. LAAS, Toulouse.
- [VH04] I.F.A. Vis and I. Harika. Comparison of vehicle types at an automated container terminal. *OR Spectrum*, 26 :117–143, 2004.
- [VK03] I.F.A. Vis and R. De Koster. Transshipment of containers at a container terminal : an overview. *European Journal of Operational Research*, 147(1) :1–16, 2003.
- [VKRP01] I.F.A. Vis, R. De Koster, K.J. Roodbergen, and L.W.P. Peeters. Determination of the number of automated guided vehicles required at a semi-automated container terminal. *Journal of the Operational research Society*, 52 :409–417, 2001.
- [VKS01] I.F.A. Vis, R. De Koster, and M.W.P. Savelsbergh. Minimum vehicle fleet size at a container terminal. Technical report ERS-2001-24-LIS, Erasmus Research Institute of Management, 2001.
- [Wei99] G. Weiss. *Multi-agent systems : a modern approach to distributed artificial intelligence*. MIT Press, Massachusetts, 1999.
- [Win00] T. Winter. Online and real-time dispatching problems. *GCA-Verlag, Herdecke*, 2000.
- [Woo99] M. Wooldridge. *Multiagent systems : A modern approach to Distributed Artificial Intelligence*, chapter Intelligent agents. MIT Press, 1999.
- [WR00] I. D. Wilson and P.A. Roach. Container stowage planning : a methodology for generating computerised solutions. *Journal of the Operational Research Society*, 51 :1248–1255, 2000.
- [WRW01] I.D. Wilson, P.A. Roach, and J.A. Ware. Container stowage pre-planning : using search to generate solutions-a case study. *Knowledge based systems*, 14 :137–145, 2001.

- [YHSK05] P. Yim, S. Hanafi, F. Semet, and O. Korbaa. Outils d'aide à la décision pour la gestion des quais et l'optimisation d'un système de transport fluvial. 2005.
- [ZLW⁺01] C. Zhang, J. Liu, Y.W. Wan, K. G. Murty, and R.J. Linn. Storage space allocation in container terminals. *Transportation Research-B*, 37, 2001.
- [ZWLL02] C. Zhang, Y. Wan, J. Liu, and R.J. Linn. Dynamic crane deployment in container storage yards. *Transportation Research-B*, 36 :537–555, 2002.
- [ZYHH02] C. Zhang, R. Ye, S. Y. Huang, and W. J. Hsu. Two equivalent integer programming models for dispatching vehicles at a container terminal. Technical report, School of Computer Engineering, Nan yang Technological University, 2002.

Résumé

Les terminaux à conteneurs constituent des interfaces inter-modales essentielles pour le réseau de transport mondial. Une manutention efficace des conteneurs dans des terminaux est d'une importance cruciale pour la réduction des coûts de transport et la détermination des plans d'embarquement. Dans ce rapport de thèse, nous proposons principalement une approche de résolution distribuée à travers la description d'un modèle d'optimisation heuristique distribuée baptisé COSAH (*C*Ontainer *S*tacking via *m*ulti-*A*gent approach and *H*euristic method) qui permet de simuler, résoudre et optimiser l'espace de stockage disponible pour manier les départs et les arrivées des conteneurs dans un port fluvial ou maritime. Autrement dit, COSAH permet de minimiser le nombre total de mouvements parasites tout en respectant des contraintes dynamiques d'espace et de temps.

Les performances de COSAH sont ensuite évaluées sur des instances générées aléatoirement, ainsi que des instances extraites de la réalité d'un port maritime tunisien : le port de Radès. En effet, nous avons procédé à une étude expérimentale implémentant et comparant COSAH à la version centralisée associée, toutes deux basées sur un algorithme de recherche non informée et un algorithme de recherche informée. Les résultats obtenus, présentés et illustrés, montrent l'efficacité de COSAH en particulier, et d'une méthode d'optimisation heuristique distribuée alliant les deux concepts : Agent et Heuristique, en général.

Mots clés : approche multi-agent, éco-résolution, méthode heuristique, réseaux de Petri colorés, résolution distribué de problèmes. stockage de conteneurs, transport conteneurisé.

Abstract

Container terminals are essential intermodal interfaces in the global transportation network. Efficient container handling at terminals is of vital importance for the reduction of transportation costs as well as the determination of shipping schedules. In this thesis report, we propose a distributed solving approach through the development of a distributed *Heuristic-Optimization model denoted COSAH (C*Ontainer *S*tacking via *m*ulti-*A*gent approach and *H*euristic method) that allows to simulate, solve and optimize the amount of storage space for handling container departures and arrivals within a fluvial or maritime port. COSAH allows to minimize the expected total number of rehandles while respecting spatio-temporal constraints.

Performances of COSAH are then evaluated based on random-generated instances, as well as on instances extracted from a real-world case study : the port of Radès. Indeed, we conducted an experimental study implementing and comparing COSAH to the associated centralized version based on non informed-search algorithm and an informed-search algorithm. The obtained results, presented and illustrated, evince the efficiency of COSAH in particular, and of a distributed heuristic-optimization method combining two concepts : Agent and Heuristic, in general.

Keywords : colored Petri nets, container stacking, containerized transport, eco-solving, distributed problem-solving, heuristic method, multi-agent approach.