Thèse de Doctorat

Pour obtenir le grade de Docteur de

l'UNIVERSITÉ POLYTECHNIQUE HAUTS-DE-FRANCE

Spécialité: **INFORMATIQUE**

Présentée et soutenue par Placide NDUWAYO

Le 29 Juin 2020 à Valenciennes

École doctorale:

Sciences Pour l'Ingénieur (ED SPI 072)

Équipe de recherche, Laboratoire:

Département Informatique

Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines

(LAMIH - UMR CNRS 8201)

Formulations Mathématiques et Algorithmes pour le Problème d'Affectation des Quais du Cross-dock

JURY

Président du jury :	Luce Brotcorne, Directrice de Recherche, HDR, INRIA Lille - Nord Europe
Rapporteurs :	Xavier Gandibleux, Professeur, Université de Nantes Ali Ridha Mahjoub, Professeur, Université Paris Dauphine
Examinateurs:	Sandra Ulrich Ngueveu, Maître de conférences, Université de Toulouse Bernard Gendron, Professeur, Université de Montréal Igor Machado Coelho, Professeur, Université d'État de Rio de Janeiro, Brésil
Directeur de thèse :	Saïd Hanafi, Professeur, Université Polytechnique Hauts-de-France
Co-encadrant :	Christophe Wilbaut, Maître de conférences, Univ Polytechnique Hauts-de-France

Abstract

Cross-docking is a strategy originally introduced to optimize operations inside a warehouse as part of the optimization of the Supply Chain. Like traditional warehouses, products are collected from numerous freight yards such that suppliers, factories, manufactures,etc., using trucks, and are moved towards processing centers named cross-docks. At cross-dock yard, products first get unloaded on inbound dock doors. Afterwards, they are sorted according to their destinations and are immediately transferred, using handling devices, to appropriate outbound dock doors to be sometimes consolidated with other products of the same destination and are reloaded into shipping trucks. Unlike traditional warehouse where storage period of products is indefinite, for cross-dock, goods are unloaded and reloaded the same day without waiting in temporary storage area or can wait less than one day. In this PhD thesis, we study an NP-hard optimization problem raised by cross-dock referred to "Cross-dock Door Assignment Problem (CDAP)". The CDAP consists in assignment of incoming and outgoing trucks to inbound and outbound dock doors of cross-dock, respectively. The goal is to minimize the total transportation cost inside the cross-dock. The standard quadratic formulation of the CDAP includes the Generalized Assignment Problem as subproblem. In this dissertation, we perform an extensive cross-docking literature review. Then, we carry out a broad analysis of the standard quadratic formulation as well as the standard linearization of the CDAP. From this in-deph study, we propose several new non standard Mixed Integer Linear Programming models for the CDAP. To detect the best linear model among those we propose and those existing, we compare the performance of these models on instances proposed in the literature. We next propose a Lagrangian Relaxation approach to produce the best new lower bounds to optimal solution value. This Lagrangian Relaxation is applied to the model that produces the best LP relaxation bounds. The Lagrangian dual is solved using a subgradient algorithm. According to the experiments it seems that large-scale instances cannot be solved with an exact method in reasonable running times and memory requirements. Thus, we propose and implement two heuristics based on "Probabilistic Tabu Search" to operate efficiently with larger instances of the CDAP. To assess the effectiveness of these proposed heuristics, we compare their performance, first between them and then with recent heuristics in the literature. The results demonstrate the efficiency of the proposed approaches on data sets from the literature.

Keywords: *cross-docking, dock door assignment, mixed integer programming, linear programming relaxation, lagrangian relaxation, subgradient optimization, heuristics, probabilistic tabu search*

Résumé

Le cross-docking est une stratégie utilisée pour optimiser les opérations à l'intérieur de l'entrepôt dans le cadre de l'optimisation de la chaine logistique . Comme pour les entrepôts traditionnels, les produits sont collectés depuis plusieurs origines de production tels que les fournisseurs, les usines, les fabricants, etc., par des camions, puis ils sont acheminés vers des plateformes appelées cross-docks. Arrivés au cross-dock, les produits sont d'abord déchargés sur des quais d'entrée du cross-dock. Ils sont ensuite triés selon leurs destinations et sont immédiatement transférés, à l'aide des matériels de manutention, vers des quais de sortie correspondants pour, quelques fois être consolidés avec d'autres produits allant à la même destination et sont rechargés dans des camions sortants. Contrairement aux entrepôts traditionnels où la durée de stockage des produits est indéfinie, pour le cross-dock, ils sont déchargés et rechargés le même jour sans attendre dans la zone de stockage temporaire, ou peuvent attendre moins d'un jour. Dans cette thèse, nous étudions le problème d'optimisation NP-difficile apparaissant dans le cross-dock appelé "Cross-dock Door Assignment Problem (CDAP)". Le CDAP consiste à affecter des camions entrants et sortants, respectivement aux quais d'entrée et de sortie du cross-dock. Le but est de minimiser le coût total de transport à l'intérieur du cross-dock. La formulation quadratique standard du CDAP inclut le problème d'affectation généralisée comme sous-problème. Dans cette thèse, nous effectuons une revue de littérature étendue du cross-docking. Nous nous concentrons ensuite sur la modélisation mathématique du CDAP via une formulation quadratique standard, ainsi que sur la linéarisation standard de ce modèle. À partir de cette étude approfondie, nous proposons plusieurs nouveaux modèles linéaires non standard pour formuler le CDAP. Nous comparons ensuite ces modèles entre eux et ensuite avec les modèles récents de la littérature afin de déterminer le meilleur modèle linéaire. Nous proposons ensuite une Relaxation Lagrangienne pour produire de meilleures nouvelles bornes inférieures à la valeur de la solution optimale. La Relaxation Lagrangienne est appliquée au modèle qui produit de meilleures bornes inférieures de la relaxation continue. Le dual lagrangien est résolu par l'algorithme du sous-gradient. Les expérimentations montrent que les instances de grande taille ne peuvent pas être résolues par une méthode exacte en temps et ressources mémoires raisonnables. Nous proposons et implémentons donc deux heuristiques basées sur la recherche taboue probabiliste pour pouvoir traiter efficacement les instances de grande taille du CDAP. Pour évaluer l'efficacité de ces deux heuristiques, nous comparons leurs performances, d'abord entre elles, ensuite avec celles d'heuristiques récentes de la littérature. Les résultats obtenus montrent l'efficacité de nos méthodes sur des jeux de données de la littérature.

Mots-clés: *cross-docking, affectation des quais, programmation en nombres entiers mixtes, relaxation continue, relaxation lagrangienne, algorithme du sous-gradient, heuristiques, recherche taboue probabiliste*

Acknowledgements

A PhD thesis is big and interesting experience. At the end of this period of my PhD thesis, I would like to express my sincere gratitude to all of the persons who have supported and helped me all along this experience.

First of all, I sincerely thank my wife Belyse Arakiza and my two daughters Holy Dorita Nduwayo and Holy Thallia Nduwayo for your patience and confidence.

I especially want to thank my supervisor Pr. Saïd Hanafi for his support and encouraging advice.

My gratitude goes also to Mfc. Christophe Wilbaut, Dr. Raca Todosijević and Dr. Walid Guemri for good collaboration and contribution in my research.

I am grateful to Pr. Monique Guignard for kindly providing with benchmark instances as well as the results with which we have been able to evaluate our contributions.

For my beloved wife and daughters

For my Parents and my Parents-in-law

For all my families

LIST OF ACRONYMS

AIDC Auto ID Data Capture

AP Assignment Problem

B&B Branch and Bound

BKS Best Known Solution

CDAP Cross-dock Door Assignment Problem

CHR Convex Hull Relaxation

CPU Central Processing Unit

EJOR European Journal of Operational Research

FTL Full Truck Load

GA Genetic Algorithm

GAP Generalized Assignment Problem

GQ3AP Generalized Quadratic three-dimensional Assignment Problem

GQAP Generalized Quadratic Assignment Problem

IDE Integrated Development Environment

IT Information Technology

LR Lagrangian Relaxation

LS Local Search

LTL Less Than TruckLoad

MIP Mixed Integer Programming

MILP Mixed Integer Linear Programming

NP Non-deterministic Polynomial

OS Operating System

PTS Probabilistic Tabu Search

QAP Quadratic Assignment Problem

RFID Radio Frequency Identification

RLT Reformulation Linearization Technique

SC Supply Chain

SCM Supply Chain Management

SS Scatter Search

TDAP Truck-to-dock Door Assignment Problem

TDAPP-CD Truck-to-Door Assignment and Product placement Problem in Cross-Dock

TS Tabu Search

VRP Vehicle Routing Problem

WLAN Wireless Local Area Network

WMS Warehouse Management System

CONTENTS

LIST OF ACRONYMS vii

LIST OF FIGURES xiii

LIST OF TABLES xv

LIST OF ALGORITHMS xvii

Chapter 1 Introduction 1

Chapter 2 Warehouse operations management 7

2.1 Introduction 7

2.2 Products flow and units handling 8

2.3 Warehouse operations 8

2.3.1 Inspection and receiving 10

2.3.2 Put-away 10

2.3.3 Order-picking 10

2.3.4 Accumulation, sorting and packing 11

2.3.5 Shipping 11

2.4 Conclusion 11

Chapter 3 Cross-docking and Supply Chain 13

3.1 Introduction 14

3.2 Cross-docking definitions 15

3.3 Companies implementing cross-docking 17

3.4 Cross-docking decision levels . . . 19
3.4.1 Strategic level decisions . . . 19
3.4.2 Tactical level decisions . . . 21
3.4.3 Operational level decisions . . . 25
3.5 Operational optimization problems . . . 26
3.5.1 The Cross-dock Door Assignment Problem . . . 27
3.5.2 Truck Scheduling Problem . . . 28
3.5.3 Trucks Sequencing Problem . . . 29
3.5.4 Transshipment Problem . . . 30
3.5.5 Cross-dock Congestion Problem . . . 31
3.5.6 Vehicle Routing Problem with Cross-docking . . . 32
3.6 Cross-dock Door Assignment Problem . . . 34
3.6.1 Introduction . . . 34
3.6.2 Background for CDAP and Literature review . . . 37
3.6.3 Mathematical quadratic formulation for CDAP . . . 40
3.6.4 Some variants of the CDAP . . . 44
3.6.5 Connections with other Assignment problems . . . 49
3.7 Conclusion . . . 53

Chapter 4 Mathematical Programming Formulations for the Cross-dock Door Assignment Problem **55**

4.1 Introduction . . . 56
4.2 Standard Formulation for the CDAP . . . 58
4.2.1 Standard quadratic formulation . . . 59
4.2.2 Standard linearization for the CDAP . . . 60
4.3 Non Standard Assignment and Capacity constraints . . . 61
4.3.1 Assignment constraints . . . 62
4.3.2 Capacity constraints . . . 65
4.4 MIP models and integrality properties . . . 67
4.4.1 Eleven MIP models . . . 67

4.4.2 Integrality properties of MIPs . . . 69
4.5 Lagrangian Relaxation for the CDAP . . . 70
4.6 Computational results . . . 75
4.6.1 Comparison of models - integrality requirement on $z_{m,i,n,j}$ relaxed . 77
4.6.2 Comparison of models - integrality requirement imposed on $z_{m,i,n,j}$. 81
4.6.3 Lower bounds comparison for $\mathcal{M}^{2,1}$ and Nassief et al.(2016) . . . 86
4.7 Conclusion . . . 89

Chapter 5 Probabilistic Tabu Search for the Cross-dock Door Assignment Problem **91**

5.1 Introduction . . . 92
5.2 Main ingredients of the Probabilistic Tabu Search approaches . . . 93
5.2.1 Constructive heuristic to generate an initial solution . . . 95
5.2.2 Neighborhood structures and move evaluation . . . 96
5.3 Probabilistic Tabu Search . . . 103
5.3.1 Probabilistic Tabu Search : Variant 1 . . . 104
5.3.2 PTS : Variant 2 . . . 106
5.4 Computational Results . . . 108
5.4.1 Comparison of exhaustive and heuristic exploration of swap neighborhood . . . 108
5.4.2 Comparison with methods from the literature . . . 109
5.5 Conclusion . . . 113

Chapter 6 Conclusions and Future works **115**

Appendix A Computational results of new MIP Models for the CDAP **119**

A.1 Comparison of models - Integrality requirement on variables $x_{m,i}$, $y_{n,j}$ and $z_{m,i,n,j}$ imposed . . . 120
A.1.1 Quadratic model Q and Linear Models $\mathcal{M}^{0,0}$, $\mathcal{M}^{0,1}$, $\mathcal{M}^{1,0}$, $\mathcal{M}^{1,1}$, $\mathcal{M}^{2,0}$ for each class of instances and general average . . . 120
A.1.2 Linear Models $\mathcal{M}^{2,1}$, $\mathcal{M}^{3,0}$, $\mathcal{M}^{3,1}$, $\mathcal{M}'^{0,0}\mathcal{M}'^{0,1}$ and $\mathcal{M}'^{1,1}$ for each class of instances and general average . . . 122

A.2 Comparison of models - Integrality requirement on variable $z_{m,i,n,j}$ relaxed 124

A.2.1 MIP Models $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$, $\mathcal{M}'^{1,0}$, $\mathcal{M}'^{1,1}$, $\mathcal{M}'^{2,0}$ and $\mathcal{M}'^{2,1}$class of instances and general average 124

A.2.2 MIPs Models $\mathcal{M}^{3',0}, \mathcal{M}^{3',1}, \mathcal{M}'^{0',0}, \mathcal{M}'^{0',1}$ and $\mathcal{M}'^{1',1}$ for each class of instances and general average 126

A.3 Comparison of models - LP relaxation 128

A.4 Wilcoxon signed rank statistical tests for the models : Solution quality and runtime 131

A.5 Computational results of Lagrangian Relaxation approach for the CDAP . 132

Appendix B Computational results of PTS1 and PTS2 for the CDAP 135

Bibliography 139

LIST OF FIGURES

2.1 Pallet movement from upstream to downstream in a Supply Chain, Bartholdi and Hackman (2011) 8

2.2 Typical warehouse operations, Bartholdi and Hackman (2011) 9

3.1 An example of a cross-dock with temporary storage, Gelareh et al. (2020) . 16

3.2 Example of distribution center material handling devices 17

3.3 One single network configuration, Zhang (2016) 22

3.4 Single layer of cross-docks network configuration, Zhang (2016) 22

3.5 Hub-and-spoke system network configuration, Zhang (2016) 23

3.6 Mixed mode of door service, Shakeri et al. (2008) 25

3.7 Cross-docking facility operations, Stephan and Boysen (2011) 27

3.8 *Single-stage* or *two-touch* cross-dock, Gue and Kang (2001) 32

3.9 A *two-stage* or *multiple-touch* cross-dock, Gue and Kang (2001) 33

3.10 Cross-dock : Exclusive mode of dock door, Gelareh et al. (2020) 39

4.1 Confidence interval plot of the optimality gap-integrality requirement relaxed 79

4.2 Performance profile-solution values : integrality requirement relaxed 80

4.3 Performance profile-CPU times : integrality requirement relaxed 81

4.4 Confidence interval plot of the optimality gap-integrality requirement imposed 84

4.5 Performance profile-solution values : integrality requirement imposed . . . 85

4.6 Performance profile-CPU times : integrality requirement imposed 85

LIST OF TABLES

3.1 The Cross-dock Door Assignment Problems, dock doors assignment strategy 38

3.2 The variants of Cross-dock Door Assignment Problem (CDAP) 44

4.1 Number of constraints for each MIP model 68

4.2 Comparison of models - integrality requirement on variables $z_{m,i,n,j}$ relaxed 78

4.3 Comparison of models on each instance class - integrality requirement on variables $z_{m,i,n,j}$ relaxed 78

4.4 Comparison of models - integrality requirement imposed 82

4.5 Comparison of models on each instance class - integrality requirement imposed 82

4.6 Comparison of LP relaxations 86

4.7 Lower bounds comparison for $\mathcal{M}^{2,1}$ and Nassief et al. (2016) 88

5.1 Heuristic vs. Exhaustive exploration of swap neighborhood 109

5.2 Summary results on "SetA" instances 110

5.3 Summary results on "SetB" instances 111

5.4 Comparison of methods in terms of solution quality 111

A.1 Quadratic model Q and linear models $\mathcal{M}^{0,0}, \mathcal{M}^{0,1}, \mathcal{M}^{1,0}, \mathcal{M}^{1,1}$ and $\mathcal{M}^{2,0}$ . . 122

A.2 Linear Models $\mathcal{M}^{2,1}$, $\mathcal{M}^{3,0}$, $\mathcal{M}^{3,1}$, $\mathcal{M}'^{0,0}\mathcal{M}'^{0,1}$ and $\mathcal{M}'^{1,1}$ 124

A.3 MIP Models $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$, $\mathcal{M}'^{1,0}$, $\mathcal{M}'^{1,1}$, $\mathcal{M}'^{2,0}$ and $\mathcal{M}'^{2,1}$ 126

A.4 MIPs Models $\mathcal{M}^{3',0}, \mathcal{M}^{3',1}, \mathcal{M}'^{0',0}, \mathcal{M}'^{0',1}$ and $\mathcal{M}'^{1',1}$ 128

A.5 LP relaxation for all Models 130

A.6 p-values for all pairs of models in terms of solution value and CPU time - integrality requirement on variables $z_{m,i,n,j}$ imposed 131

A.7 p-values for all pairs of models in terms of solution value and CPU time - integrality requirement on variables $z_{m,i,n,j}$ relaxed 132

A.8 Detailed comparison of lower bounds for $\mathcal{M}^{2,1}$ and Nassief et al.(2016) . . 134

B.1 Detailed results on "SetA" instances 137

B.2 Detailed results on "SetB" instances 138

LIST OF ALGORITHMS

4.1 Sub-gradient Optimization Algorithm 74

5.1 Constructive heuristic 96

5.2 Exploration of Swap Neighbourhood $(o, X_i, X_{i'})$ 102

5.3 Probabilistic Tabu Search : General Framework 103

5.4 Candidate list construction in PTS1 104

5.5 Solution Selection in PTS1 106

5.6 Candidate list construction in PTS2 106

5.7 Solution Selection in PTS2 107

Chapter 1

Introduction

In today's competitive market where customers require short delivery time of their orders, Supply Chain (SC) with only classic distribution centers such that warehouses, is not enough to meet customers' requirements. Although warehouses remain involved and needed, when they are used alone they can not efficiently meet customers' need. In a typical warehouse, products are received at warehouse yard, then, they are inspected and stored into pallet racks waiting customers' orders. When a customer order is received into warehouse system, the product concerned by the order is retrieved from storage area it is stored and operations related to that order are started until it is loaded into shipping truck. Therefore, the main operations in a typical warehouse are **receiving**, **inspection**, **storage**, **order-picking**, **shipping**, etc. The interested readers can view the details in Agustina et al. (2010); Bartholdi and Hackman (2011); Van Belle et al. (2012). Two of those warehousing operations, namely **storage** and **order-picking** are the most expensive because of the highest cost of inventory holding and intensive workforce, see e.g., Bartholdi and Hackman (2011); Van Belle et al. (2012); Ramaa et al. (2012); Habazin et al. (2017).

Nowadays new strategies are implemented to fulfill warehouse gaps. Those new strategies help to improve the functioning of Supply Chain while remaining costs saving, see e.g., Stalk et al. (1992); Ladier and Alpan (2016a); Nguyen (2017). However, those strategies raise other new problems, many of which are combinatorial optimization problems. The emergence of new effective methods of resolution helps to solve those combinatorial optimization problems raised and adapt the Supply Chain against the increasing of customers' requirement. Cross-docking is one of strategic and innovative techniques implemented along Supply Chain. This strategy aims to eliminate the two expensive operations

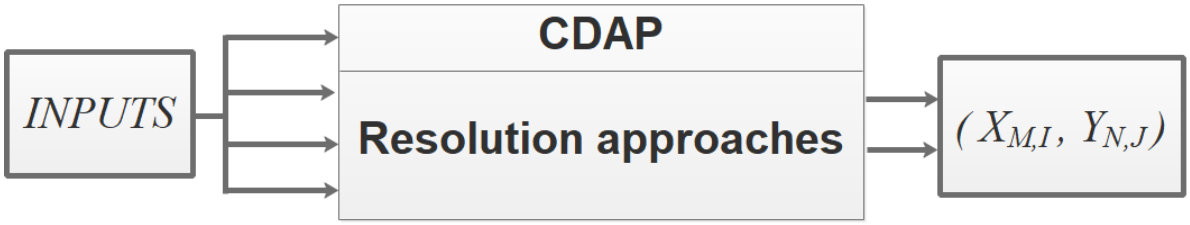
of warehouse because, for cross-docking products are transferred to their respective destinations the same day. As products are transferred immediately, cross-docking strategy shorten delivery time which is the most requirement for customers. In this PhD. thesis, we focus on one of combinatorial optimization problems raised by cross-docking strategy. That combinatorial optimization problem is concerning the management of a particular warehouse referred to **cross-docking facility** or **cross-dock**.

Many definitions of cross-docking appear in the literature review, see e.g., Bartholdi and Gue (2004); Agustina et al. (2010); Van Belle et al. (2012). As warehouses, cross-docking is concerning to transfer products coming from various freight yards such that suppliers, factories, manufacturers, etc. The products are immediately transferred from inbound to outbound dock doors of cross-dock without going through temporary storage area, or even though some of these products can go through temporary storage, they spend there just less than one day, see e.g., Van Belle et al. (2012); Guignard et al. (2012). Sometimes products are required to be transferred within less than an hour, refer to e.g., Bartholdi and Gue (2004). Thus, cross-docking facility is simply a transit area into which products "cross-dock" the facility to be shipped to the respective destinations.

Cross-docking strategy raises many combinatorial optimization problems that cannot be solved in reasonable time (polynomial time) using only exact methods because of the size and/or the practical constraints that make hard the resolution of those optimization problems. In view of this fact, many recent works in the literature review showed that the scientific community is increasingly turning to hybrid resolution techniques in which the aim is to combine at the best the components of different exact methods, heuristics and metaheuristics. Those combinatorial optimization problems are classified according to three levels of decision namely strategic, tactical and operational, see e.g., Agustina et al. (2010); Van Belle et al. (2012). The combinatorial optimization problem we tackle in this PhD. thesis belongs to the operational level of decision and it is referred to the **Cross-dock Door Assignment Problem (CDAP)**. The standard mathematical formulation of this optimization problem has been introduced for the first time by Zhu et al. (2009). We are going to focus our works particularly on theoretical and methodological aspects and will propose new innovative linear models and algorithms to solve efficiently that problem that is a well-known NP-hard combinatorial optimization problem, refer to e.g., Garey and Johnson (1983) about an NP-complete problem.

The Cross-dock Door Assignment Problem can be briefly described as follows : On cross-dock yard, receiving trucks get their products unloaded on input dock doors. Using

products handling devices inside cross-dock like forklifts, pallet jack, etc., the products are immediately transferred to output dock doors to be loaded into shipping trucks. Accordingly, the CDAP can be represented with inputs parameters that are a set of incoming trucks and a set of outgoing trucks to be docked on sets of inbound and outbound dock doors of cross-dock, respectively, with the goal of to look up an optimal assignment of those incoming and outgoing trucks on inbound and outbound dock doors, respectively, that minimizes the total weighted distance traveled by products handling devices inside cross-dock. In resolution approaches, we propose several new Mixed Integer Linear Pro-



gramming (MILP) models, a Lagrangian Relaxation approach and two heuristics based on Tabu Search (TS) to solve the Cross-dock Door Assignment Problem (CDAP). The proposed resolution approaches carry out :

- Assignment of a set M of all incoming trucks on a set I of inbound dock doors represented by a partition $X_{M,I}$
- Assignment of a set N of all outgoing trucks on a set J of outbound dock doors represented by a partition $Y_{N,J}$

All those proposed resolution approaches have proven a better performance to solve CDAP than those recently proposed in literature review.

This PhD. thesis contains three contributions, two of them have been successfully published and the third contribution is a manuscript in preparation for submission to the publication. The first contribution entitled "**A comparative study of formulations for a Cross-dock Door Assignment Problem**" has been accepted for publication in December 2018 into *The International Journal of Management Science (Omega)* and co-authored with Pr. Saïd Hanafi, Dr. Raca Todosijević, Dr. Oualid Guemri, Pr. Fred Glover and Pr. Shahin Gelareh (contribution order). The second contribution "**A Lagrangian heuristic approach for the Cross-dock Door Assignment Problem**" is a manuscript under preparation for submission to publication. It is an improvement of the conference article entitled "**A Lagrangian Relaxation for Cross-dock Door Assignment Problem**" presented into *11^{th} edition of Journées Polyèdres et Optimisation*

Combinatoire (JPOC'11) co-authored with Pr. Saïd Hanafi and Mcf Christophe Wilbaut. The third contribution "**Probabilistic Tabu Search for Cross-dock Door Assignment Problem**" has been published in March 2019 into *European Journal of Operational Research (EJOR)* and co-authored with Dr. Oualid Guemri, Pr. Saïd Hanafi, Dr. Raca Todosijević, and Pr. Fred Glover.

This dissertation is composed by 6 chapters and is organized as follows. The chapter 2 presents the main operations into warehouse, the cost incurred by operations and how big pallet are split into small units and the movement of those units. Among those warehouse's internal operations, the two most costly operations are reviewed.

The chapter 3 covers the state of the art for the cross-docking strategy. This chapter describes in more details cross-dock warehouse and the strategies involved by cross-docking ; the three levels of decision such that strategic, tactical and operational. The studies related to those decision levels include optimization problems like cross-dock network design (strategic level), cross-dock layout design (tactical level), dock-door assignment, truck scheduling, etc. (operational level). We detail different combinatorial optimization problems that arise in operational decision level because we focus on an operational optimization problem. Afterwards, we emphasize on Cross-dock Door Assignment Problem that is the case of study of this dissertation.

The chapter 4 is a broad development of two articles, a published journal and a conference articles. In this chapter, we carry out a broad analysis of a standard quadratic mathematical formulation proposed in Zhu et al. (2009) and a standard linearization for CDAP based on the linearization made in Fortet (1960). Then, we propose eight new non standard MILP models for the CDAP. To detect the best MILP model among those we proposed and the models recently proposed into the literature, we have performed an exhaustive empirical analysis using benchmark data sets from literature introduced by Guignard et al. (2012). Those MILP models are implemented and solved using CPLEX solver into Java environment. Afterwards, we choose one MILP model to which we apply a Lagrangian Relaxation (LR) procedure. The goal of the Lagrangian Relaxation approach is to find new and better lower bounds on the optimal solution value. We use sub-gradient optimization algorithm to solve the Lagrangian dual model. The same benchmark data sets, Guignard et al. (2012), are used to evaluate the performance of our Lagrangian Relaxation approach. The results of Lagrangian dual are compared with those given by LP relaxation of the corresponding MILP model and the recent results given by Lagrangian Relaxation from literature, Nassief et al. (2016). The Lagrangian Relaxation model is

solved using CPLEX solver and the sub-gradient optimization algorithm is implemented into Java environment.

The chapter 5 is also a large development of a published journal article. In this chapter, we implemented in Java environment two heuristics based on Probabilistic Tabu Search. We have referred those heuristics to PTS1 and PTS2. Our probabilistic algorithm is based on Tabu Search proposed by Glover (1986). We use Probabilistic Tabu Search (PTS) to solve efficiently large scale instances of CDAP for which there is so far no exact method that can solve them. We embedded an exploration heuristic to this algorithm for neighborhood exhaustive exploration avoidance. We have carried out computational experiments to analyse on the one hand the performance of PTS1 and PTS2 between and on the other hand we compare their performance with recent heuristics from literature.

The chapter 6 closes this dissertation with a general conclusion and draws up future research directions. The conclusion and future works are followed by two appendices that give, in tables, the detailed computational results and by a bibliography.

Chapter 2

Warehouse operations management

Contents

2.1 Introduction 7

2.2 Products flow and units handling 8

2.3 Warehouse operations 8

2.3.1 Inspection and receiving 10

2.3.2 Put-away 10

2.3.3 Order-picking 10

2.3.4 Accumulation, sorting and packing 11

2.3.5 Shipping 11

2.4 Conclusion 11

2.1 Introduction

A Supply Chain (SC) can be defined as a sequence of processes through which products flow from upstream (suppliers) into large pallets namely pallet-load to downstream (customers) into small units or into pieces. Warehouses as well as cross-docks are the intermediate points of Supply Chain where products are unloaded, inspected and managed before shipping. Those distribution centers are considered as inventory buffer points that link the flow of products between suppliers from upstream and customers to downstream.

2.2 Products flow and units handling

In Supply Chain, products are handled generally in small boxes while they are moving down from upstream to downstream. From upstream, products flow in large boxes e.g., in pallet-load. At each move to downstream, a pallet-load is successively broken down and sometimes repalletized into small units such that carton, inner pack, unit, etc., until it can no longer be divided, see e.g., Figure 2.1 below.

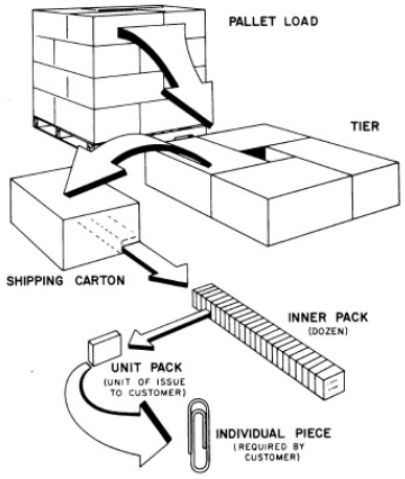


FIGURE 2.1 – Pallet movement from upstream to downstream in a Supply Chain, Bartholdi and Hackman (2011)

2.3 Warehouse operations

The Figure 2.2 is a schematic representation of the main operations that are handled in a typical warehouse. It gives the general pattern of product flow inside typical warehouse.

A warehouse is designed to have enough storage area to store incoming products from suppliers, that implies a rigorous management of storage space. The products can come from various freight yard into pallet-load and are shipped into cartons. Other products can arrive at warehouse into cartons and shipped into inner-packs, sometimes into pieces. The shipment into cartons or inner-packs,...will depend on the customers commands and whether products concerned by the customers commands are shipped to the same destination or not. This deep division of pallet-load requires more labor. Generally the smaller the handling unit, the greater the handling cost, see e.g., Habazin et al. (2017); Bartholdi and Hackman (2011). At the arrival, the products are received and inspected, those for which arrival coincides with customer's orders are not stored, they go immediately to next

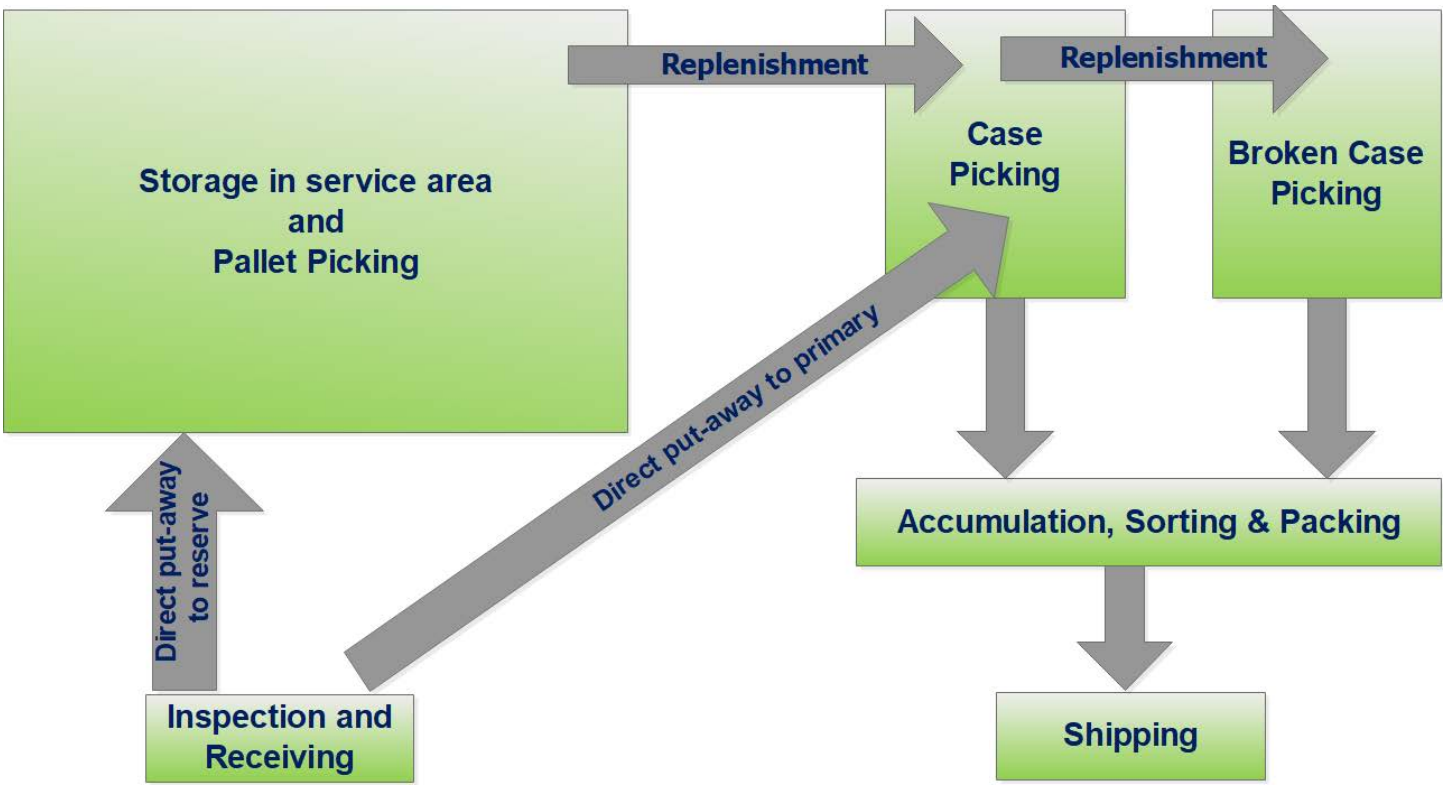


FIGURE 2.2 – Typical warehouse operations, Bartholdi and Hackman (2011)

operations for shipping, that is, direct put-away to primary operation (see Figure 2.2).

The time products spend in warehouse storage area is not known in advance. When customer order is received into Warehouse System, the product concerned by this customer order is retrieved from storage area and is picked for being loaded into shipping trucks (order-picking). We can summarize the process as follows:

When products arrive at warehouse

- Inspection : checking damaged products
- Receiving : reorganization and repackaging
- Put-away : direct put-away to reserve or to primary

When customer order is received

- Order-picking : preparing customer order
- Accumulation, sorting and packing
- Shipping

These main activities are briefly described in the following subsections.

2.3.1 Inspection and receiving

Arrival of products is notified and unloading starts. Cartons are scanned, products are inspected to raise possible exceptions such that damage, wrong description etc., any exception is notified. Large cartons are broken into small ones before being put-away for storage or being put-away to primary. When the large boxes are broken, additional boxes with small size are sometimes needed for repackaging, see e.g., Bartholdi and Hackman (2011).

2.3.2 Put-away

The operation put-away to primary happens when the arrival of products coincides with customer order, therefore the products concerned by the order are immediately handled for shipping. For put-away to reserve, products are stored into a specified location area of warehouse during indefinite time. For this operation, an appropriate storage space is determined. This is important because it makes easy to retrieve pallet from its location area upon the reception of customer order for this product. The put-away to reserve operation requires more labor to move product until its storage space. Before put-away to reserve operation, an inventory of storage areas is performed to know occupied, available areas and where everything is stored, see e.g., Habazin et al. (2017). Storage area of a product is scanned to make easy finding when it will be ordered.

2.3.3 Order-picking

When a customer order is received into warehouse system, an inventory of products is performed to verify if the customer order is possible for shipping. The flow time is the time elapsed from reception of customer order until the products concerned by the order are loaded into shipping truck. A Warehouse Management System (WMS) is an important software system used to manage and coordinate automatically the warehouse's main operations like keeping record of storage capacity, checking and updating stock, check of a customer order against available quantity of the ordered product in stock, raising shortage for a product and maintaining accurate inventory by recording warehouse transactions, etc. The WMS updates stock using real-time information, for instance Auto ID Data Capture (AIDC) technology such that barcode scanner, mobile computers, Wireless Local Area Network (WLAN), Radio Frequency Identification (RFID) to efficiently mo-

nitor the flow of products. For other details, interested readers can refer to Ramaa et al. (2012).

2.3.4 Accumulation, sorting and packing

Accumulation involves putting together different orders of a given customer in single package (packing). Before accumulation, orders are sorted according to their destinations. Those operations need a great precaution in order to meet customer's order accuracy.

2.3.5 Shipping

If all checks related to an order are done, it is ready to be loaded into shipping truck. The shipping process is not expensive compared to operations detailed above.

2.4 Conclusion

In this chapter, the flow of products from upstream to downstream into Supply Chain and the management of large pallet-loads into small packs have been presented. The main operations of a warehouse have been reviewed. Among those operations, we have shown that storage and order-picking are the most expensive due to the highest cost of inventory holding and intensive labor, respectively. In the next chapter, we are going to make the state of the art for cross-docking showing the improvement of cross-docking strategy on classical warehouse. Afterwards we are going to focus on the main combinatorial optimization problems that are raised by cross-docking strategy. We emphasize on **Cross-dock Door Assignment Problem (CDAP)** that is the case of study of this dissertation.

Chapter 3

Cross-docking and Supply Chain

Contents

3.1 Introduction 14

3.2 Cross-docking definitions 15

3.3 Companies implementing cross-docking 17

3.4 Cross-docking decision levels 19

3.4.1 Strategic level decisions 19

3.4.2 Tactical level decisions 21

3.4.3 Operational level decisions 25

3.5 Operational optimization problems 26

3.5.1 The Cross-dock Door Assignment Problem 27

3.5.2 Truck Scheduling Problem 28

3.5.3 Trucks Sequencing Problem 29

3.5.4 Transshipment Problem 30

3.5.5 Cross-dock Congestion Problem 31

3.5.6 Vehicle Routing Problem with Cross-docking 32

3.6 Cross-dock Door Assignment Problem 34

3.6.1 Introduction 34

3.6.2 Background for CDAP and Literature review 37

3.6.3 Mathematical quadratic formulation for CDAP 40

3.6.4 Some variants of the CDAP 44

3.6.5 Connections with other Assignment problems 49

3.7 Conclusion 53

In this chapter we describe in more details cross-docking in Supply Chain and how some transportation companies have managed to lower their cost after having implemented cross-docking. Different decisions classified in three levels to successfully implement the cross-docking are also discussed in this chapter. A literature review for most combinatorial optimization problems raised by cross-docking strategy is conducted. Finally, the combinatorial optimization problem tackled in this PhD thesis is described and different variants of this problem are also discussed.

3.1 Introduction

In Supply Chain, products packaged in cartons or other packaging materials are moved from one point to another point. Thereby, a carton of products is brought to travel a certain distance that separates the point upstream to the point downstream. Logistic or Supply Chain is then defined as a sequence of processes consisting of moving products from upstream to downstream and vice versa, in other words from producer to final customer and in reverse logistic, from the final customer to the producer. That is, Supply Chain Management (SCM) is the set of methods, resources and processes that are used to manage and improve the Supply Chain performance. The goal of a Supply Chain manager is to estimate the exact need at each stage, from the production of the finite products until delivery to the final customers.

At each step of flow of product until to its final destination, a supply chain cost is incurred. This cost can be expressed in several ways such that a function of distance that the product must travel from the point of production to the point of destination or vice versa, the damage that the product may incur during its displacements, the delays or distance that incurs transportation equipment used to deliver product from one point to another, the cost of storage of the product in a warehouse, etc. To try to optimize this cost, an important question is raised and need to be efficiently answered : how to move down products from upstream to downstream at very low cost ? Behind this question, several other questions and decisions can be listed, such that the best time and the frequency for the move, the best path to be taken in the logistic network, at what dock door of the platform to allocate mean of transportation, etc. This question does not only address to the means of transportation, but also the organization aspects.

The ultimate goal of Supply Chain is to meet the customer requirements, i.e, a high ser-

vice level, a short delivery time and a customer order accuracy with a minimum cost knowing that customer satisfaction is a key performance indicator in logistic. Cross-docking is one of strategies that helps to tackle those challenges.

3.2 Cross-docking definitions

Generally, the consumer goods factory produces the same consumer good in very large quantities. Then, these quantities of a given product are loaded until truck is fully loaded. The trucks fully loaded are sent to storage warehouses which can be located either near the factory or retail store or somewhere else in the network of suppliers, retailers and customers. In fact, retail store and customers generally do not need a high volume of the same product. Usually, small stores and customers need a small quantity of each product supplied by a storage warehouse. In the storage warehouse, the product are stored for unknown time. In addition, during the time the product is stored, it does not generate a profit. On the contrary it incurs an inventory holding which is one of the two most expensive operations in warehouse as mentioned in chapter 2.

Cross-docking is an alternative dynamic and Just-in-Time strategy which consists to transfer directly the products from trucks that come from different suppliers (called incoming trucks) to trucks going to different retailers and customers (called outgoing trucks) without storing those products. If shipping truck of a product is not available on the yard, that product will need to be stored for a while into a temporary storage area of the platform. Unlike classical warehouse, the storage time cannot exceed more than a day, sometimes products are stored for less than an hour, see e.g., Bartholdi and Gue (2004); Agustina et al. (2010); Van Belle et al. (2012); Guignard et al. (2012). To cope with this non storage of products, cross-docking strategy is involved for the products whose final retailers and customers are already known in advance before they leave the suppliers. That is, origins and destinations of products are known in advance. The fact products are not stored or can be stored for few hours in platform accelerates their flow from their origins going through distribution center to their respective destinations. Accordingly, all costs that were related to the storage and order picking operations namely inventory holding cost and intensive labor cost, respectively, are significantly reduced or completely dropped. This is the key difference between cross-docking strategy and classical warehouse.

In Van Belle et al. (2012), the authors give a general definition of cross-docking as "*the process of unloading freight from inbound vehicles and loading these goods into out-*

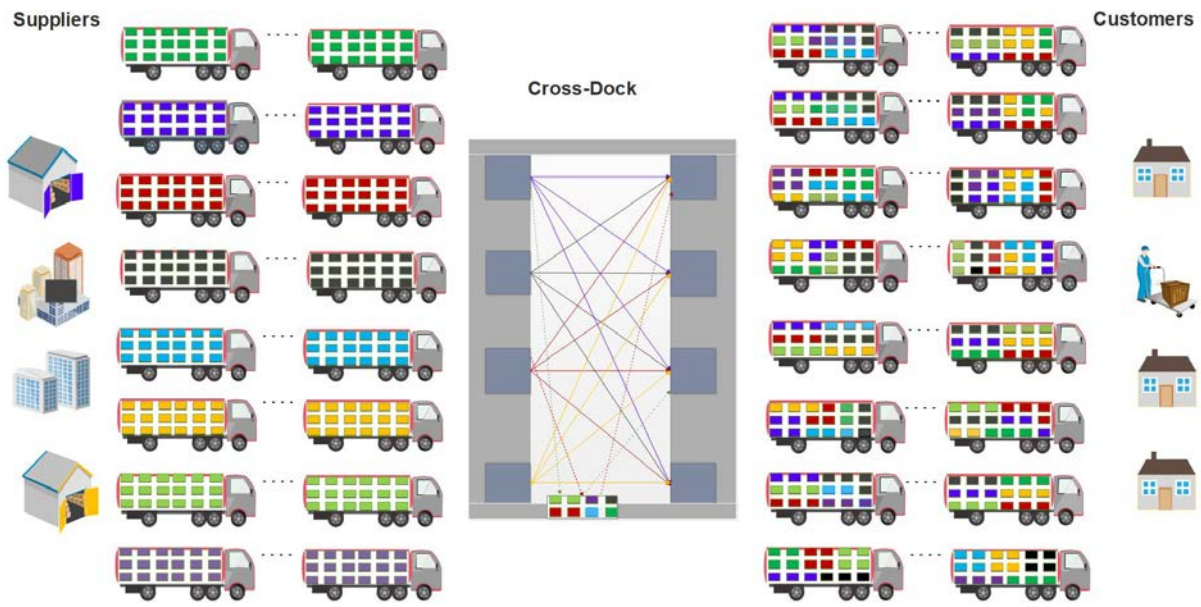


FIGURE 3.1 – An example of a cross-dock with temporary storage, Gelareh et al. (2020)

bound vehicles, with minimal handling and with little or no storage in between". Cross-docking strategy takes place in a distribution warehouse called "**cross-docking facility**" or "**cross-dock**".

The cross-dock warehouse consists of a set of dock doors on each of its sides (inbound and outbound), ideally without temporary storage area. Cross-docks have several layouts, the Figure 3.1 taken from our published paper in Gelareh et al. (2020) is one of the schematic representations of cross-dock with a temporary storage area and an example of flow of products inside the facility. In the same way as in Van Belle et al. (2012), the term cross-docking expresses the process of receiving products on inbound dock doors and then transferring them directly across the cross-dock to outbound dock doors. In simple terms, incoming products arrive through means of transportation such that trucks, trailer trucks and are docked on inbound dock doors of the cross-dock terminal. Once incoming trucks or trailer trucks have been docked, the packed products (pallets) get unloaded, sorted and screened to identify their end destinations. Afterwards, the pallets are moved to outbound dock doors of cross-dock terminal using manual material handling devices such that hand pallet truck, forklift, electric pallet truck, etc., see e.g., Figure 3.2, or using an automated mode such that a network system of conveyor belts. To outbound dock doors side, products can be consolidated with those in pending for the same destination and are loaded into outgoing trucks that are already docked. After loading operation, the products can then make their way to the final destinations. Generally, the number of incoming and outgoing trucks by period is larger than the number of dock doors of cross-dock. A dock



Source :www.sunnforest.com/Material-Handling-Equipment/Warehouse-Equipment.html

FIGURE 3.2 – Example of distribution center material handling devices

door of cross-dock is an arranged area where trucks or trailer trucks are unloaded, for inbound dock doors and loaded, for outbound dock doors. Here, to dock a truck or trailer truck means to place it to the dock door. The operations of handling packed products in cross-dock are sometimes similar to those handled on harbor or airport. In fact, at harbor yard, after ships are docked, the containers get unloaded and then, they are put away in temporary gate, waiting to be loaded into another ship or a trailer truck. At airport, the determination of how airplanes are affected to gates and the system of transferring passengers from gate to gate aims to minimize traveled distance by passengers. Fore more details, interested reader can refer to e.g., Van Belle et al. (2012); Zeinebou and Abdellatif (2013).

3.3 Companies implementing cross-docking

Cross-docking strategy has been implemented and has lowered the cost for several transportation companies. For instance, the distribution chain company Wal-Mart in US is cited to be on the top of position of retailer companies to begin implementation of the cross-docking in the retail sector in the late 1980s. Wal-Mart success began by defining the goals that were consisting to provide to customers the access to quality of goods, to make those goods available where and when customers need them, to develop a cost structure that enables competitive pricing, to build and maintain a reputation for absolute reliability, for more details, readers can refer to Stalk et al. (1992); Nguyen (2017).
The following paragraph is the citations of Stalk et al. (1992) about the success of Wal-Mart company :

"The key to achieving these goals was to make the way the company replenished

> inventory the centerpiece of its competitive strategy. This strategic vision reached its fullest expression in a largely invisible logistics technique known as 'cross-docking'. In this system, goods are continuously delivered to Wal-Mart's warehouses where they are selected, repacked, and then dispatched to stores, often without ever staying in inventory space. Instead of spending valuable time in the warehouse, goods just cross from one loading dock to another in 48 hours or less." Stalk et al. (1992)

In Bartholdi and Gue (2004), the authors stated that the transportation company Home Depot has operated cross-docking in Philadelphia that serves more than 100 stores in the Northeast. But, before using cross-docking, Home Depot used to order each store separately from suppliers and the orders were sent directly to Home Deport stores in partially loaded truck known as Less Than TruckLoad (LTL). Now, to reduce the costs, the company uses cross-docking. Here is how Home Deport has processed to reduce transportation costs from suppliers to its facility and from facility to retailers. Therefore, instead of using LTL, all the orders are consolidated among the stores on suppliers side and are loaded in full truck load quantities. That is, each truck (incoming truck) leaves vendor and comes to Home Deport facility fully loaded with consolidated orders, this is known as Full Truck Load (FTL). On a specific day in week, each of 100 Home Deport store places orders for each supplier or vendor at time. All the orders are consolidated on the supplier side and are loaded into fully loaded truck, then fully loaded trucks are sent to Home Deport cross-dock. As the orders arrive at cross-dock already consolidated, the workers are only transferring products to loading trucks (outgoing trucks or delivery trucks) destined to distribute them to individual stores, retailers or customers. Outgoing trucks are fully loaded with consolidated orders coming from many suppliers before leaving the cross-dock facility. Transportation cost is extremely reduced by the fact that incoming and outgoing trucks come in and leave cross-dock in fully loaded.

Other examples of successful application of cross-docking in Europe are viewed in Ladier and Alpan (2016a). In this work, authors made a survey during which they visited eight transportation companies located in France that apply cross-docking strategy in their daily transportation activities. The authors compared the problems related to cross-docking that are studied in academic literature and application of cross-docking in those transportation companies visited. The goal was to compare academic study related to cross-docking and cross-docking application in industry. After the survey, the authors raised a certain gap between cross-docking in theory and in industry practice and gave some directions of research to bring literature closer to practice.

3.4 Cross-docking decision levels

Even though cross-docking offers significant cost saving, like any functional system, it is not a complete package of solutions for all the problems of the Supply Chain. In fact, cross-docking success, when this strategy is used appropriately, depends on how well the transportation network is designed and managed. For instance, the success of cross-docking depends on how well the network of cross-docks (cross-docking network) is located to connect origins and destinations of products together. Therefore, the success of an individual cross-dock is influenced by how well is designed, exploited and managed the whole network of cross-docks in which that individual cross-dock is located, see e.g., Yang et al. (2010). Other factors that affect an individual cross-dock are for instance the size, shape, number of dock doors of the platform ; the number of handling devices used inside cross-dock, the time spent by material handling devices traveling between dock doors pair while they are moving pallets, congestion caused by the movements of those material handling devices, etc. Therefore, to successfully plan, design, implement and manage a cross-dock, several decisions are to be made. Those decisions are classified in three levels, namely, strategic, tactical and operational.

3.4.1 Strategic level decisions

The decisions to be made at the strategic level are long-term decisions and are usually concerning physical characteristic of cross-dock. Those decisions are taken in order to strengthen the time span of the cross-dock and to influence further operations inside the platform. Some examples of strategic decisions to be taken into account when planning to design cross-docking are as follows:

The cross-docking network refers to a network of one or several cross-docks connected together. It is a subsystem of supply chain formed by one or several cross-docks with inbound and outbound transport routes and the stakeholders connected to the cross-docks by those routes, see e.g., Buijs et al. (2014).

The location of an individual cross-dock geographically in cross-docking network and in the network of suppliers, retailers and customers is an important strategic decision. To learn more about location of cross-dock, we refer the interested reader to e.g., Agustina et al. (2010); Van Belle et al. (2012).

The Layout design of a cross-dock is also an important physical characteristic. Once

location of individual cross-dock in cross-docking network is determined, the decision concerning the layout of cross-dock has to be made. The layout design of a cross-dock refers to the shape, the size and the number of dock doors of a cross-dock. The interested reader can refer to the extensive overview made on cross-docking concept in Van Belle et al. (2012). As far as shape is concerned, cross-docks have a lot of variety of shapes which are usually described by a letter such that I, L, T, H, U, E and X. Usually the number of dock doors imposes the shape that must have a cross-dock.

In Bartholdi and Gue (2004), the authors have focused on how the shape affects cross-dock performance. In view of this fact, they have studied different shapes of cross-docks with the purpose to find out what have to be the best shape for cross-dock. They have listed the commonly used shapes such that I, L, and T and other unused such that U, H and E. The authors conducted computational experiments on I, L, T, H, and X-shape cross-docks considering several characteristics such that the size, the shape, the flows concentration and a part of inbound dock doors. Through the results of their computational experiments, they concluded that performance of cross-dock depends on two factors such that the size and the shape and that most of cross-docks are I-shape, that is, long, narrow rectangles. The authors argued that for number of dock doors, I-shape is the best layout with few than about 150 dock doors on each of the two sides of cross-dock, T-shape is the best efficient for cross-dock with intermediate size, that is, between 150 and 200 dock doors and that X-shape is the best for approximately more than 200 dock doors.

In Stephan and Boysen (2011), the authors stated that L-shapes and H-shape cross-docks are less efficient than I-shape cross-docks because those cross-docks provide additional corners that not help improving traveled distance.

The cross-docks visited in Ladier and Alpan (2016a), seven out of the eight cross-docks are I-shape. In fact, all those cross-docks have less than one hundred dock doors where I-shape is the most efficient in accordance with Bartholdi and Gue (2004). Another reason that the authors argued to explain why all cross-docks are I-shape is that in France the biggest cross-docks are built by real estate agents who choose I-shape layout because it can be easy to divide when it comes to renting to logistic companies. This survey is a chapter of the thesis of Ladier (2014).

Number of dock doors is also a strategic decision. Once the shape of cross-dock is defined, the number of dock doors must also be decided and how they are placed along the cross-dock, either on only one side or on two sides or on all the sides of the platform.

Clearly, the decision of shape is influenced by the size of the platform.

Internal transportation system is also a strategic decision that must be considered. This decision is concerning how products are moved from inbound to outbound dock doors of the cross-dock. Thus, products can be moved either manually by workers using material handling devices, see e.g., Figure 3.2, or using an automated mode like a network system of conveyor belts or a combination of those two first internal transportation modes. For more details about flow of products inside cross-dock, interested reader can refer to e.g., Stephan and Boysen (2011); Van Belle et al. (2012); Ladier and Alpan (2016a). In Ladier and Alpan (2016a), the majority of visited cross-docks use internal transportation system in the following distribution: 63% of cross-docks use manual transportation, 13% use automated transportation mode and 25% use combined mode of transportation. For the visited cross-docks, the authors explained why an automated and combination modes are used widely more than what is stated in academic literature by the fact that when automated and combined transportation mode are used, some operational decisions are delegated to Information Technology (IT) system.

The number of material handling devices used inside cross-dock is also a strategic decision, refer to e.g., Agustina et al. (2010).

3.4.2 Tactical level decisions

Decisions made on tactical level are mid-term decisions for cross-dock. Those decisions influence directly operational decisions. Some of tactical decisions include:

Flow of products through cross-docking network configuration. The cross-docking network configuration refers to how products flow from cross-dock to cross-dock in the cross-docking network until they are delivered to retailers and customers. There are several cross-docking network configurations. Those configurations are classified according to the size of the cross-docking network. For more details about cross-docking network configurations, interested readers can refer to Buijs et al. (2014). The simple cross-docking network configuration is so-called *one single* configuration. In this network configuration, there is only one cross-dock in which all products go through, see e.g., Figure 3.3. The second cross-docking network configuration is referred to *single layer of cross-docks* configuration. In this network configuration, several cross-docks are connected together but each product crosses one cross-dock bypassing other cross-docks, see e.g., Figure 3.4. The third network configuration is called *hub-and-spoke system* network configuration. In this

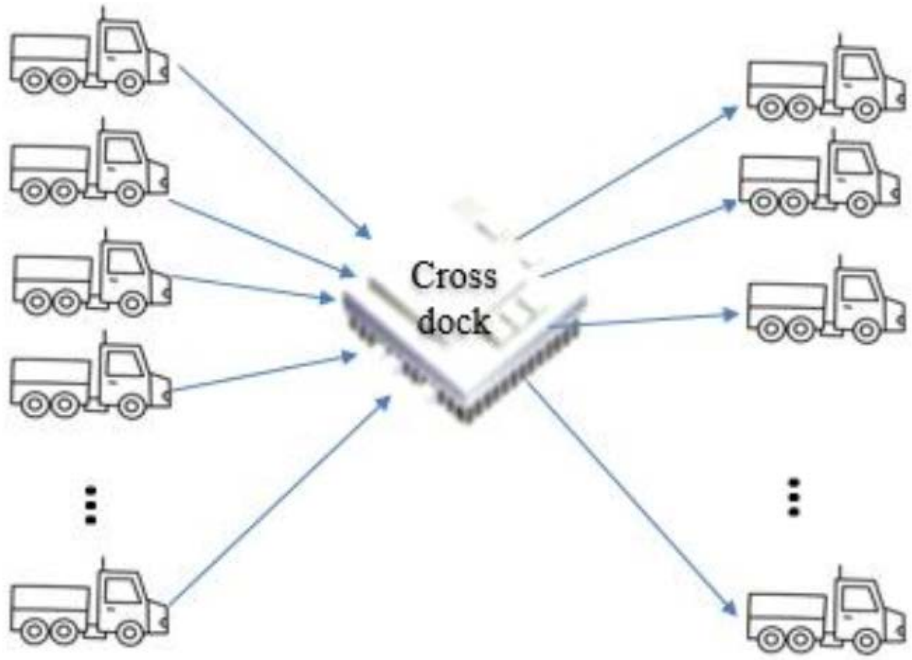


FIGURE 3.3 – One single network configuration, Zhang (2016)

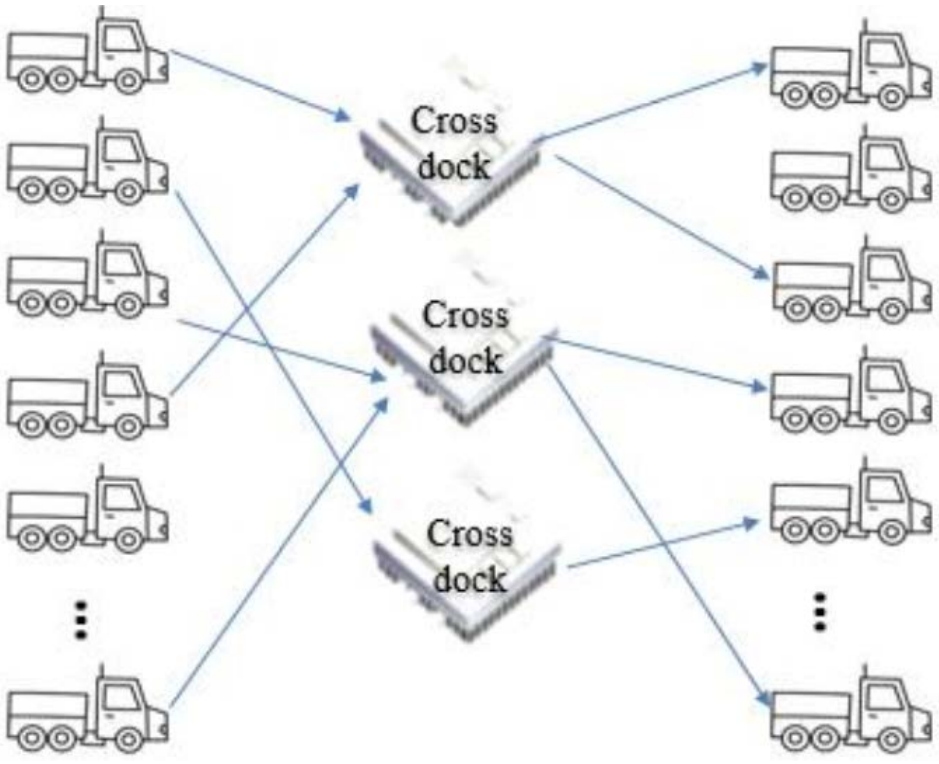


FIGURE 3.4 – Single layer of cross-docks network configuration, Zhang (2016)

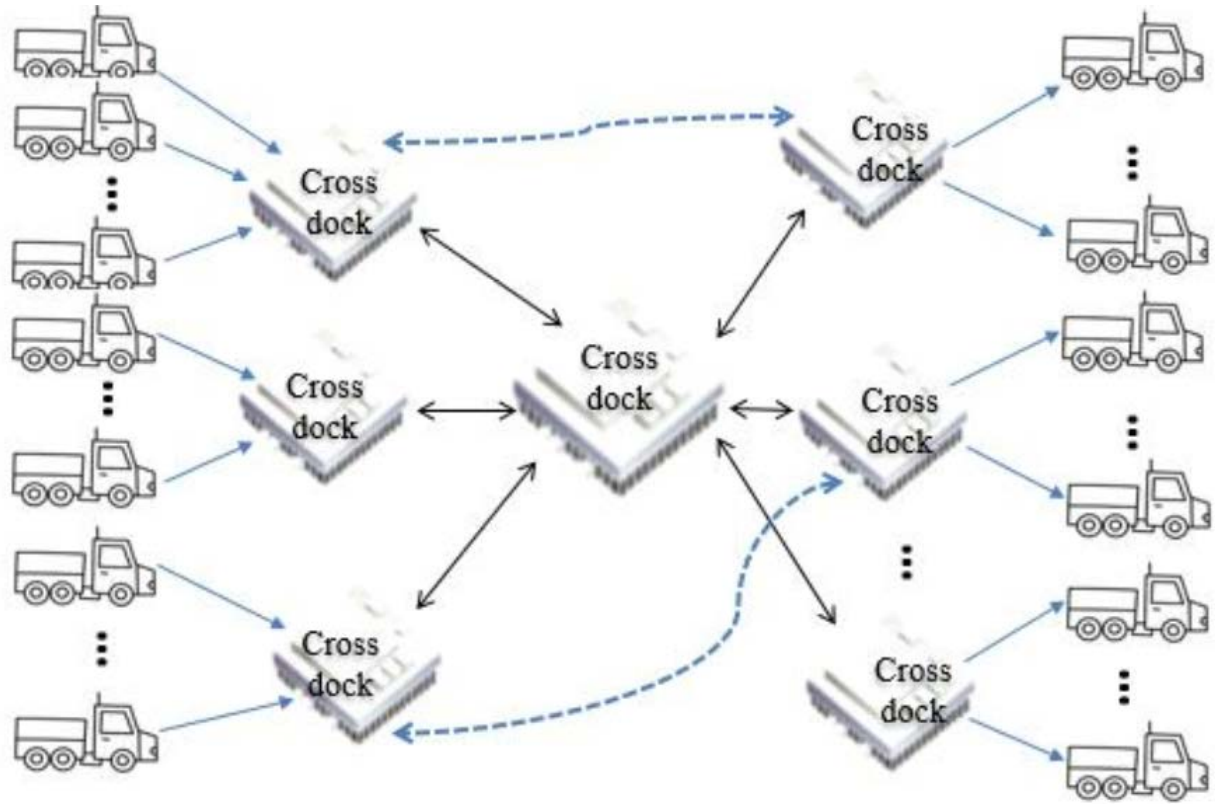


FIGURE 3.5 – Hub-and-spoke system network configuration, Zhang (2016)

configuration, goods can be shipped through multiple cross-docks grouped in stages, see e.g., Figure 3.5.

Preemption is also a tactical decision. In individual cross-dock, operations manager must decide whether or not to allow ongoing unloading/loading of a truck to be interrupted. If preemption is allowed, the unloading or loading of truck can be interrupted at any time, then the truck is put-away and is replaced by the next one. The uncompleted unloading/loading truck that has been put-away will be reassigned and processed later either to the same dock door or to an another dock door according to the case that minimizes the cost. The cost of moving away the truck from the dock door and the cost to back it for reassignment to dock door are taken into account. In Ladier and Alpan (2016a) no one of the visited cross-docks use preemption.

Temporary storage and its capacity is consisting to decide whether intermediate storage area is allowed or not and if the capacity of storage area is limited or unlimited. If temporary storage is allowed, an unloaded product can be put-away into temporary storage area for a short time that can not exceed 24 hours. Practically, a product have to be temporarily stored in the prepared storage area in the case the shipping truck destined to that product is not yet available on outbound dock door (not yet docked), see e.g., Ladier

and Alpan (2016a). Resources like number of workers or conveyed belts used to handle products inside the platform are also decided into tactical level decisions.

Service mode of dock doors of cross-dock is another important tactical decision. Having two kinds of dock doors (inbound and outbound), to set up and optimize the cross-dock, operating mode of dock doors is decided. There are two common operating modes of dock doors: Exclusive mode and mixed mode of service and the third service mode not commonly used is referred to combined mode, see e.g., Van Belle et al. (2012); Ladier and Alpan (2016a).

3.4.2.1 Exclusive mode

For a good management of the cross-dock and a good traveling of material handling devices between inbound and outbound dock doors of the platform, the service mode of dock doors that is commonly used is exclusive mode. That is, inbound and outbound dock doors are dedicated and fixed exclusively to inbound and outbound operations, respectively. In simple words, when this service mode is used, inbound side is used to serve origins exclusively and outbound side is used to serve destinations exclusively. The Figure 3.1 depicts an axample of I-shape cross-dock configured to use exclusive mode of dock doors with a small temporary storage area. In Ladier and Alpan (2016a), four cross-docks out of all visited cross-docks use exclusive mode of dock doors. As this service mode have the fixed inbound dock doors for inbound operations and fixed outbound dock doors for outbound operations which makes easy internal operations for managers, the exclusive mode is widely used. In this service mode, each outbound dock door can serve a fixed set of destinations, in that case it is called *destination exclusive mode*, and each inbound dock door can serve a fixed set of origins, in this case it is called *origin exclusive mode*, see e.g., Ladier and Alpan (2016a).

3.4.2.2 Mixed mode

In this service mode of dock doors, an intermixed sequence of receiving and shipping trucks to be processed per dock door is allowed, that is, a same dock door can have dual function of receiving incoming and shipping trucks. In other words, incoming or outgoing truck can be assigned at any dock door or to the same dock door. In Ladier and Alpan (2016a), five out of eight visited cross-docks use mixed mode of service. The Figure 3.6 depicts the mixed mode of dock doors.

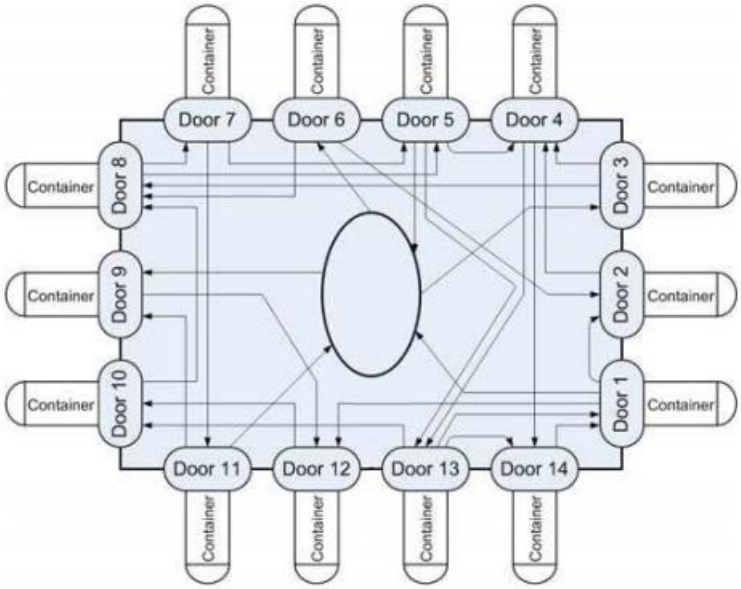


FIGURE 3.6 – Mixed mode of door service, Shakeri et al. (2008)

3.4.2.3 Combined mode

The combination mode of service is the combination of exclusive and mixed mode. In the combined mode, a subset of dock doors works in exclusive mode and the rest of dock doors work in mixed mode, see e.g., Van Belle et al. (2012). The combined mode cannot be covered in practice because it would cause confusion between exclusive and mixed mode of dock doors,see e.g., Ladier and Alpan (2016a).

3.4.3 Operational level decisions

The operational level decisions are short-term decisions. Those operational decisions are made on daily or weekly basis by cross-dock operations manager. This decision level raises several questions related to the management of cross-dock operations. For instance, the question like how trucks will be unloaded when they arrive on cross-dock ? This question raises a set of decisions that must be made such that unloading will be done either manually using material handling devices or with automated system ; whether big cartons will be divided and repacked or not ; etc. Other questions like when unloading operation will start and when trucks leave cross-dock ? The cross-dock operations manager can decide, for example, either that all incoming and shipping trucks have to be available at the cross-dock ground from the beginning of operations until the end or at any time incoming truck arrives it is processed and as soon as it ends up being unloaded it leaves

and outgoing truck leaves as soon as it ends up being loaded.

A cross-dock operations manager can decide also to ship goods directly from origins to destinations in the case all shipping trucks are available on the platform yard. In that case no product can go through temporary storage area. This is the case for frozen or fresh products that have to be shipped immediately to the respective destinations so that they are not damaged. The products that have been temporarily stored can be consolidated inside cross-dock with new arriving products of the same destination before being loaded into shipping truck. Other decision can be made about the number of material handling devices, that is, the fewer or more forklifts to use inside the cross-dock to handle shipments. For all those operational decisions, the interested reader can refer to e.g., Ladier and Alpan (2016a), for the impact of operations manager decisions on the operational effectiveness of cross-dock, the reader can refer to e.g., Yang et al. (2010) in which the authors have used a computer simulation considering an I-shape cross-dock and forklifts as devices used to handle pallets inside the cross-dock.

3.5 Operational optimization problems

As we have just seen in Section 3.4.3 above, the operational level raises a lot of decisions that are made on daily or weekly basis. It is obvious that each decision raises one or more optimization problems so that if we solve this or these optimization problems, we optimize the operational level of the cross-dock at the point where these problems have been solved. An optimization problem consists of finding the best solution among all feasible solutions. The best solution can be the shortest path, the shortest duration of operations (makespan), a maximum benefit using a minimum resource, etc. Those optimization problems are usually NP-hard. An optimization problem is NP-hard if it can not be solved optimally in polynomial time, i.e., reasonable time using a known polynomial algorithm, assuming P !=NP. To solve such problems, we use approximation algorithms such that heuristics that do not guarantee an optimal solution but an approximate solution in reasonable time.

As our study addresses one of the operational class of optimization problems, we perform in the following sections a large description of the main optimization problems that occur at the operational level of decision. Then we emphasize on the combinatorial optimization problem “The Cross-dock Door Assignment Problem (CDAP)” that we deal with in this PhD thesis as well as the configuration of the cross-dock to which we apply that

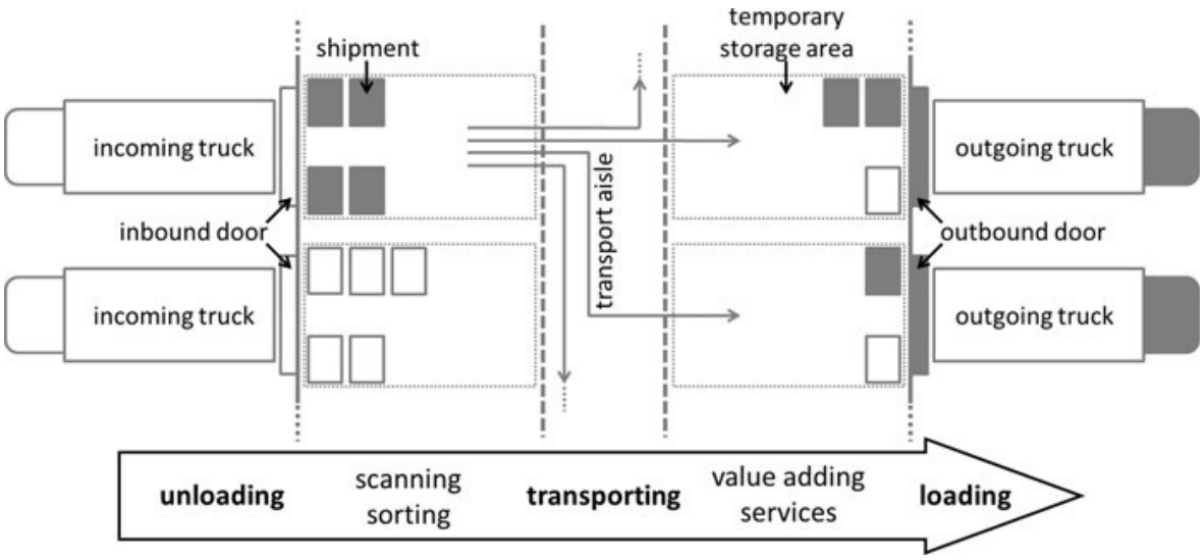


FIGURE 3.7 – Cross-docking facility operations, Stephan and Boysen (2011)

optimization problem. We continue with presenting different variants of this optimization problem as proposed in literature.

3.5.1 The Cross-dock Door Assignment Problem

The Cross-dock Door Assignment Problem (CDAP) consists of assigning trucks to dock doors in order to optimize (minimize, maximize) operations inside the cross-dock. The schematic representation of those operations inside cross-dock is depicted in Figure 3.7. The Cross-dock Door Assignment Problem assumes that all incoming and outgoing trucks are available on cross-dock yard before the beginning of planning the cross-docking operations. The objective is to minimize transportation cost inside the cross-dock by finding an optimal assignment of trucks dock doors. Therefore, the distance traveled by pallets handling device inside the cross-dock while transferring products between inbound and outbound dock doors can be reduced if outbound dock door on which a destination truck of item is assigned is near to the inbound dock door where the corresponding origin is docked. The Cross-dock Door Assignment Problem is also known as the Truck-to-dock Door Assignment Problem (TDAP) because this optimization problem deals with assignment of trucks to dock doors. In fact, when an incoming truck arrives on the facility, it is necessary to decide on what inbound dock door it will be assigned. Due to the flows of products between incoming and shipping trucks, it is also necessary to decide on what outbound dock door this shipping truck will be assigned. Therefore, a good assignment of all incoming and outgoing trucks to dock doors will decrease dock door delay, will affect

the total time of unloading of products, total traveling time inside the facility and the time during which all products get loaded, in short, it will reduce cross-dock internal operations duration. The traveled distance can be expressed as the total time used to transfer all products from inbound to outbound dock doors. The Cross-dock Door Assignment Problem is more explicitly explained further in section 3.6

3.5.2 Truck Scheduling Problem

Having more trucks than dock doors available, dock doors can be scheduled over time. While Cross-dock Door Assignment Problem takes into account the space dimension explicitly by seeking to minimize the total weighted distance traveled inside the cross-dock by material handling devices, this optimization problem ignores temporal dimension. That is, CDAP does not consider arrival and departure time of each truck at and from cross-dock (time window of truck). The Truck Scheduling Problem takes into account the temporal dimension explicitly by considering the time window of trucks, that is, arrival and departure time of each truck on and from cross-dock, respectively, the time at which each truck will be processed on dock door. A good scheduling of trucks to dock doors will cause a good flowing of products. Therefore, the makespan which is the total operational time span from the start of unloading the first incoming truck until end of loading the last shipping truck will be shortened and congestion inside the facility will be minimized. The objective of this optimization problem is to minimize the makespan. For more details, we refer the reader to e.g., Agustina et al. (2010); Van Belle et al. (2012).

Generally, in the standard CDAP, origin and destination of trucks are set, that is, the trucks coming from the same origin or serving the same destination are assigned to the same inbound/outbound dock door respectively while dock door capacity is still imposed. In the case of the Truck Scheduling Problem, the trucks coming from the same origin or serving the same destination can be assigned to different dock doors if the assignment minimizes the makespan.

The Truck Scheduling Problem considers a deterministic environment in which all data are certain and reliable. That is, no condition can cause delay of trucks from arriving on time. Thus, arrival and departure time of each truck are fixed. Nevertheless, in realistic environment, the arrival or departure time of truck can be delayed due to several conditions such that traffic congestion, truck failure, road accidents, the weather, etc. Those life conditions make arrival time of truck to be uncertain. In Ladier and Alpan (2016b), the

authors have proposed a *Robust Truck-to-dock Door Scheduling Problem*. For this variant of trucks scheduling, the authors have considered realistic conditions in the formulation of the problem taking the trucks arrival times as a decision variable. The authors concluded that the mathematical formulation of the Robust Truck-to-dock Door Scheduling Problem remains feasible and stable even though it faces those disruptions.

Some combinatorial optimization problems combine Trucks scheduling and CDAP to benefit advantages of both optimization problems. The purpose is to determine when and at which dock door each incoming and outgoing truck will be handled in order to ensure the quick turnover and on-time deliveries. Therefore, a new combinatorial optimization problem referred to *Truck-to-dock Door Scheduling Problem* is raised. In Miao, Lim and Ma (2009), the authors have considered a combined CDAP with operational time constraint within cross-dock. In this problem, they considered also the time windows of arrival and departure of a truck, operational time for a cargo and the capacity of cross-dock. The cargo whose shipment is not fulfilled in current cross-dock could be handled in another cross-dock where treatment capacity is available. The goal is to minimize the operational time of the cargoes to ship and the number of cargoes not shipped. The cost is the total operation time of cargoes and the total penalty incurred. The penalty cost occurs when a cargo misses the shipping truck or its shipping truck is not yet docked. Other works about CDAP combined with Truck Scheduling can bee viewed in e.g., Ting and Rodríguez López (2012). For literature review and relevant works related to the scheduling of trucks on dock doors of cross-dock, we refer interested readers to e.g., Boysen and Fliedner (2010); Buijs et al. (2014); Gelareh et al. (2016); Lim et al. (2006); Miao et al. (2014); Miao, Lim and Ma (2009); Shakeri et al. (2012).

3.5.3 Trucks Sequencing Problem

Unlike Truck-to-dock Door Assignment and Truck Scheduling Problems, Truck sequencing problem does not take into account neither space dimension nor temporal dimension. In fact, having the number of incoming and outgoing trucks greater enough than the number of inbound and outbound dock doors, only a part of incoming and outgoing trucks can be assigned and get unloaded and loaded simultaneously while excess incoming and outgoing trucks are put-away in waiting buffer until the previous assigned ones end up getting unloaded/loaded. Therefore, while waiting, all those trucks are sequenced into queue to influence the efficiency of the cross-dock. For instance, a product in incoming

truck which is in waiting queue while the corresponding outgoing truck of this product is assigned on dock door ready to be loaded and can not be replaced by another outgoing truck, this kind of situations leads to a delay which affects the performance of the cross-dock. The Truck Sequencing Problem seeks to determine an optimal sequence or the order into which the waiting trucks are processed to dock doors and determines on which dock door exactly each truck is processed to improve cross-dock efficiency. That is, a truck is not assigned to a specific dock door but to any available dock dock according to the established sequence. For more details, we refer the interested reader to e.g., Ladier and Alpan (2016a).

3.5.4 Transshipment Problem

This optimization problem looks for determining a good flow of products, on the one hand, between suppliers and cross-dock and on the other hand, between cross-dock and customers. This involves a good and tight synchronization between incoming and outgoing trucks. The purpose is to minimize the transportation, the penalty and the inventory costs in distribution network. The transshipment problem considers also the decisions made by operations' managers on how goods are moved inside the facility like if incoming goods can be consolidated in staging area with some goods present in temporary storage to complete the freight, see e.g., Alpan et al. (2011). In Miao, Yang and Fu (2009), the authors have studied the case of transshipment problem considering the fixed transportation schedules, that is, arrival and departure time for transportation schedules are fixed. On the one hand, the suppliers ship the packed products towards cross-docks through fixed transportation schedules, the cargoes can delay into facility for consolidation and the related inventory holding cost is applied as a penalty, the cargoes that will need inventory service at the last time point of its time horizon will incur the high inventory penalty cost. On the other hand, the customers receive their cargoes from cross-dock. In transshipment-inventory models, a frequent assumption is that a demand which cannot be fulfilled by one supply point can be completed through some other supply points. The goal is then to evaluate a control policy for replenishment. The total cost denotes transportation cost from suppliers to cross-docks, transportation cost from cross-docks to customers, the inventory cost and the penalty cost.

3.5.5 Cross-dock Congestion Problem

Beside all those combinatorial optimization problems that arise within cross-docking, most of studies in literature review seemed to ignore the congestion issue of material handling devices inside the cross-docking terminal. However, this congestion is still a relevant problem and should be considered if we seek to manage the movement of material handling devices inside the cross-dock. In agreement with Shuib and Fatthi (2012), as the amount of incoming products entering into cross-dock increases, knowing in addition that the products have to be quickly handled inside the terminal in order to be immediately shipped, accordingly, the number of material handling devices must be multiplied and the speed up of movement of those handling devices is required to meet to goal of cross-docking strategy. This may create a congestion within the terminal. In Tsui and Chang (1990, 1992), the authors stated that the longer it takes to empty an incoming truck, the more material handling devices are required and the more congested the dock door will be. Due to this, the congestion and interference of pallets handling devices inside the terminal increase. In Ladier and Alpan (2016a), the authors have stated that looking for minimizing total traveled distance inside cross-dock leads to group unloading and loading goods in the same area that generates the congestion and slowdowns the overall process. Accordingly, within the cross-dock, seeking to minimize the congestion is in conflict with minimizing distance and vice versa.

In Yang et al. (2010), the auhors have performed a simulation in which they showed that a cross-dock with staging area at each inbound and outbound dock door reduce the dock door congestion inside terminal. The simulation showed that **direct** unloading/loading increases queuing of forklifts and congestion both at inbound and outbound sides of cross-dock while **indirect** unloading/loading reduces queuing and congestion and produces a higher throughput of pallets shipped from receiving side to shipping side but it requires a higher labor time to handle completely a pallet. Direct unloading/loading is applicable for pure cross-docking where unloaded pallets are directly sent to outbound dock door to be loaded into shipping trucks without going through temporary storage area. For a cross-dock with indirect unloading/loading, each inbound and outbound dock door have a staging area. Pallets are unloaded from incoming trucks to a staging area of inbound dock door and then are transferred from inbound staging area to staging area of outbound dock door, and finally from staging area of outbound dock door to be loaded into shipping trucks. Therefore, a pallet is picked up three times which consumes lot of

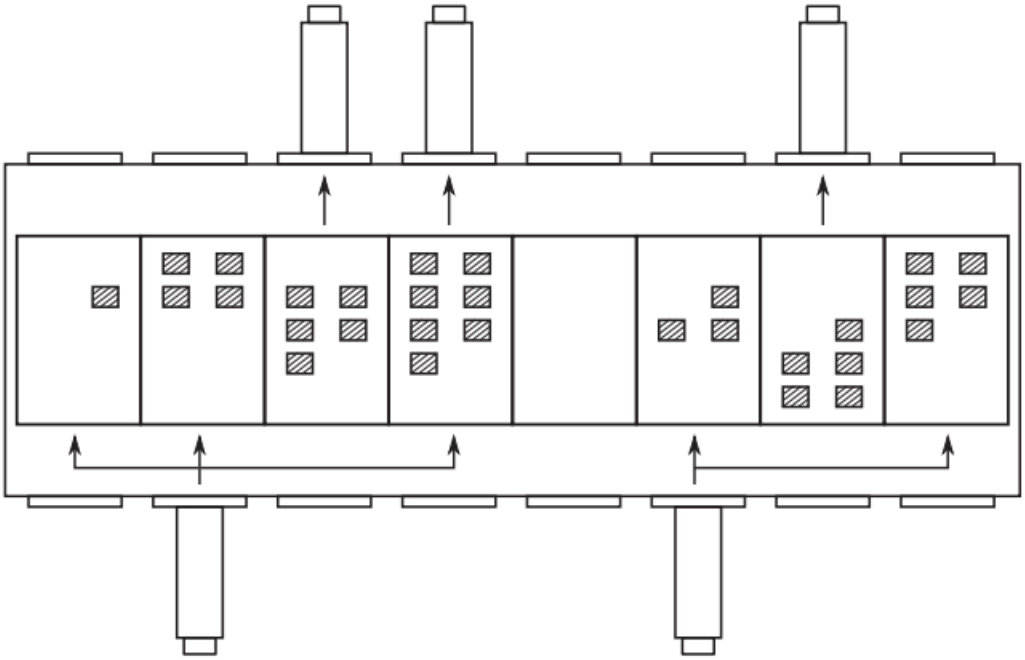


FIGURE 3.8 – *Single-stage* or *two-touch* cross-dock, Gue and Kang (2001)

space and time, increases labor cost and possibility of damaging a pallet.

In Gue and Kang (2001); Yang et al. (2010); Van Belle et al. (2012), the authors have defined cross-dock according to the number of touches a pallet can undergo inside the cross-dock. A *one-touch* cross-dock corresponds to direct unloading/loading (pure cross-docking). For a *two-touch* or *single-stage* cross-dock, products are received either into a staging area of inbound dock door or into that of outbound dock door. The Figure 3.8 depicts the two-touch cross-dock. The *multiple-touch* or *two-stage* corresponds to indirect unloading/loading, see e.g., Figure 3.9. In two-touch and multiple touch (single-stage and two-stage) the pallet is handled one more time and more floor space inside cross-dock is needed. This implies a larger cross-dock which increases the weighted distance traveled by handling devices. The interested readers on a single-stage and two-stage cross-docks can refer to e.g., Gue and Kang (2001) about the queuing of cross-dock's entities.

3.5.6 Vehicle Routing Problem with Cross-docking

In Supply Chain, goods are picked from various suppliers or other freight yards and are loaded into trucks and moved towards the distribution centers. In this case, the distribution center is referred to the cross-dock. Those goods have to be shipped to multiple destinations after having been sorted according to their respective destinations and sometimes consolidated into cross-dock. That is, the cross-dock is considered to be the

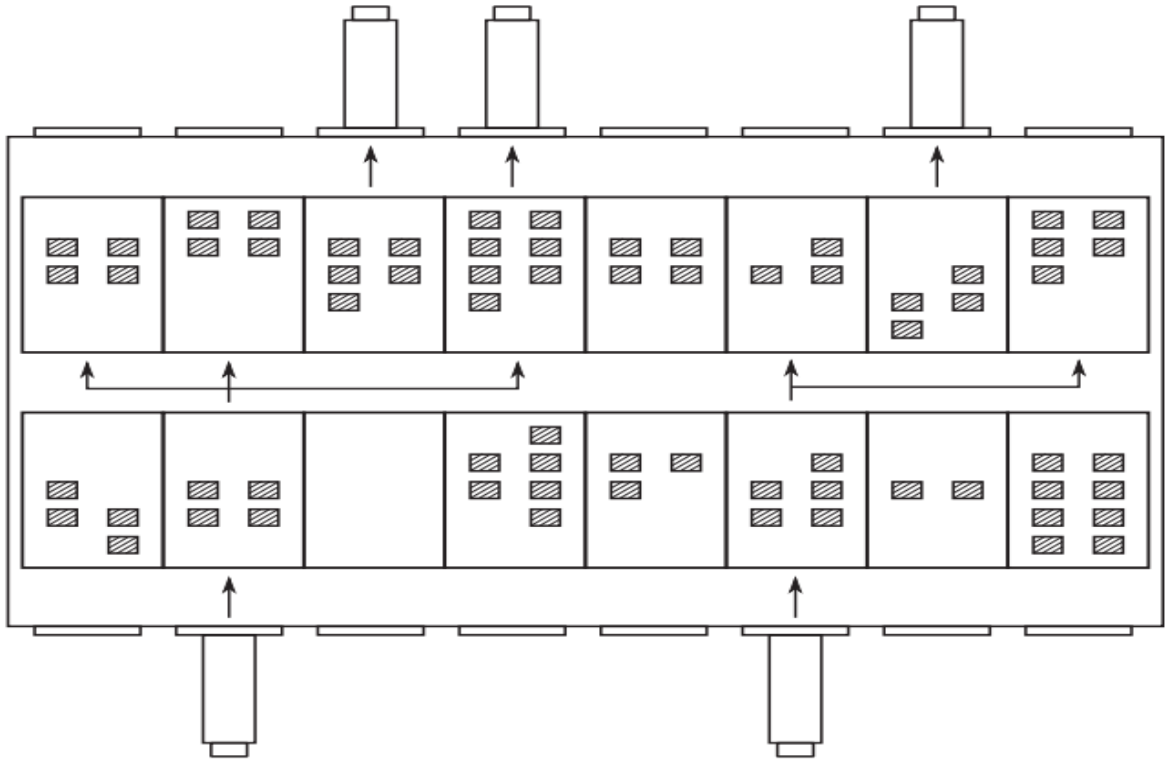


FIGURE 3.9 – A *two-stage* or *multiple-touch* cross-dock, Gue and Kang (2001)

departure and arrival node of all vehicles involved in the defined transportation network. In a given transportation network, the process of products picking from a defined node and products delivering to the customers in that transportation network is known under Vehicle Routing Problem (VRP). The goal of VRP is to determine the number of vehicles to use in that transportation network, the optimal path that each used vehicle have to follow from suppliers to customers through cross-dock node.

An efficient optimization for the Vehicle Routing Problem may increase the throughput of the cross-dock. In Wen et al. (2009), the authors proposed a vehicle routing problem with cross-docking using homogeneous vehicles for transportation of customers' orders via a cross-dock as a node. The objective was to minimize the travelling time of trucks taking into account the time window of each truck. The authors proposed a Mixed Integer Programming (MIP) model and a Tabu Search heuristic to solve the problem. The interested readers can also refer to Birim (2016).

3.6 Cross-dock Door Assignment Problem

The NP-hard combinatorial optimization problem we tackle in this PhD thesis is an optimization problem referred to "Cross-dock Door Assignment Problem (CDAP)". Truck scheduling, truck sequencing or transshipment problems are also concerned by the assignment of trucks to dock doors of the facility taking into account other factors like for instance the time window. In this dissertation, we focus more precisely on the optimization of the activity related to the cross-dock itself while supposing that some external elements have been decided previously. The corresponding problem is the Cross-dock Door Assignment Problem whose the standard quadratic mathematical formulation has been introduced in Zhu et al. (2009).

3.6.1 Introduction

Cross-dock Door Assignment Problem looks for to optimize allocation of dock doors of cross-dock taking into account several factors, namely the shape, the flow of volume of pallets from incoming to outgoing trucks, those pallets cross the cross-dock from inbound to outbound dock doors. We start our study by tackling this optimization problem in general terms and further we describe different variants of CDAP as they have been proposed in Tsui and Chang (1990, 1992); Cohen and Keren (2008, 2009); Zhu et al. (2009); Tarhini et al. (2016). Afterwards we focus on the variants of this problem proposed in Zhu et al. (2009) . We choose this variant because it takes into account more real application such that the management of the capacity of dock doors. This optimization problem has also been studied in Guignard et al. (2012); Nassief et al. (2016)

The authors in Tsui and Chang (1990, 1992), have been the first to be interested and worked on the Cross-dock Door Assignment Problem. In Tsui and Chang (1990), the authors have proposed a basic bi-linear programming model for assigning receiving and shipping trucks to respectively inbound and outbound dock doors considering only assignment constraints both for dock doors and trucks. That is, according to the mathematical formulation, each inbound (respectively outbound) dock door is assigned to a single receiving (respectively shipping) truck and each receiving (respectively shipping) truck is assigned to a single inbound (respectively outbound) dock door. This assumes that the number of receiving (respectively shipping) trucks is equal to the number of inbound (respectively outbound) dock doors. In Tsui and Chang (1992), the authors improved their

former solution proposing a new solution based on Branch and Bound (B&B) algorithm. In Bermudez and Cole (2001), the authors have developed a Genetic Algorithm (GA) to solve the basic mathematical model introduced by Tsui and Chang (1990) in a breakbulk terminal to minimize the total weighted distance traveled inside the terminal composed by inbound and outbound dock doors and an open dock door. In Cohen and Keren (2008, 2009), the authors studied another variant of Cross-dock Door Assignment Problem. They extended the formulation proposed in Tsui and Chang (1990) by relaxing the assignment constraints between outbound dock doors and outgoing trucks. That is, the authors proposed a new mathematical formulation that allows several outbound dock doors to serve a single destination. In that new mathematical model, the capacity of each outbound dock door is set to be equal the capacity of a truck. At inbound side, they kept assignment constraint as in Tsui and Chang (1990). Thus, the authors defined a new parameter for truck capacity, considering that tall trucks have the same capacity, and an additional decision variable. For a broad explanation, after a truck is unloaded, the freight is sent to corresponding destination at load dock door (outbound dock door). If the total freight sent to that destination exceeds the capacity of truck, the amount of that freight is split into several shipping dock doors that will be reserved to allocate that destination. We let recall that in the author's formulation, the capacity of each outbound dock door is equal the capacity of an outgoing truck.

In Tarhini et al. (2016), the authors proposed a slight change in the mathematical formulation of Tsui and Chang (1990). In fact, Tarhini et al. (2016) include in their formulation an additional decision variable that controls the assignment of trucks on dock doors. That is, the objective function value will be distorted if any incoming truck was assigned to outbound dock door. As the problem is still hard for large instances, the authors proposed an evolutionary Scatter Search (SS) algorithm based on Genetic Algorithm metaheuristic to minimize the total traveled distance.

In Küçükoğlu and Öztürk (2017), the authors proposed a new variant of combinatorial optimization problem named "*Truck-to-Door Assignment and Product placement Problem in Cross-Dock (TDAPP-CD)*". In that study, the authors considered a Truck-to-dock Door Assignment Problem in cross-docking center with a temporary storage area. The study takes into account moving steps of products according to the product flow path: "**inbound dock doors → temporary storage area → outbound dock doors**". The authors proposed a Mixed Integer Programming model to minimize the total distance traveled by the product on the path.

Taking into account the increasing of amount of freights to handle on cross-dock yard, it is not practical for a dock door to allocate a single origin, even less to reserve several dock doors for a single destination. Accordingly, the configuration in Tsui and Chang (1990, 1992) and those in Cohen and Keren (2008, 2009); Tarhini et al. (2016) are not suitable for realistic applications unlike the configuration considered in Zhu et al. (2009) where the authors have proposed a generalized mathematical formulation that takes into account more realistic considerations. That is, in Zhu et al. (2009) mathematical formulation, the authors take into consideration the capacity of each dock door and the fact that each origin (respectively destination) must be assigned to one dock door. That is, each inbound (respectively outbound) dock door can allocate more than one origin(respectively destination) and each origin (respectively destination) is assigned to one and only one inbound (respectively outbound) dock door. Below are the considered outlines for the generalized mathematical formulation for Zhu et al. (2009) for the Cross-dock Door Assignment Problem:

- Incoming (respectively shipping) trucks are aggregated into origins (respectively destinations), i.e., from a given origin can come more than one truck and a given destination can be served by more than one truck.
- Assignment of origins and destinations instead of assignment of trucks.
- Each dock door can allocate more than one origin (respectively destination) as long as the capacity of dock door is not exceeded.

From the two side of an I-shape cross-dock, the resulting mathematical model can be seen as a bi-Generalized Assignment Problem (GAP) if we consider separately one side, see e.g.,Guignard et al. (2012).

Due to the nature of the CDAP such that the large amount of freights to handle and the dynamic nature of freight flow patterns which increases the number of material handling devices to use for transferring pallets from inbound to outbound dock doors, increasing and arrangement of number of dock doors and assignment of trucks to dock doors, the Cross-dock Door Assignment Problem is a known NP-hard combinatorial optimization problem. In addition, as seen above, the variant of Zhu et al. (2009) that we deal with all along this dissertation includes the Generalized Assignment Problem as a sub-problem. As GAP is an NP-hard combinatorial optimization problem, see e.g., Ross and Soland (1975); Sahni and Gonzalez (1976), the CDAP is also NP-hard.

3.6.2 Background for CDAP and Literature review

A key difference between a classical warehouse and a cross-docking warehouse is that, unlike warehouses where products remain (sometimes for long duration) until they are ordered by final customers, the products handled by cross-dock are not permitted to remain on the platform beyond 24 hours, see e.g., Guignard et al. (2012), sometimes are required to be transferred within less than an hour, see e.g., Bartholdi and Gue (2004). As explicitly detailed in section 3.4, three classes of cross-docking problems can be summarized as follows : strategic problems determine a good location for the cross-dock and its layout design ; operational problems determine the best assignment of truck to dock door, locations where goods will be temporarily stored, the best synchronization between arriving and departing trucks at the dock doors of the cross-dock etc. ; and tactical problems determine the flow of products through the cross-dock to minimize costs and make supply meet demand. For a broad literature reviews in relation of other variants of cross-docking problems, we refer the interested readers to Boysen and Fliedner (2010); Bellanger et al. (2013); Bodnar et al. (2015); Hermel et al. (2016); Küçükoğlu (2016); Küçükoğlu and Öztürk (2017).

All along this dissertation, we deal with the variant of Cross-dock Door Assignment Problem in which a set of incoming trucks called origins come from various sources of products such as suppliers, manufacturers, factory, warehouses, other cross-docks, etc., and unload their pallets of products on a set of inbound dock doors, at which point unloaded pallets are sorted in a staging area based on their destinations. Finally, the pallets are directly transferred within the cross-dock using material handling devices such as forklifts to a set of outbound dock doors where they are or not consolidated with other products going on the same destination and loaded onto outgoing trucks called destinations. The goal of the Cross-dock Door Assignment Problem is to find the best assignment of origins (origin trucks) to inbound dock doors and destinations (destination trucks) to outbound dock doors so that the total cost of transporting pallets from inbound dock doors to outbound dock doors within the platform is minimized while it keeps satisfying a set of constraints. As already mentioned, the transportation cost is considered as the total weighted distance traveled inside the cross-dock by the used material handling devices.

The problems of truck-to-dock door assignment assume that there are enough dock doors to accommodate all incoming and outgoing trucks, that is, each truck may be assigned to dock door, therefore, for these problems time aspects are not taken into

(i)	*(ii)*	*(iii)*
Tsui and Chang (1990) Tsui and Chang (1992) Tarhini et al. (2016)	Cohen and Keren (2008) Cohen and Keren (2009)	Zhu et al. (2009) Guignard et al. (2012) Nassief et al. (2016) Nassief (2017)

TABLE 3.1 – The Cross-dock Door Assignment Problems, dock doors assignment strategy

account.

The Cross-dock Door Assignment Problems may be classified according to several criteria. The first criterion is based on the dock doors allocation strategy. Several types of allocation restrictions are possible : The cited papers into column *(i)* of Table 3.1 deal with the variant of CDAP where each dock door must serve only one origin(respectively destination) and each origin(respectively destination) must be assigned to only one inbound(respectively outbound) dock door ; the papers cited into column *(ii)* of Table 3.1 deal with the variant of the CDAP where each inbound dock door serves only one origin at a time, but the same destination may be assigned to several outbound dock doors ; and the variant of CDAP dealt with into the works in column *(iii)* of Table 3.1 consider a generalized case where each inbound (respectively outbound) dock door may serve more than one origin (respectively destination).

The second criterion considers whether and how capacity constraints are taken into account : In Tsui and Chang (1990, 1992) and Tarhini et al. (2016), there are no limitations of capacities on the inbound and outbound dock doors ; In Cohen and Keren (2008, 2009) there are no limitations of capacities on inbound dock doors but only capacities of outbound dock doors are taken into account and dock door capacity is considered equal to the capacity of a truck ; In Zhu et al. (2009), the authors extended the model proposed by Tsui and Chang (1990) considering capacity constraints on both inbound and outbound dock doors in order to take into account more realistic considerations. This what makes Zhu et al. (2009) model to be standard or generalized.

The third criterion is based on the layout design of a cross-dock as the specification of dock doors as either inbound or outbound dock doors. The so-called I-shape for cross-dock layout design is one of the most often considered shape in the literature, see e.g., Tsui and Chang (1990, 1992); Gue (1999); Bartholdi III and Gue (2000); Oh et al. (2006); Cohen and Keren (2008, 2009). Figure 3.10 taken from our published paper in Gelareh

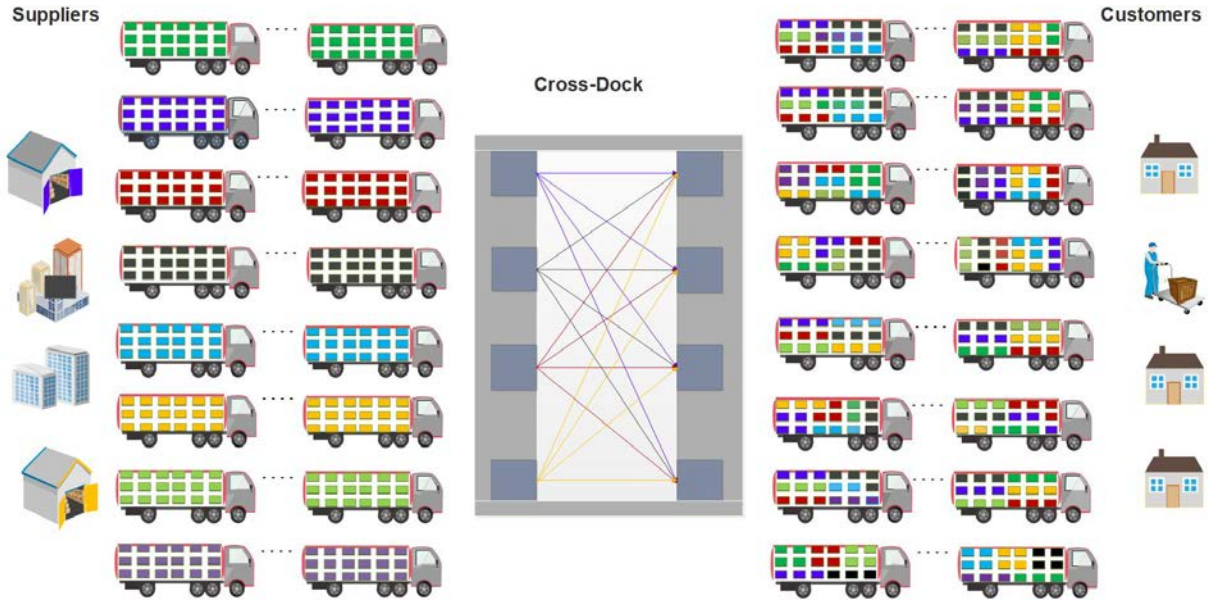


FIGURE 3.10 – Cross-dock : Exclusive mode of dock door, Gelareh et al. (2020)

et al. (2020) describes the I-shape cross-docking operations in greater detail. An I-shaped cross-dock has a rectangular shape, with receiving dock doors on one side and outbound dock doors on the other side. Therefore, rectilinear distances may accurately simulate distances traversed by the forklifts following clearly marked lanes, see Figure 3.10. Other cross-dock shapes considered in the Cross-dock Door Assignment Problems are so-called semi-permanent and dynamic layouts, see e.g., Brown (2003); Bozer and J. Carlo (2008); Yu et al. (2008). For other shapes of a cross-dock layout considered in the cross-dock literature we refer the reader to e.g., Bartholdi and Gue (2004). In this PhD thesis we deal with the CDAP where each dock door may serve more than one origin(respectively destination), capacity constraints are imposed on each dock door and I-shape cross-docking operations are allowed. This variant of the problem was introduced in Zhu et al. (2009) as a standard formulation of CDAP and as an extension of the basic formulation of Tsui and Chang (1990). In Guignard et al. (2012), the authors subsequently used the model of Zhu et al. (2009) to develop three heuristics, the first two are based on local search and the third is based on Convex Hull Relaxation (CHR). Recently, in Nassief et al. (2016), the authors proposed a Mixed Integer Programming formulation of the CDAP which consists to determine optimal paths for commodities from origins to destinations via inbound and outbound dock doors. In that same work, the authors proposed some valid inequalities for the problem as well as a Lagrangian Relaxation heuristic to tackle large-scale instances. In Nassief et al. (2018), the authors presented a study on the standard CDAP as defined in Zhu et al. (2009) with and without load and unload times. They proposed several new

formulations and a branch and price solution strategy.

The computational experimentation shows that CDAP is extremely difficult to solve to optimality. As already introduced, the CDAP includes the Generalized Assignment Problem (GAP) as a sub-problem and the GAP problem is NP-hard. The GAP is a well-established field of research in terms of both modeling and solution approaches, and has been extensively studied in papers such as e.g., Chu and Beasley (1997); Wilson (1997); Dıaz and Fernández (2001); Lorena et al. (2002); Yagiura, Ibaraki and Glover (2004); Yagiura et al. (2006); Jeet and Kutanoglu (2007); Woodcock and Wilson (2010); Liu and Wang (2015). In addition, several variants of the GAP have been proposed in the literature including the Multi-Resource GAP, see e.g., Yagiura, Iwasaki, Ibaraki and Glover (2004), the multi-level GAP, see e.g., Laguna et al. (1995), the generalized quadratic assignment problem, see e.g., Cordeau et al. (2006); Mateus et al. (2011); McKendall and Li (2017); Pessoa et al. (2010), the generalized assignment problem with special ordered sets, see e.g., French and Wilson (2007) and the quadratic three-dimensional assignment problem, see e.g., Hahn, Kim, Stuetzle, Kanthak, Hightower, Samra, Ding and Guignard (2008). In Zhu et al. (2009), the authors establish a relationship between the Generalized Quadratic three-dimensional Assignment Problem (GQ3AP) and the CDAP and show that the CDAP can be solved as a GQ3AP.

3.6.3 Mathematical quadratic formulation for CDAP

The Cross-dock Door Assignment Problem can be described mathematically as follows. On left side we are having a set M of incoming trucks named origins that have to be docked and unloaded at a set I of inbound dock doors and at right side we are having a set N of outgoing trucks named destinations to be docked to a set J of outbound dock doors. The total amount of pallets delivered to each origin $m \in M$ is $s_m > 0$, s_m is dispatched into small amounts of flows $f_{m,n}$ such that each flow $f_{m,n} \geq 0$ is destined to destination $n \in N$. Therefore, the total amounts of flows coming from origin $m \in M$ and received by different destinations can be computed as $s_m = \sum_{n \in N} f_{m,n}, \forall m \in M$. In the same way, on the outbound dock doors side, each destination $n \in N$ receives total amount of flows $r_n > 0$. This amount r_n is defined as the total amount of flows coming from different origins and destined to $n \in N$. Accordingly, r_n is computed as $r_n = \sum_{m \in M} f_{m,n}, \forall n \in N$. The statements $s_m > 0$ and $r_n > 0$ means that no origin can enter to the facility empty nor no destination can leave facility empty. The capacity of each inbound dock door $i \in I$

is denoted S_i, this parameter represents the total number of origins with a quantity of load s_m for each origin that inbound dock door i can allocate. Likewise, the capacity of each outbound dock door $j \in J$ is denoted R_j, it means the total number of destinations with quantity of load r_n each destination that outbound dock door j is able to allocate.

For an I-shape cross-dock configuration, it is supposed that all dock doors have the same capacity, that is, $\forall i \in I,\ j \in J,\ S_i = R_j$. This variant of the CDAP we deal with in this PhD thesis considers that each subset $M_i : M_i \subset M$ which denotes the subset of al origins assigned to inbound dock door $i \in I$ and likewise each subset $N_j : N_j \subset N$ which denotes the subset of all destinations assigned to outbound dock door $j \in J$ are handled the same time. From above description of the optimization problem, we will use the following notations in the rest of this thesis:

Sets

- M : Set of origins referring to incoming trucks
- N : Set of destinations referring to outgoing trucks
- I : Set of inbound (strip) dock doors
- J : Set of outbound (stack) dock doors

Parameters

- s_m : Available number of pallets from origin $m \in M$
- r_n : Number of pallets destined to destination $n \in N$ from each origin $m \in M$
- S_i : Capacity of inbound dock door $i \in I$
- R_j : Capacity of outbound dock door $j \in J$
- $d_{i,j}$: Distance between inbound and outbound dock doors pair (i, j)
- $f_{m,n}$: Number of pallets flow from origin $m \in M$ destined to destination $n \in N$

$f_{m,n}$ is considered as the number of trips required by the material handling device to move pallets originating from origin $m \in M$ assigned to dock door $i \in I$ to destination $n \in N$ assigned to dock door $j \in J$.

Decision variables

In order to formally model this combinatorial optimization problem, two binary decision variables are defined. On the left side of I-shape cross-dock a decision variable $x_{m,i}$ for managing assignment of the set M of origins to the set I of inbound dock door and on the right side of I-shape cross-dock a decision variable $y_{n,j}$ for managing assignment of the set N of destinations to the set J of outbound door.

$$\forall m \in M,\ i \in I,\ x_{m,i} = \begin{cases} 1 & \text{if origin } m \text{ is assigned to inbound dock door } i \\ 0 & \text{otherwise} \end{cases}$$

$$\forall n \in N,\ j \in J,\ y_{n,j} = \begin{cases} 1 & \text{if destination } n \text{ is assigned to outbound dock door } j \\ 0 & \text{otherwise} \end{cases}$$

The real model considered in thesis assumes the following configurations :

- Exclusive mode
- I-shape cross-dock
- More vehicles than dock doors (in both sides !), i.e, $|M| > |I|$ and $|N| > |J|$
- A same count of vehicles input/output $|M| = |N|$
- A same count of dock doors input/output i.e., $|I| = |J|$
- No preemption
- No interchangeability - each unit is assigned to a destination

Let $\Omega =< m-i-j-n >$ be the flow path to move the amount $f_{m,n}$ from origin $m \in M$ to destination $n \in N$ through inbound and outbound dock door $i \in I$ and outbound dock door $j \in J$. The flow path Ω will be established if both of the following conditions are fulfilled at the same time: origin $m \in M$ is assigned to inbound dock door $i \in I$, i.e., $x_{m,i} = 1$ **and** destination $n \in N$ is assigned to outbound dock door $j \in J$, i.e., $y_{n,j} = 1$. When the flow path is established, the used material handling device starts routing the flow $f_{m,n}$ from origin $m \in M$ to destination $n \in N$ through inbound and outbound dock doors pair $(i, j) :\ \forall i \in I, j \in J$.

According to the above sets, parameters and decision variables, the CDAP may be

formally modeled as the following quadratic model Q as proposed in Zhu et al. (2009)

$$
(Q)\begin{cases}
min\ f(x,y) = & \displaystyle\sum_{m\in M}\sum_{i\in I}\sum_{n\in N}\sum_{j\in J} f_{m,n}d_{i,j}x_{m,i}y_{n,j} & & \text{(3.1a)}\\
\text{subject to :} & & & \\
\displaystyle\sum_{i\in I} x_{m,i} = 1, & & \forall m\in M & \text{(3.1b)}\\
\displaystyle\sum_{j\in J} y_{n,j} = 1, & & \forall n\in N & \text{(3.1c)}\\
\displaystyle\sum_{m\in M} s_m x_{m,i} \le S_i, & & \forall i\in I & \text{(3.1d)}\\
\displaystyle\sum_{n\in N} r_n y_{n,j} \le R_j, & & \forall j\in J & \text{(3.1e)}\\
x_{m,i},\ y_{n,j} \in \{0,1\}, & & \forall m\in M, n\in N, i\in I, j\in J & \text{(3.1f)}
\end{cases}
$$

The quadratic interactivity (transportation cost) which denotes the weighted distance by moving flow $f_{m,n}$ along distance $d_{i,j}$ between inbound dock door $i \in I$ and outbound dock door $j \in J$ from origin $m \in M$, when m is assigned to inbound dock door i, to destination $n \in N$ when n is assigned to outbound dock door j is computed as $f_{m,n} \times d_{i,j}$, i.e., the objective function (3.1a) sums up the total weighted distance traveled from all inbound dock doors $|I|$ to all outbound dock doors $|J|$ according to all distribution flows $f_{m,n}$ between all origins and all destinations. In the quadratic model $\mathcal{Q}$, the number of constraints is computed as follows, $|M| + |N| + |I| + |J|$ and the number of variables is computed as follows, $|M||I| + |N||J|$.

In the quadratic mathematical model Q, objective function (3.1a) minimizes the total transportation cost inside the cross-dock taking into account the constraints (3.1b)-(3.1f) classified into three groups. The two sets of multiple choice constraints (3.1b) and (3.1c) ensure that each origin (respectively destination) must be allocated to one and only one inbound (respectively outbound) dock door, respectively. We group (3.1b) and (3.1c) into a set of assignment constraints *(i)*. The two sets constraints (3.1d) and (3.1e) guarantee that the capacity of each inbound (respectively outbound) dock door is respected. We group (3.1d) and (3.1e) into a set of the capacity constraints or knapsack constraints *(ii)*. And the last constraint (3.1f) imposes the binary requirement on the decision variables *(iii)*.

3.6.4 Some variants of the CDAP

We recall that the combinatorial optimization problem we are dealing with in this dissertation is related to assignment of trucks to dock doors of an I-shape cross-dock that uses an exclusive operating mode of dock doors. The standard quadratic formulation detailed above is originally proposed in Zhu et al. (2009). This mathematical formulation refers to real application where any inbound (respectively outbound) dock door is constrained to allocate more than one origin (respectively destination) respecting the dock door capacity, see e.g., the set of constraints *(ii)* and each origin (respectively destination) is assigned to one and only one dock door, see e.g., the set of constraints *(i)*. In the following of this section, we present some variants of the CDAP (summary in Table 3.2) and we raise some of their limitations as regards to their real applications.

Variants of CDAP	**Summry of related mathematical formulation**	**Related remarks and limitations**
Zhu et al. (2009)	Each dock door (inbound/outbound) can allocate more than one origin/destination managing the capacity of dock door	This is the standard or generalized variant of CDAP. This varant takes into account the amount of resources available, namely capacity of dock doors
Tsui and Chang (1990, 1992)	In this variant of CDAP, each dock door is assigned one truck and dock door capacity is not considered	The related model can not be applied in real applications given a lot of incoming/outgoing trucks exceeding number of dock doors
Tarhini et al. (2016)	Same as Tsui and Chang (1990, 1992), just a slight change of assignment of incoming trucks	Same limitations as Tsui and Chang (1990, 1992)
Cohen and Keren (2008, 2009)	At inbound side, the same as Tsui and Chang (1990, 1992), at outbound side, the feasible solution requires number of dock doors to exceed much enough number of outgoing trucks	This variant is the worst given that it requires additional dock doors than in Tsui and Chang (1990, 1992), the resources (outbound dock doors) are mismanaged

TABLE 3.2 – The variants of Cross-dock Door Assignment Problem (CDAP)

3.6.4.1 Variant of Tsui and Chang

The proposed mathematical formulation by Tsui and Chang (1990, 1992) seems not to be feasible in real applications. In fact, the formulation supposes that each inbound (respectively outbound) dock door allocates only one incoming (respectively outgoing) truck and that each incoming (respectively outgoing) truck is assigned to one inbound (respectively outbound) dock door. Only assignment constraints (i) are considered for both dock doors and trucks. The dock doors capacity is not taken into account, that makes this formulation unusable for real applications given the very large number of trucks to be assigned to dock doors which far exceeds the number of dock doors.

Tsui and Chang Model:

$$
Q^{t\&c}\begin{cases}
min\ f(x,y) = & \displaystyle\sum_{m\in M}\sum_{i\in I}\sum_{n\in N}\sum_{j\in J} f_{m,n}d_{i,j}x_{m,i}y_{n,j} & \text{(3.2a)}\\
\text{subject to :} & & \\
\displaystyle\sum_{i\in I} x_{m,i} = 1, & \forall m \in M & \text{(3.2b)}\\
\displaystyle\sum_{j\in J} y_{n,j} = 1, & \forall n \in N & \text{(3.2c)}\\
\displaystyle\sum_{m\in M} x_{m,i} = 1, & \forall i \in I & \text{(3.2d)}\\
\displaystyle\sum_{n\in N} y_{n,j} = 1, & \forall j \in J & \text{(3.2e)}\\
x_{m,i},\ y_{n,j} \in \{0,1\}, & \forall m \in M, n \in N, i \in I, j \in J & \text{(3.2f)}
\end{cases}
$$

The objective function (3.2a) and the assignment constraints (3.2b) and (3.2c) are similar to the objective function (3.1a) and assignment constraints (3.1b) and (3.1c) of previous standard model in Zhu et al. (2009). The models $Q^{T\&C}$ of Tsui and Chang (1990, 1992) and Q differ from each other to the constraints (3.2d) and (3.2e) which guarantee that each inbound (respectively outbound) dock door have to allocate a single origin (respectively destination).

According to that mathematical formulation of Tsui and Chang, the model requires on left side the number of origins $|M|$ to be equal to the number of inbound dock doors $|I|$ and likewise, on the right side, the number of destinations $|N|$ to be equal to the number of outbound dock doors $|J|$. Those requirements are due to the pairs of constraints (3.2b) and (3.2d) that handle trucks to dock doors on left side and (3.2c) and (3.2e) on right side of I-shape cross-dock. This problem configuration makes the Tsui and Chang (1990, 1992)

mathematical model unusable in real applications, for instance, in the case the number of inbound dock doors is greater that the number of incoming trucks, i.e., $|I| > |M|$ and/or the number of outbound dock doors is greater than the number of outgoing trucks, i.e., $|J| > |N|$, fictitious incoming and/or outgoing trucks with zero quantities will be created. A slight more realistic configuration of Tsui and Chang model can be proposed to avoid those fictitious incoming and outgoing trucks by modifying constraints (3.2d) and (3.2e) related to assignment of dock doors as follows:

$$\begin{cases} \sum_{m \in M} x_{m,i} \leq 1, & \forall i \in I \quad (3.3a) \\ \sum_{n \in N} y_{n,j} \leq 1, & \forall j \in J \quad (3.3b) \end{cases}$$

Those constraints (3.3a) and (3.3b) ensure that any inbound dock door $i \in I$ (respectively outbound dock door $j \in J$) can allocate one incoming truck $m \in M$ (respectively one outgoing truck $n \in N$) or not. For the reverse case, that is, the number of incoming trucks is greater than the number of inbound dock doors, i.e., $|M| > |I|$ and/or the number of outgoing trucks is greater than the number of outbound dock doors, i.e., $|N| > |J|$, the model $Q^{T\&C}$ will be infeasible.

3.6.4.2 Variant of Tarhini et al.

In Tarhini et al. (2016), the authors proposed a new formulation of Cross-dock Door Assignment Problem directly based on the model introduced in Tsui and Chang (1990). In the quadratic model, the authors defined an additional decision variable to control only the assignment of origins to dock doors. The authors' mathematical model is depicted as follows:

Additional decision variable:

$$z_{m,i} = \begin{cases} 1 & \text{if incoming truck } m \in M \text{ is assigned to inbound dock door } i \in I \\ MAX & \text{if incoming truck } m \text{ is assigned to any outbound dock door } j \in J \end{cases}$$

Tarhini et al. model :

$$Q^t : \begin{cases} minimize \quad \sum_{m \in M} \sum_{i \in I} \sum_{n \in N} \sum_{j \in J} f_{m,n} d_{i,j} x_{m,i} y_{n,j} z_{m,i} & (3.4a) \\ \text{subject to :} \\ \qquad (3.2b) - (3.2f) \\ \qquad z_{m,i} \in \{1, MAX\}, \quad \forall m \in M,\ i \in I & (3.4b) \end{cases}$$

The constraint (3.4b) is related to the additional integer variable. The variable will take value 1 for normal dock doors assignment, i.e., according to I-shape cross-docking facility, normal assignment imposes origins to be assigned to inbound dock doors and destinations to outbound dock doors. If any incoming truck is assigned to any outbound dock door which corresponds to abnormal assignment according to the authors in Tarhini et al. (2016), $z_{m,i}$ will take a highest integer value, MAX, that will distort the value of the final solution.

3.6.4.3 Variant of Cohen and Keren

As for Zhu et al. (2009), Cohen and Keren consider also the assignment of origins and destinations instead of trucks, see e.g., Cohen and Keren (2008, 2009). The authors take into account another criterion of assigning destinations to outbound dock doors as described below.

While each inbound dock door $i \in I$ allocates one origin at time as in Tsui and Chang (1990, 1992), on outbound side, a single destination $n \in N$ can be assigned to several loading dock doors $j, k, l \in J$. In fact, the freight flow $f_{m,n}$ destined to destination n is split and sent to load dock doors j, k, l to which n is assigned. It means that, in case the total freight flows $r_n = \sum_{m \in M} f_{m,n},\ \forall n \in N$ sent to destination n imposes several shipping trucks, the freight $f_{m,n}$ is split into several outbound dock doors $j, k, l \in J$ to which those shipping trucks serving destination n are docked. That is, the capacity of dock door is considered to be equal to the capacity of a truck. The authors considered that the capacity C_m of truck $m \in M$ must respect the following condition $\sum_{n \in N} f_{m,n} \leq C_m,\ \forall m \in M$ and that shipping and incoming trucks have the same capacity. Therefore $\nu = (\sum_{m \in M} f_{m,n})/C, \forall n \in N$ denotes the number of trucks serving the same destination n where C stands for the capacity of incoming (respectively shipping) truck. This number ν of trucks is equal to the number of load dock doors that are reserved to allocate the destination n. In that case, $f_{m,n}$ must be split and sent to ν load dock doors. The split is a certain percentage of $f_{m,n}$, however, all the splits of $f_{m,n}$ are not known in advance but are related to ν dock doors. In view of this fact, for any combination of $n \in N$ and $m \in M$ according to the flow $f_{m,n}$ between m and n, a new decision variable $\alpha_{m,i,j,n}$ is defined to handle how the percentages of $f_{m,n}$ are sent to ν outbound dock doors. Those percentages are such that for all inbound dock door $i \in I$ the percentages $\alpha_{m,i,j,n} \times f_{m,n},\ \alpha_{m,i,k,n} \times f_{m,n},\ \alpha_{m,i,l,n} \times f_{m,n},\ \ldots$ are sent from i to the shipping dock doors

$$j, k, l, \cdots \in J : j \neq k, k \neq l, j \neq l, \text{ etc.}$$

Additional parameters:

C : The capacity of a truck

D : Capacity of a dock door

Additional decision variable:

$\alpha_{m,i,j,n}$ is the portion of $f_{m,n}$ that was originating from m unloaded at inbound dock door i, and moved to shipping dock door j. All combinations having receiving dock door i that was not assigned to origin m have $\alpha_{m,i,j,n} = 0$. The same case is true for all combinations having shipping dock door j to which destination n is not assigned. However, if m is assigned to inbound dock door i and n assigned to outbound dock door j we have $0 \leq \alpha_{m,i,j,n} \leq 1$.

Cohen and Keren model:

$$
Q^{ck}\begin{cases}
min\ f(x,y,\alpha) = & \displaystyle\sum_{m\in M}\sum_{i\in I}\sum_{n\in N}\sum_{j\in J} d_{i,j} f_{m,n}\alpha_{m,i,j,n}x_{m,i}y_{n,j} \quad (3.5a)\\
\text{subject to :} & \\
\displaystyle\sum_{i\in I} x_{m,i} = 1, & \forall m \in M \quad (3.5b)\\
\displaystyle\sum_{j\in J} y_{n,j} = [(\sum_{m\in M} f_{m,n})/D], & \forall n \in N \quad (3.5c)\\
\displaystyle\sum_{i\in I}\sum_{j\in J} x_{m,i}y_{n,j}\alpha_{m,i,j,n} = 1, & \forall m \in M, \forall n \in N \quad (3.5d)\\
\displaystyle\sum_{m\in M} x_{m,i} \leq 1, & \forall i \in I \quad (3.5e)\\
\displaystyle\sum_{n\in N} y_{n,j} \leq 1, & \forall j \in J \quad (3.5f)\\
\displaystyle\sum_{i\in I}\sum_{m\in M}\sum_{n\in N} x_{m,i}f_{m,n}\alpha_{m,i,j,n}y_{n,j} \leq C, & \forall j \in J \quad (3.5g)\\
0 \leq \alpha_{m,i,j,n} \leq 1, & \forall m \in M, i \in I, n \in N, j \in J \quad (3.5h)\\
x_{m,i},\ y_{n,j} \in \{0,1\}, & \forall m \in M, i \in I, n \in N, j \in J \quad (3.5i)
\end{cases}
$$

The objective function (3.5a) minimizes the weighted distance traveled inside the cross-dock by finding optimal flow in the case of several shipping dock doors for a single destination. The constraint (3.5b) guarantees that any origin must be assigned to only one inbound dock door while the constraint (3.5c) ensures that each destination is assigned to enough loading dock doors, the constraints (3.5d) means that for each flow $f_{m,n}$ between origin m and destination n, the sum of percentages $\alpha_{m,i,j,n}$ is equal to 1. The constraints (3.5e) and (3.5f) ensure that each inbound(respectively outbound) dock door can allocate one incoming (respectively outgoing) truck. The constraint (3.5g) guarantees that the total flows sent to a shipping dock door j cannot exceed the capacity of a truck. The

constraint (3.5h) defines the bounds of the continuous variable $\alpha_{m,i,j,n}$. Constraint (3.5i) defines the classical integrality requirement of binary variables $x_{m,i}$ and $y_{n,j}$.

In Tsui and Chang (1990, 1992); Tarhini et al. (2016) the number of dock doors must be equal to the number of trucks and in Cohen and Keren (2008, 2009), at outgoing side a destination can be docked to more than one dock door, this requires the number of load dock doors to be big enough than the number of destinations. All those other variants of CDAP are still simplistic compared to the variant of Zhu et al. (2009) given that nowadays the amount of freights flow is still increasing day after day. Therefore, the number of trucks is big enough than the number of dock doors. The main ingredients for the variant in Zhu et al. (2009) is that the authors consider the management of resources that are dock doors. Each dock door, whether inbound or outbound dock door, is managed as knapsack able to receive more than one object increasing the profit of using the dock door considered as a knapsack.

3.6.5 Connections with other Assignment problems

In this section we present some Assignments problems and depict some connections between those problems and the CDAP standard problem we deal with in this dissertation.

3.6.5.1 Assignment and Generalized Assignment Problem

In Ross and Soland (1975), the authors have given the first formal definition of Generalized Assignment Problem (GAP). They have described the GAP as a special case of classical Assignment Problem (AP). In fact, in the GAP, each task is assigned to one and only one agent while each agent can be assigned to more than one task. For broad details, a set of tasks have to be assigned to a set of agents where each agent has limited resources. The agent's resources can be for instance the number of hours per day. To be achieved, each task requires a certain amount of agent's resources. The importance of the Generalized Assignment Problem is not only its direct application but it appears as a sub-problem in many practical and complex combinatorial optimization problems in the literature, as for instance in our variant of the CDAP.

Ross and Soland (1975) mathematical formulation for the GAP is as follows. Let $M = \{1, 2, ..., |M|\}$ refers to the set of tasks' index and $I = \{1, 2, ..., |I|\}$ refers to the set of agents' index. The parameter $c_{t,p}$ is the cost incurred when task $t \in M$ is executed or

assigned to agent $p \in I$. To be completely executed $a_{t,p}$ refers to the amount of resource that task t requires from agent p and $\mathcal{A}_p$ is the amount of resource available for agent p. A decision variable $\varepsilon_{t,p}$ is defined to ensure an optimal assignment of all tasks $|M|$ to agents $I' \subseteq I$, that is, whether a given task $t \in M$ is executed by an agent $p \in I$ or not.

Taking into account those sets and parameters above, the GAP is mathematically formulated as follows.

$$\forall t \in M,\ p \in I,\ \varepsilon_{t,p} = \begin{cases} 1 & \text{if task } t \text{ is assigned to agent } p \\ 0 & \text{otherwise} \end{cases}$$

$$(GAP) \begin{cases} \textit{minimize} \quad \displaystyle\sum_{t \in M} \sum_{p \in I} c_{m,i} \varepsilon_{t,p} & (3.6a) \\ \text{subject to :} & \\ \displaystyle\sum_{i \in I} \varepsilon_{t,p} = 1 \qquad \forall t \in M & (3.6b) \\ \displaystyle\sum_{m \in M} a_{t,p} \varepsilon_{t,p} \leq \mathcal{A}_p \qquad \forall i \in I & (3.6c) \\ \varepsilon_{t,p} \in \{0,1\}, \qquad \forall t \in M,\ p \in I & (3.6d) \end{cases}$$

The objective function (3.6a) minimizes the total cost when all tasks $|M|$ are assigned to agents $I' \subseteq I$, that is, the solution does not guarantee that all available resources will be used, this means that in optimal solution, some agents can still free of assignment. The assignment constraint (3.6b) ensures that each task $t \in M$ must be assigned to one agent $p \in I$, the knapsack constraint (3.6c) ensures that an agent p can be assigned multiple tasks respecting the capacity $\mathcal{A}_p$ of that agent. Constraint (3.6d) fixes the classical binary decision variable.

The Generalized Assignment Problems do not assume whether or not this task must be executed by this agent, those optimization problems just seek an optimal arrangement of tasks to agents until all tasks are assigned and completely executed with a minimal cost. Accordingly, the parameter $a_{t,p}$ is constant for each agent, that is, $a_{t,p} = a_t$ and means the amount of resource required by task t regardless the agent p.

In comparison with an I-shape cross-dock and the variant of CDAP we deal with in this study, there is a relation between the Generalized Assignment Problem (GAP) and the Cross-dock Door Assignment Problem (CDAP). In fact, an inbound dock door $i \in I$ (respectively an outbound dock door $j \in J$) can be seen as an agent $p \in I$ and an origin $m \in M$ (respectively a destination $n \in N$) can be compared to a task $t \in M$. On

the two sides of I-shape cross-dock, volume of goods s_m from an origin m (respectively volume of demand r_n from a destination n) can be compared to the resource a_t required by task t to be executed and the capacity S_i of inbound dock door i (respectively capacity R_j of outbound dock door j) can be compared to he capacity $\mathcal{A}_p$ of the agent p, see e.g., Guignard et al. (2012). Accordingly, CDAP includes GAP as sub-problem on each side of I-shape cross-dock.

In Ross and Soland (1975); Sahni and Gonzalez (1976), the authors showed that the Generalized Assignment Problem is NP-hard. Accordingly, CDAP is also NP-hard. In addition, according to the computational experiments, the CDAP is still complicated using exact methods and more specially for large scale instances.

3.6.5.2 Quadratic and Generalized Quadratic Assignment Problems

The Quadratic Assignment Problem (QAP) is one of the difficult combinatorial optimization problems to solve using exact methods, especially when the size of the problem grows up. In Adams et al. (2007), the authors have defined the QAP as following

$$QAP\ min\left\{\sum_{i\in N}\sum_{j\in N} b_{i,j}x_{i,j} + \sum_{\substack{i\in N\\ i\neq k}}\sum_{\substack{j\in N\\ j\neq l}}\sum_{k\in N}\sum_{l\in N} c_{i,j,k,l}x_{i,j}x_{k,l} : x\in X, x\in\{0,1\}\right\} \tag{3.7a}$$

where:

$$x\in X \equiv \left\{x\geq 0 : \begin{array}{ll}\displaystyle\sum_{i\in N} x_{i,j}=1, & \forall j\in N\\ \displaystyle\sum_{j\in N} x_{i,j}=1, & \forall i\in N\end{array}\right.$$

The objective function and the constraints all gathered into (3.7a) minimize to total cost $b_{i,j}$ of implanting units $|N|$ to location $|N|$ (first term of objective) and the quadratic cost $c_{i,j,k,l}$ of material flow between a unit i implanted to location j and unit k implanted to location l (second term of objective function). The locations j and l are separated by a distance $d_{j,l}$ and the units i and k exchanges a flow $w_{i,k}$. Therefore the cost $c_{i,j,k,l} = d_{j,l}w_{i,k}$ denotes the weighted distance which corresponds to the product of flow $w_{i,k}$ between unit i and unit k with the distance $d_{j,l}$ between the location j and location l where unit i and unit k are implanted, respectively. The assignment constraints ensure that in each location j is implanted one unit i and each unit i is implanted in one location j. Those two constraints require that the number of units must be equal the number of locations.

In Hahn, Kim, Guignard, Smith and Zhu (2008), the authors have extended this model

by formally modeling the QAP as a Generalized Quadratic Assignment Problem (GQAP). Afterwards, using Reformulation Linearization Technique (RLT), the authors have proposed a linearization of quadratic objective function. Like GAP, the GQAP consider that locations j and l can allocate each one more than one unit respecting the capacity of each location. This implies that the number of units can be greater that the number of locations. Additional parameters such that location capacity, the space needed for each unit are defined.

- M : the set of units
- N : the set of locations
- $a_{i,j}$ is the space needed to implant unit i in location j
- b_j is the available space for location j

Hahn et al. GQAP model :

$$GQAP\ min\left\{\sum_{i\in M}\sum_{j\in N} b_{i,j}x_{i,j} + \sum_{\substack{i\in M\\ i\neq k}}\sum_{\substack{j\in N\\ j\neq l}}\sum_{k\in M}\sum_{l\in N} c_{i,j,k,l}x_{i,j}x_{k,l} : x\in X, x\in\{0,1\}\right\} \quad (3.8a)$$

where:

$$x\in X\equiv\left\{x\geq 0:\ \begin{array}{ll}\displaystyle\sum_{i\in M} a_{i,j}x_{i,j}\leq b_j, & \forall j\in N\\ \displaystyle\sum_{j\in N} x_{i,j}=1, & \forall i\in M\end{array}\right.$$

The constraint

$$\sum_{i\in M} a_{i,j}x_{i,j}\leq b_j, \qquad \forall j\in N$$

ensures that a location j can allocate several units i respecting location capacity b_j while constraint

$$\sum_{j\in N} x_{i,j}=1, \qquad \forall i\in M$$

ensures that each unit i is implanted in one location. As the QAP is hard to solve mainly due the quadratic term, the GQAP is also hard to solve.

The common relation between GQAP and the variant of CDAP we deal with in this study is that both combinatorial optimization problems seek to minimize a quadratic objective function of a cross-product of binary decision variables while managing a finite

amount of resources such that locations and dock doors, respectively. Moreover, in Hahn, Kim, Stuetzle, Kanthak, Hightower, Samra, Ding and Guignard (2008), it has been proved that Cross-dock Door Assignment Problem (CDAP) can be solved as Generalized Quadratic three-dimensional Assignment Problem (GQ3AP). The Standard formulation of CDAP presenting a quadratic term of the two binary decision variables $x_{m,i}$ and $y_{n,j}$ and being hard to solve, some linearization techniques slightly similar to those used to linearize the GQAP have been employed to linearize the quadratic model of Zhu et al. (2009).

We also took a look at the Reformulation and Linearization Technique (RLT) applied on 0-1 integer programming problem in Adams and Sherali (1990). In Adams and Sherali reformulation, a new set of constraints is added for each variable x_j as follows. For each constraints $\sum_{i\in N} a_i x_i \leq b$ of original problem, both constraints $\sum_{i\in N} a_i x_i x_j \leq b x_j$ and $\sum_{i\in N} a_i x_i (1 - x_j) \leq b(1 - x_j)$ are added; and for each constraint $\sum_{i\in N} c_i x_i = d$ of original problem, the constraint $\sum_{i\in N} c_i x_i x_j = d x_j$ is added. This RLT is known to produce the tight linear programming relaxation bounds, see e.g., Pessoa et al. (2010). The intent was to apply Adams and Sherali Adams and Sherali (1990) RLT on the standard quadratic formulation of the CDAP Zhu et al. (2009) to confirm the tightness of the linear programming relaxation bounds of the standard CDAP. The computational results given by this RLT to the standard CDAP are not presented in this dissertation.

3.7 Conclusion

In this chapter we have defined cross-docking and we have given some applications in practice. The difference between warehouse and cross-docking and the strategies employed by cross-docking have been shown, among those strategies, cross-docking eliminates the two most costly operations of warehouses, namely storage and order-picking. The three decision levels, namely strategic, tactical and operational, used to successfully implement a cross-docking have been described. We then have focused on operation level to describe relevant optimization problems raised by cross-docking on that level including the optimization problem we deal with in this PhD thesis. We have emphasized on the combinatorial optimization problem referred to Cross-dock Door Assignment Problem (CDAP) that we tackle here. We have explained the complexity of CDAP due to the presence of cross-product of binary decision variables and to the nature of CDAP. Some variants of this problem have been described and their mathematical formulations have been discussed

and compared with the variant we tackle. A certain relation between the variant of CDAP we deal with here and other classes of assignment problems have been established. In the next chapter we are going to develop and present some new solutions, more precisely we develop new non standard Mixed Integer Linear Programming (MILP) models and we prove their equivalence as well as their equivalence to the standard linear MIP for CDAP.

Chapter 4

Mathematical Programming Formulations for the Cross-dock Door Assignment Problem

Chapter notes : This chapter is a broad development of two articles, a journal article published September 2018 into *The International Journal of Management Science (Omega)* and a manuscript undergoing preparation for submission to publication for which the preliminary experimentation results have been presented into 11^{th} *edition of Journées Plyhèdres et Optimization Combinatoire (JPOC'11)* https://jpoc11.event.univ-lorraine.fr/resource/page/id/9

Contents

4.1 Introduction 56

4.2 Standard Formulation for the CDAP 58

4.2.1 Standard quadratic formulation 59

4.2.2 Standard linearization for the CDAP 60

4.3 Non Standard Assignment and Capacity constraints 61

4.3.1 Assignment constraints 62

4.3.2 Capacity constraints 65

4.4 MIP models and integrality properties 67

4.4.1 Eleven MIP models 67

4.4.2 Integrality properties of MIPs 69

4.5 Lagrangian Relaxation for the CDAP 70

4.6 Computational results 75

4.6.1 Comparison of models - integrality requirement on $z_{m,i,n,j}$ relaxed 77
4.6.2 Comparison of models - integrality requirement imposed on $z_{m,i,n,j}$ 81
4.6.3 Lower bounds comparison for $\mathcal{M}^{2,1}$ and Nassief et al.(2016) . . 86
4.7 Conclusion 89

In this chapter, as widely developed into chapter 3, we tackle the variant of the Cross-dock Door Assignment Problem (CDAP) proposed for the first time in Zhu et al. (2009). This variant of CDAP is an extension of the classical mathematical model introduced in Tsui and Chang (1990, 1992) to take into account the management of dock doors as finite resources. We propose new Mixed Integer Linear Programming (MILP) formulations for the Cross-dock Door Assignment Problem, afterwards, we prove the equivalence between those new MILP models and carry out an extensive comparative study on benchmark data sets from the literature to compare performance between these models and with existing MIP models from literature. To the best of our knowledge, the best MILP model for the CDAP is the one we have proposed for the first time, the results of which are published in Gelareh et al. (2020). Afterwards, we pick one of the proposed MILP models that gives a good compromise between the best lower bounds by the Linear Programming (LP) relaxation and running time consumption and then we apply a Lagrangian Relaxation (LR) approach to generate new and better lower bounds on the optimal solution value given by that MILP model. We use the sub-gradient optimization method to solve the Lagrangian dual problem. The computational results show that Lagrangian dual improves significantly the LP relaxation lower bound and the lower bound given by a recent Lagrangian Relaxation from literature, Nassief et al. (2016). However, this improvement is offset by an important increase of the computational effort needed to solve the relaxation especially when the size of the problem increases.

4.1 Introduction

As already pointed out in chapter 3, cross-docking is a strategy implemented into the cross-dock. We recall that a cross-dock is a type of warehouse in supply chain management that allows orders to be prepared without going through the phase of storing products in the warehouse and subsequently selecting them for delivery. We also recall that Cross-dock Door Assignment Problem (CDAP) is concerning the management of fully loaded incoming trucks named origins that enter to a cross-dock facility and unload their products on inbound dock doors of that facility. The unloaded products are immediately sorted and

organized according to their destinations and, using material handling devices inside the facility such that forklift, those products are immediately transferred to outbound dock doors to be loaded into outgoing trucks named destinations or delivery trucks for being distributed to final customers. Unlike classical warehouses, on cross-dock yard, products are unloaded and loaded without placing them in temporary storage. The goal of the CDAP is to assign origins on inbound dock doors and destinations on outbound dock doors so that the total transportation cost inside the cross-dock is minimized.

The standard quadratic formulation for the CDAP as proposed in Zhu et al. (2009) is hard to solve, even for small-sized instances. Because of the NP-hard character of the CDAP, most of the studies of this combinatorial optimization problem in the literature have been dedicated to developing efficient heuristic solution approaches to cope with large scale instances.

To the best of our knowledge, there are only three Mixed Integer Programming (MIP) formulations for the CDAP in the literature, namely the standard MIP model and the MIP models proposed in Nassief et al. (2016, 2018). In this chapter, we propose eight new non standard MILP models for the CDAP and demonstrate the mathematical equivalence of all 11 models, together with rigorously proving some of their properties. In order to detect which of these 11 models is the best, we conduct an extensive comparative analysis on benchmark instances from the literature, which discloses that the best MILP model is one proposed in this chapter for the first time. We further prove the equivalence of these formulations and identify their integrality properties. Finally, we perform an extensive comparative study of their performance on benchmark instances from the literature, reporting the number of instances solved optimally or not, upper bounds they provide, and CPU time consumed by a CPLEX MIP solver applied to each formulation. More precisely, the comparison of performance between the models is not done analytically as in Nassief et al. (2018), but empirically.

We next propose a Lagrangian Relaxation approach that we apply to the best MILP formulation we proposed for the CDAP to produce a new lower bound on the optimal value and to improve the bound provided by the LP relaxation. The choice of that MILP model for Lagrangian Relaxation is based on a compromise between the LP bound and the processing time to obtain it. The proposed Lagrangian Relaxation relaxes the derived knapsack constraints (new capacity constraints) related to dock doors that have been added to strengthen that best MILP model. The Lagrangian dual problem is solved using the sub-gradient optimization algorithm. The results of computational experiments show

that the Lagrangian dual model improves significantly the LP relaxation bound but still consuming more important CPU time than LP relaxation.

For solving the MILP and Lagrangian models, we have selected the CPLEX solver of IBM since it is one of the most effective solvers and because it is a good indicator of model performance, in the respect that if one model performs better than another using CPLEX then the same ranking of the models occurs when applying other leading solvers.

The rest of this chapter is organized as follows. In section 4.2, we describe the standard quadratic mathematical model originally proposed in Zhu et al. (2009) and the standard linearization of quadratic model. Then, we present the customary approaches to linearize this standard quadratic model of CDAP, that is, we describe the way we replace the cross-product of decision variables in the objective function and the standard MIP model for CDAP. In section 4.3 we introduce new sets of constraints and build new non-standard MILP models for CDAP. We additionally prove the equivalence of those new non standard MILP models between them as well as their equivalence to the standard MIP model presented in literature. In section 4.4, we deal with the integrality requirement on the additional decision variable used to linearize the quadratic objective function and prove that the relaxation of the integrality requirement on some variables will not affect the optimal solution value. In section 4.5 we present the Lagrangian Relaxation (LR) approach and the sub-gradient optimization method we use to solve the Lagrangian dual model. In section 4.6, we provide an exhaustive comparative analysis of the MILP models in order to identify the best MILP model, afterwards, we provide Lagrangian dual results. Computational experiments on both MILP models and Lagrangian Relaxation are carried out on the benchmark data set from the literature. The last section 4.7 of this chapter is dedicated to the conclusion.

4.2 Standard Formulation for the CDAP

In this section we present the standard quadratic formulation for the CDAP proposed in Zhu et al. (2009) together with the standard approach for linearizing this model. In addition, we present some valid equalities and inequalities for the resulting Mixed Integer Programming model.

4.2.1 Standard quadratic formulation

We recall that the standard quadratic formulation of the CDAP is as follows. Given a set M of incoming trucks (origins), a set N of outgoing trucks (destinations), a set I of inbound dock doors and a set J of outbound dock doors, each inbound/outbound dock door may serve more than one origin/destination respectively subjected to the dock doors' capacity constraints and to the assignment constraints of origins/destinations. If the origin $m \in M$ is assigned to the inbound dock door $i \in I$ and the destination $n \in N$ is assigned to the outbound dock door $j \in J$, a transportation cost is incurred. The transportation cost is the product of $d_{i,j}$ and $f_{m,n}$, where $d_{i,j}$ refers to the distance between inbound dock door i and outbound dock door j, and $f_{m,n}$ is the number of pallets moved from the origin m to the destination n. The total number of pallets delivered by an origin $m \in M$ can be computed as $s_m = \sum_{n \in N} f_{m,n}$ and the total number of pallets received at destination $n \in N$ is computed as $r_n = \sum_{m \in M} f_{m,n}$. The capacity of an inbound dock door $i \in I$ is denoted by S_i and the capacity of an outbound dock door $j \in J$ is denoted by R_j. In order to formally model the CDAP, we use the binary decision variable $x_{m,i}$ to indicate whether origin $m \in M$ is assigned to inbound dock door $i \in I$ or not; and the binary decision variable $y_{n,j}$ to indicate whether destination $n \in N$ is assigned to outbound dock door $j \in J$ or not.

According to those sets and decision variables, the CDAP is mathematically formulated in the model Q^{zh} below, see e.g., Zhu et al. (2009).

$$
(Q^{zh}) \begin{cases}
min\ f(x,y) = \displaystyle\sum_{m \in M} \sum_{i \in I} \sum_{n \in N} \sum_{j \in J} f_{m,n} d_{i,j} x_{m,i} y_{n,j} & \text{(4.1a)} \\
\text{subject to :} & \\
\displaystyle\sum_{i \in I} x_{m,i} = 1, \qquad \forall m \in M & \text{(4.1b)} \\
\displaystyle\sum_{j \in J} y_{n,j} = 1, \qquad \forall n \in N & \text{(4.1c)} \\
\displaystyle\sum_{m \in M} s_m x_{m,i} \leq S_i, \qquad \forall i \in I & \text{(4.1d)} \\
\displaystyle\sum_{n \in N} r_n y_{n,j} \leq R_j, \qquad \forall j \in J & \text{(4.1e)} \\
x_{m,i},\ y_{n,j} \in \{0,1\}, \qquad \forall m \in M, n \in N, i \in I, j \in J & \text{(4.1f)}
\end{cases}
$$

In the objective function (4.1a), the quadratic cost $c_{m,i,n,j} = f_{m,n} d_{i,j}$ of product flow $f_{m,n}$ between origin $m \in M$ and destination $n \in N$ and the distance $d_{i,j}$ between inbound

dock door $i \in I$ and outbound dock door $j \in J$ is linked by the quadratic term $x_{m,i}y_{n,j}$ of binary decision variables $x_{m,i}$ and $y_{n,j}$. The objective function (4.1a) minimizes the total transportation cost inside the cross-dock. The two sets of constraints (4.1b) and (4.1c) stand for assignment constraints constraints while the sets of constraints (4.1d) and (4.1e) are the capacity constraints. The last set of constraints (4.1f) imposes the binary requirement on the decision variables, see chapter 3 section 3.6.3.

4.2.2 Standard linearization for the CDAP

In this part, we depict the standard approach for linearizing the quadratic term $x_{m,i}y_{n,j}$ in the quadratic model Q^{zh}. The linearization most used and probably the most natural was first presented in Fortet (1960). It is sometimes called classic or standard linearization.

Given that a quadratic objective function is harder to solve than a linear objective, the quadratic mathematical formulation Q^{zh} may be linearized by dropping the quadratic term $x_{m,i} \times y_{n,j}$ of binary decision variables. Therefore, we introduce a new binary decision variable $z_{m,i,n,j}$ to replace the quadratic term and setting $z_{m,i,n,j}$ as follows:

$$z_{m,i,n,j} = x_{m,i}y_{n,j}, \qquad \forall m \in m, i \in I, n \in N, j \in J$$

In this case of study of the CDAP, as told in chapter 3 this new introduced decision variable $z_{m,i,n,j}$ indicates whether a path $\Omega =< m - i - j - n >$ is established or not. Ω denotes a transfer path of flow $f_{m,n}$ of products from the origin $m \in M$ to the destination $n \in N$ according to whether m is assigned to inbound dock door $i \in I$ or not and n is assigned to outbound dock door $j \in J$ or not. That is, if an origin $m \in M$ is assigned to inbound dock door $i \in I$, i.e., $x_{m,i} = 1$ and a destination $n \in N$ assigned to outbound dock door $j \in J$, i.e., $y_{n,j} = 1$, the path Ω is established, this implies that, $x_{m,i}y_{n,j} = 1 = z_{m,i,n,j}$ ($x_{m,i}$, $y_{n,j}$ and $z_{m,i,n,j}$ binary). In addition, to ensure that the new binary variable $z_{m,i,n,j}$ for all $m \in m$, $i \in I$, $n \in N$, $j \in J$ satisfies its required property (e.g., $z_{m,i,n,j} = 1$ iff $x_{m,i} = y_{n,j} = 1$), the following constraints (4.2a)-(4.2d) need to be stated and added in the standard linear model $\mathcal{M}^{0,0}$ for CDAP.

$$\begin{cases} z_{m,i,n,j} \leq x_{m,i}, & \forall m \in M, \forall i \in I, \forall n \in N, \forall j \in J \quad (4.2a) \\ z_{m,i,n,j} \leq y_{n,j}, & \forall m \in M, \forall i \in I, \forall n \in N, \forall j \in J \quad (4.2b) \\ z_{m,i,n,j} \geq x_{m,i} + y_{n,j} - 1, & \forall m \in M, \forall i \in I, \forall n \in N, \forall j \in J \quad (4.2c) \\ z_{m,i,n,j} \geq 0 & \forall m \in M, \forall i \in I, \forall n \in N, \forall j \in J \quad (4.2d) \end{cases}$$

Thereby, the resulting Mixed Integer Programming model $\mathcal{M}^{0,0}$ is :

$$
(\mathcal{M}^{0,0}) \begin{cases}
min\ g(z) = & \displaystyle\sum_{m\in M}\sum_{i\in I}\sum_{n\in N}\sum_{j\in J} d_{i,j} f_{m,n} z_{m,i,n,j} & \text{(4.3a)} \\
\text{subject to} & & \\
\displaystyle\sum_{i\in I} x_{m,i} = 1, & \forall m \in M & \text{(4.3b)} \\
\displaystyle\sum_{j\in J} y_{n,j} = 1, & \forall n \in N & \text{(4.3c)} \\
z_{m,i,n,j} \leq x_{m,i}, & \forall n \in N, m \in M, i \in I, j \in J & \text{(4.3d)} \\
z_{m,i,n,j} \leq y_{n,j}, & \forall n \in N, m \in M, i \in I, j \in J & \text{(4.3e)} \\
z_{m,i,n,j} \geq y_{n,j} + x_{m,i} - 1, & \forall n \in N, m \in M, i \in I, j \in J & \text{(4.3f)} \\
\displaystyle\sum_{m\in M} s_m x_{m,i} \leq S_i, & \forall i \in I & \text{(4.3g)} \\
\displaystyle\sum_{n\in N} r_n y_{n,j} \leq R_j, & \forall j \in J & \text{(4.3h)} \\
x_{m,i}, y_{n,j} \in \{0,1\}, & \forall m \in M, i \in I, n \in N, j \in J. & \text{(4.3i)} \\
z_{m,i,n,j} \geq 0 & \forall m \in M, i \in I, n \in N, j \in J. & \text{(4.3j)}
\end{cases}
$$

The assignment constraints (4.3b) − (4.3c) and the capacity constraints (4.3g) − (4.3h) are still the same as in Q^{zh} model above. Constraints (4.3d) and (4.3e) ensure that if the origin m is not assigned to the receiving dock door i and the destination n is not assigned to the shipping dock door j, then the transfer path Ω cannot be established. On the other hand, if the origin m is assigned to the receiving dock door i and the destination n is assigned to the shipping dock door j, then the transfer path Ω is established due to constraints (4.3f)

The set of constraints that the MIP $\mathcal{M}^{0,0}$ must satisfy can be decomposed into two sets : i) the set of assignment constraints (4.3b) − (4.3f) and the constraints (4.3i) and (4.3j) on the decision variables which will be gathered into set as $\mathcal{A}^0$, and ii) the set of capacity constraints (4.3g) − (4.3h) which will be gathered into set as $\mathcal{C}^0$.

4.3 Non Standard Assignment and Capacity constraints

In this section we present three sets of assignment constraints that are deduced from the set $\mathcal{A}^0$ as a result of the valid equalities and inequalities as stated in the preceding section. Additionally, we prove the equivalence of these sets of constraints. We also present

a set of capacity constraints deduced directly from the set $\mathcal{C}^0$ and we prove the equivalence between the constraints gathered into those two sets.

4.3.1 Assignment constraints

In this part, we present some valid equalities and inequalities for the resulting Mixed Integer Programming model.

The next proposition provides some valid equalities for above model $\mathcal{M}^{0,0}$.

Proposition 4.3.1 *The constraints of the following system*

$$\begin{cases} \sum_{i\in I} z_{m,i,n,j} = y_{n,j}, & \forall m \in M, n \in N, j \in J \quad (4.4a) \\ \sum_{j\in J} z_{m,i,n,j} = x_{m,i}, & \forall m \in M, n \in N, i \in I \quad (4.4b) \\ \sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1, & \forall m \in M, n \in N \quad (4.4c) \end{cases}$$

are valid for the MIP $\mathcal{M}^{0,0}$.

Proof. The valid equalities (4.4a) and (4.4b) are directly deduced from constraints (4.3b) and (4.3c) by multiplying them by $y_{n,j}$ and $x_{m,i}$, respectively. On the other hand, the valid equality (4.4c) is a direct consequence of the valid equalities (4.4a) and (4.4b) taking into account the constraints (4.3b) and (4.3c). □

The two sets of valid equalities (4.4a) and (4.4b) imply that if the origin m is assigned to the inbound dock door i, then the commodity from the origin m to the destination n must be routed through inbound dock door i and some outbound dock door j ; similarly, if the destination n is assigned to an outbound dock door j, then the commodity from the origin m to the destination n must be routed through outbound dock door j and some inbound door i. The set of inequalities (4.4c) imply that the commodity from the origin m to the destination n is routed via a unique pair (i, j) of inbound dock door i and outbound dock door j.

The first set of assignment constraint that we present here is based on the observation that the nature of the problem implies that the large set of constraints (4.3f) may be replaced by a smaller one as stated in the next proposition.

Proposition 4.3.2 *The constraints* (4.3f) *may be replaced by the set of equalities*:

$$\sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1, \qquad \forall m \in M, n \in N$$

Proof. Constraints (4.3f) ensure that if $x_{m,i} = y_{n,j} = 1$ then $z_{m,i,n,j} = 1$ as well, otherwise they are redundant. On the other hand equalities $\sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1$ for all $m \in M, n \in N$ require that for each m and n there are unique i' and j' so that $z_{m,i',n,j'} = 1$. From constraints (4.3b) and (4.3c), it follows that for each m and n there are as well unique i'' and j'' so that $x_{m,i''} = y_{n,j''} = 1$. Taking into account constraints (4.3d) and (4.3e) we have $z_{m,i,n,j} = 0$ if $i \neq i''$ or $j \neq j''$ and $z_{m,i,n,j} \leq 1$ if $i = i''$ or $j = j''$. This implies that $i' = i''$ and $j' = j''$ and therefore if $x_{m,i''} = y_{n,j''} = 1$ then $z_{m,i'',n,j''} = 1$ □.

As a consequence of the preceding property we obtain the following set of assignment constraints :

Assignment constraints $\mathcal{A}^1$:

$$(\mathcal{A}^1): \begin{cases} \sum_{i\in I} x_{m,i} = 1, & \forall m \in M \quad (4.5a)\\ \sum_{j\in J} y_{n,j} = 1, & \forall n \in N \quad (4.5b)\\ z_{m,i,n,j} \leq x_{m,i}, & \forall n \in N, m \in M, i \in I, j \in J \quad (4.5c)\\ z_{m,i,n,j} \leq y_{n,j}, & \forall n \in N, m \in M, i \in I, j \in J \quad (4.5d)\\ \sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1, & \forall m \in M, n \in N \quad (4.5e)\\ x_{m,i}, y_{n,j} \in \{0,1\}, & \forall m \in M, i \in I, n \in N, j \in J. \quad (4.5f)\\ z_{m,i,n,j} \geq 0, & \forall m \in M, i \in I, n \in N, j \in J.J. \quad (4.5g) \end{cases}$$

The following corollary is a direct consequence of the preceding property.

Corollary 4.3.1 *Assignment constraints $\mathcal{A}^0$ and $\mathcal{A}^1$ are equivalent.*

Replacing the constraints (4.5c) and (4.5d) with equalities :

$$\begin{cases} \sum_{j\in J} z_{m,i,n,j} = x_{m,i}, & \forall m \in M, n \in N, i \in I\\ \sum_{i\in I} z_{m,i,n,j} = y_{n,j}, & \forall m \in M, n \in N, j \in J \end{cases}$$

respectively we obtain the set of assignment constraints gathered into $\mathcal{A}^2$

Assignment constraints $\mathcal{A}^2$:

$$(\mathcal{A}^2): \begin{cases} \sum_{i\in I} x_{m,i} = 1, & \forall m \in M & (4.6a) \\ \sum_{j\in J} y_{n,j} = 1, & \forall n \in N & (4.6b) \\ \sum_{i\in I} z_{m,i,n,j} = y_{n,j}, & \forall m \in M, n \in N, j \in J & (4.6c) \\ \sum_{j\in J} z_{m,i,n,j} = x_{m,i}, & \forall n \in N, m \in M, i \in I & (4.6d) \\ x_{m,i}, y_{n,j} \in \{0,1\}, & \forall m \in M, i \in I, n \in N, j \in J. & (4.6e) \\ z_{m,i,n,j} \geq 0, & \forall m \in M, i \in I, n \in N, j \in J. & (4.6f) \end{cases}$$

The equality (4.5e)

$$\sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1, \qquad \forall m \in M, n \in N$$

can be a result from (4.5a) and (4.5b). Therefore, the inclusion of equalities (4.6c) and (4.6d) together with (4.5a) and (4.5b) make constraints $\sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1$ for all $m \in M, n \in N$ redundant and therefore we do not need to include them in the set $\mathcal{A}^2$. The equivalency of the sets of constraints $\mathcal{A}^1$ and $\mathcal{A}^2$ is then formally proved by the next proposition.

Proposition 4.3.3 *Constraints $\mathcal{A}^1$ and $\mathcal{A}^2$ are equivalent.*

Proof.

($\Rightarrow$) From constraints (4.5e) we have that for each $m \in M$ and $n \in N$ there are unique $i \in I$ and $j \in J$ so that $z_{m,i,n,j} = 1$. This, together with constraints (4.5a) - (4.5d), further implies that $x_{m,i} = 1$ and $x_{m,i'} = 0, i' \in I, i' \neq i$ as well as $y_{n,j} = 1$ and $y_{n,j'} = 0, j' \in J, j' \neq j$. Therefore we have $\sum_{i'\in I} z_{m,i',n,j'} = y_{n,j'} = 0, j' \in J, j' \neq j$ and $\sum_{i'\in I} z_{m,i',n,j} = y_{n,j} = 1$. Similarly, we have $\sum_{j'\in J} z_{m,i',n,j'} = x_{m,i'} = 0, i' \in I, i' \neq i$ and $\sum_{j'\in J} z_{m,i,n,j'} = x_{m,i} = 1$. Consequently, constraints $\mathcal{A}^1$ imply constraints $\mathcal{A}^2$.

($\Leftarrow$) Constraints (4.6c) and (4.6d) imply constraints (4.5c) and (4.5d), respectively. On the other hand, constraints (4.6a) together with constraints (4.6d) imply constraints (4.5e) $\forall m \in M, n \in N$ and constraint (4.6b) together with constraint (4.6c) imply constraint (4.5e) $\forall m \in M, n \in N$. Consequently, constraints $\mathcal{A}^2$ imply constraints $\mathcal{A}^1$.□

As already pointed out, the following constraints

$$\sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1, \qquad \forall m \in M, n \in N$$

are redundant for the set $\mathcal{A}^2$. However, an interesting observation is that replacing constraints (4.6a) and (4.6b) by constraints

$$\sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1, \qquad \forall m \in M, n \in N$$

leads to another valid set of assignment constraints, as follows.

Assignment constraints $\mathcal{A}^3$:

$$(\mathcal{A}^3): \begin{cases} \sum_{i\in I} z_{m,i,n,j} = y_{n,j}, & \forall m \in M, n \in N, j \in J \quad (4.7a)\\ \sum_{j\in J} z_{m,i,n,j} = x_{m,i}, & \forall n \in N, m \in M, i \in I \quad (4.7b)\\ \sum_{i\in I}\sum_{j\in J} z_{m,i,n,j} = 1, & \forall m \in M, n \in N \quad (4.7c)\\ x_{m,i}, y_{n,j}, z_{m,i,n,j} \in \{0,1\}, & \forall m \in M, i \in I, n \in N, j \in J. \quad (4.7d)\\ z_{m,i,n,j} \geq 0, & \forall m \in M, i \in I, n \in N, j \in J. \quad (4.7e) \end{cases}$$

Note that this set of constraints is in the same format as the constraints already proposed in the paper of Nassief et al. (2016); Nassief (2017).

Proposition 4.3.4 *Constraints $\mathcal{A}^2$ and $\mathcal{A}^3$ are equivalent.*

Proof.
($\Rightarrow$) Constraints (4.6a) and (4.6d) imply constraints (4.7c) and therefore constraints $\mathcal{A}^1$ imply constraints $\mathcal{A}^3$.
($\Leftarrow$) Constraints (4.7a) and (4.7c) imply constraints (4.6a), while constraints (4.7b) and (4.7c) imply constraints (4.6b). Hence, constraints $\mathcal{A}^3$ imply constraints $\mathcal{A}^1$. □

From propositions 4.3.2, 4.3.3 and 4.3.4 we have the following consequence.

Corollary 4.3.2 *Assignment constraints $\mathcal{A}^0$, $\mathcal{A}^1$, $\mathcal{A}^2$ and $\mathcal{A}^3$ are equivalent.*

4.3.2 Capacity constraints

Starting from the capacity constraints (4.3g) and (4.3h) gathered into a set $\mathcal{C}^0$ as below :

$$(\mathcal{C}^0): \begin{cases} \sum_{m\in M} s_m x_{m,i} \leq S_i, & \forall i \in I\\ \sum_{n\in N} r_n y_{n,j} \leq R_j, & \forall j \in J \end{cases}$$

as $x_{m,i}$ and $y_{n,j}$ are positive ($x_{m,i},\ y_{n,j} \in \{0,1\}$), we may derive the following set of valid inequalities :

$$(\mathcal{C}^1): \begin{cases} \sum_{m\in M} s_m z_{m,i,n,j} \leq S_i y_{n,j}, & \forall i \in I, n \in N, j \in J \quad (4.8a) \\ \sum_{n\in N} r_n z_{m,i,n,j} \leq R_j x_{m,i}, & \forall j \in J, m \in M, i \in I. \quad (4.8b) \end{cases}$$

Indeed, these two sets of constraints are obtained by multiplying the capacity constraints (4.3g) and (4.3h) by $y_{n,j}$ and $x_{m,i}$, respectively. In Nassief et al. (2016) these two constraints are also considered as valid inequalities. The meaning of the newly established constraints is as follows. Constraints (4.8a) ensure that the total amount of commodities with the destination n routed via the inbound - outbound dock door pair (i,j) do not exceed the capacity limit of the inbound dock door i. Similarly, constraints (4.8b) ensure that the total amount of commodities with the origin m routed via the inbound - outbound dock door pair (i,j) respects the capacity bound of the outbound dock door j.

As already mentioned, the constraints gathered into set $\mathcal{C}^1$ provide also valid inequalities in Nassief et al. (2016). In this section we go further and prove the equivalence between capacity constraints $\mathcal{C}^0$ and $\mathcal{C}^1$ for the CDAP. The proof is based on the fact that $z_{m,i,n,j} = x_{m,i}y_{n,j}$ and the observation that assignment constraints guarantee the existence of $n' \in N$ and $j' \in J$ such that $y_{n',j'} = 1$ as well as the existence of $m' \in M$ and $i' \in I$ such that $x_{m',i'} = 1$ (due to the problem definition).

Proposition 4.3.5 *Capacity constraints $\mathcal{C}^0$ and $\mathcal{C}^1$ for the CDAP are equivalent.*

Proof.

($\Rightarrow$) Multiplying constraints (4.3g) by $y_{n,j}$ for all $n \in N, j \in J$, we obtain the following inequality

$$\sum_{m\in M} s_m z_{m,i,n,j} \leq y_{n,j} S_i, \quad \forall i \in I, n \in N, j \in J, \text{ (using the fact that} z_{m,i,n,j} = x_{m,i}y_{n,j})$$

Similarly, we show that constraints (4.3h) imply constraints (4.8b).

($\Leftarrow$) If we consider the constraint (4.8a), we have

$$\sum_{m\in M} s_m z_{m,i,n,j} = \sum_{m\in M} s_m x_{m,i} y_{n,j} \leq S_i y_{n,j}, \qquad \forall i \in I, n \in N, j \in J$$

Keeping in mind that there exist $n' \in N$ and $j' \in J$ such that $y_{n',j'} = 1$ (this follows from assignment constraints) we have

$$\sum_{m\in M} s_m x_{m,i} \leq S_i, \qquad \forall i \in I$$

Similarly, we can show that constraints (4.8b) imply constraints (4.3h).□

4.4 MIP models and integrality properties

In this section we present MIP models that may be deduced by combining the assignment and capacity constraints presented in the preceding sections. In addition, we identify the integrality properties of these models.

4.4.1 Eleven MIP models

Having four equivalent sets of assignment constraints $\mathcal{A}^k$, for all $k = 0, ..., 3$ and two equivalent sets of capacity constraints $\mathcal{C}^h$, for all $h = 0, 1$ we come up with 8 different new MIP formulations. These 8 MIPs may be stated in general form as :

$$(\mathcal{M}^{k,h})\ min\{g(z) : \mathcal{A}^k, \mathcal{C}^h\},\ \text{for all } k = 0, 1, 2, 3,\ \text{for all } h = 0, 1.$$

The following proposition enable us to generate three new MIP models.

Proposition 4.4.1 *The constraints* (4.3d) *and* (4.3e) *are redundant in the MIP models* $\mathcal{M}^{0,0}$, $\mathcal{M}^{0,1}$ *and* $\mathcal{M}^{1,1}$.

Proof. In models $\mathcal{M}^{0,0}$ and $\mathcal{M}^{0,1}$ constraints (4.3d) and (4.3e) are redundant since we seek to minimize the objective function and the objective coefficients in the CDAP are positive, that is, $c_{m,i,n,j} = f_{m,n}d_{n,j} \geq 0$, $f_{m,n} \geq 0$ for all $m \in M, n \in N$ and $d_{i,j} > 0$ for all $i \in I, j \in J$. In addition, in both models the equality $z_{m,i,n,j} = x_{m,i}y_{n,j}$ remains true even if we exclude constraints (4.3d) and (4.3e), due to the fact that the $z_{m,i,n,j}$ variables are bounded from below only by constraints (4.3f). Namely, if $x_{m,i} = y_{n,j} = 1$, then due to constraints (4.3f) $z_{m,i,n,j}$ will equal 1 as well, while otherwise $z_{m,i,n,j}$ takes the value 0 (again due to the fact that the objective coefficients in the CDAP are positive). The preceding reasoning leads as well as to the conclusion that in models $\mathcal{M}^{0,0}$ and $\mathcal{M}^{0,1}$ with excluded constraints (4.3d) and (4.3e), the integrality requirement on variables $z_{m,i,n,j}$ may be relaxed.
On the other hand, in the model $\mathcal{M}^{1,1}$ the constraints 4.3d (resp. (4.3e)) force $z_{m,i,n,j}$ to be equal to zero if $x_{m,i} = 0$ (resp. $y_{n,j} = 0$). Since the parameters $f_{m,n}$ are positive and by consequence the data s_m and r_n are also positive, the capacities constraints (4.8a) and (4.8b) imply $z_{m,i,n,j} = 0$ if $x_{m,i} = 0$ or $y_{n,j} = 0$.

TABLE 4.1 – Number of constraints for each MIP model

MIP	**Total number of constraints**
$\mathcal{M}^{0,0}$	$3\|I\|\|J\|\|M\|\|N\| + \|I\| + \|J\| + \|M\| + \|N\|$
$\mathcal{M}^{0,1}$	$\|I\|\|J\|(3\|M\|\|N\| + \|M\| + \|N\|) + \|M\| + \|N\|$
$\mathcal{M}^{1,0}$	$(2\|I\|\|J\| + 1)\|M\|\|N\| + \|I\| + \|J\| + \|M\| + \|N\|$
$\mathcal{M}^{1,1}$	$\|I\|\|J\|(2\|M\|\|N\| + \|M\| + \|N\|) + \|M\|\|N\| + \|M\| + \|N\|$
$\mathcal{M}^{2,0}$	$(\|M\|\|N\| + 1)(\|I\| + \|J\|) + \|M\| + \|N\|$
$\mathcal{M}^{2,1}$	$(\|M\| + \|N\|)(1 + \|I\|\|J\|) + \|M\|\|N\|(\|I\| + \|J\|)$
$\mathcal{M}^{3,0}$	$\|M\|\|N\|(\|I\| + \|J\| + 1) + \|I\| + \|J\|$
$\mathcal{M}^{3,1}$	$\|M\|\|N\|(\|I\| + \|J\| + 1) + \|I\|\|J\|(\|M\| + \|N\|)$
$\mathcal{M}'^{0,0}$	$\|M\|\|N\|\|I\|\|J\| + \|I\| + \|J\| + \|M\| + \|N\|$
$\mathcal{M}'^{0,1}$	$\|I\|\|J\|(\|M\|\|N\| + \|M\| + \|N\|) + \|M\| + \|N\|$
$\mathcal{M}'^{1,1}$	$(\|I\|\|J\| + 1)(\|M\|\|N\| + \|M\| + \|N\|)$

Hence the constraints (4.3d) and (4.3e) are redundant in the MIP models $\mathcal{M}^{0,0}$, $\mathcal{M}^{0,1}$ and $\mathcal{M}^{1,1}$.□

As a consequence of the above proposition, we have three new MIP models $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$ and $\mathcal{M}'^{1,1}$ obtained from the corresponding models $\mathcal{M}^{k,h}$ by dropping the constraints (4.3d) and (4.3e). In the model $\mathcal{M}^{1,0}$ the constraints (4.3d) and (4.3e) cannot be omitted because if this is the case, it will be missing a connection between variables $z_{m,i,n,j}$ and $x_{m,i}$ on the one hand and variables $z_{m,i,n,j}$ and $y_{n,j}$ on the other hand.

The 11 MIPs have the same number of binary variables, i.e., $|I||J||M||N| + |I||M| + |J||N|$. The Table 4.1 provides the number of constraints in each of the 11 MIP models $\mathcal{M}^{k,h}$, for all $k = 0, .., 3$, $h = 0, 1$ and $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$, $\mathcal{M}'^{1,1}$.

Comparing the number of constraints for each of these 11 MIP models shown in Table 4.1, it may be inferred that the number of constraints in the model $\mathcal{M}^{2,0}$ is smaller than in any other model. As will be shown in Section 4.7 reserved for computational results, the reason why the model $\mathcal{M}^{2,0}$ is solved the fastest is probably due the the fact it has fewer constraints than other models.

4.4.2 Integrality properties of MIPs

This section provides properties which show that in all our MIP formulations the requirement $z_{m,i,n,j} \in \{0,1\}$ for all $m \in M, n \in N, i \in I, j \in J$, can be relaxed to require just $z_{m,i,n,j} \in [0,1]$ for all $m \in M, n \in N, i \in I, j \in J$.

Proposition 4.4.2 *The integrality requirement on variables $z_{m,i,n,j} \in \{0,1\}$ for all $m \in M, n \in N, i \in I, j \in J$, in constraints $\mathcal{A}^0$ may be relaxed. Moreover, the binary variables $z_{m,i,n,j} \in \{0,1\}$ may be replaced by $z_{m,i,n,j} \geq 0$.*

Proof. Suppose $z_{m,i,n,j} = \alpha > 0$ for some $m \in M, n \in N, i \in I, j \in J$. Then, due to constraints (4.3d) and (4.3e) we have $x_{m,i} = 1$ and $y_{n,j} = 1$ respectively. This further implies $z_{m,i,n,j} = \alpha \geq 1$ from the constraint (4.3f) and therefore $\alpha = 1$. The last statement is deduced from constraints (4.3d) and (4.3e), and the fact that the variables $x_{m,i}$ and $y_{n,j}$ are binary. □

Proposition 4.4.3 *The integrality requirement $z_{m,i,n,j} \in \{0,1\}$ in constraints $\mathcal{A}^1$ may be relaxed.*

Proof. Let suppose that we just impose requirement $z_{m,i,n,j} \geq 0$ and for some $m \in M$ and $n \in N$ and some $i \in I$ and $j \in J$ we have $z_{m,i,n,j} = \alpha > 0$. Because of constraints (4.5d) we have $\alpha \leq 1$. In addition, constraints (4.5c) and (4.5d) imply that $y_{n,j} = 1$ and $x_{m,i} = 1$. On the other hand, constraints (4.5a) and (4.5b) imply that $y_{n,j'} = 0$ for all $j' \in J, j' \neq j$ and $x_{m,i'} = 0$ for all $i' \in I, i' \neq i$. This implies in turn that $z_{m,i'',n,j'} = 0$ for all $j' \in J, j' \neq j, i'' \in I$ (from constraints (4.5c)) and $z_{m,i',n,j''} = 0$ for all $i' \in I, i' \neq i, j'' \in J$ (from constraints (4.5d)). Hence, taking into account constraint (4.5e) we have $1 = \sum_{i'' \in I} \sum_{j'' \in J} z_{m,i'',n,j''} = z_{m,i,n,j} = \alpha$. Consequently, the integrality requirement $z_{m,i,n,j} \in \{0,1\}$ in constraints $\mathcal{A}^1$ may be relaxed. □

Proposition 4.4.4 *The integrality requirement $z_{m,i,n,j} \in \{0,1\}$ in constraints $\mathcal{A}^2$ may be relaxed.*

Proof. Suppose we impose requirement $z_{m,i,n,j} \geq 0$. Because of constraint (4.6c) we have $z_{m,i,n,j} \leq 1$. Suppose then for some fixed $m \in M$ and $n \in N$ and some $i \in I$ and $j \in J$, we have $z_{m,i,n,j} = \alpha \in \{0,1\}$. Then, this implies that $y_{n,j} = 1$ and $x_{m,i} = 1$ because of constraints (4.6c) and (4.6d). Hence, from constraints (4.6c) and (4.6d)

follow $\sum_{i' \in I, i' \neq i} z_{m,i',n,j} = 1 - \alpha$ and $\sum_{j' \in J, j' \neq j} z_{m,i,n,j'} = 1 - \alpha$. Taking into account that $\sum_{i \in I} \sum_{j \in J} z_{m,i,n,j} = \sum_{i \in I} x_{m,i} = \sum_{j \in J} y_{n,j} = 1$ (this chain of equalities is deduced by summing the constraints (4.6c) over set J and the constraints (4.6d) over set I noting that $\sum_{i \in I} x_{m,i} = 1$ and $\sum_{j \in J} y_{n,j} = 1$) we have

$$1 = \sum_{i \in I} \sum_{j \in J} z_{m,i,n,j} \geq \sum_{i' \in I, i' \neq i} z_{m,i',n,j} + \sum_{j' \in J, j' \neq j} z_{m,i,n,j'} + z_{m,i,n,j} = 2 - \alpha.$$

This implies $\alpha \geq 1$ which is a contradiction. Hence, the integrality requirement may be relaxed. □

Proposition 4.4.5 *The integrality requirement $z_{m,i,n,j} \in \{0,1\}$ in constraints $\mathcal{A}^3$ may be relaxed.*

Proof. Analogous to the proof of Proposition 4.4.4. □

Note that the preceding property of the $z_{m,i,n,j}$ variables in constraints $\mathcal{A}^3$ has also been detected in Nassief et al. (2016, 2018).

Proposition 4.4.6 *The integrality requirement may be relaxed on the variables $z_{m,i,n,j} \in \{0,1\}$ for all $m \in M, n \in N, i \in I, j \in J$, in models $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$ and $\mathcal{M}'^{1,1}$.*

Proof. The proof is a direct consequence of the preceding propositions and Proposition 4.4.1, which implies that in each of models $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$ and $\mathcal{M}'^{1,1}$ constraints (4.3d) and (4.3e) may be deduced from the constraints in a model. □

To the best of our knowledge, the standard MIP formulation $\mathcal{M}^{0,0}$ was already considered in Zhu et al. (2009), while another type of reformulation that looks like the model $\mathcal{M}^{3,0}$ has been proposed in Nassief et al. (2016). On the other hand, the remaining MIPs have not been yet considered for solving the CDAP.

4.5 Lagrangian Relaxation for the CDAP

In section 4.7, computational experiments carried out on benchmark instances from the literature show that MILP model $(\mathcal{M}^{2,0})\, min\, \{g(z) : \mathcal{A}^2, \mathcal{C}^0\}$ is the fastest model for CDAP known so far. We also recall that the derivative MILP model $(\mathcal{M}^{2,1})\, min\, \{g(z) : \mathcal{A}^2, \mathcal{C}^1\}$ is obtained by replacing initial capacity constraints (3.1d) and (3.1e) gathered into $\mathcal{C}^0$ by (4.8a) and (4.8b) gathered into $\mathcal{C}^1$ to strengthen $\mathcal{M}^{2,0}$. In Nassief et al. (2016),

the authors have considered the MILP model $(\mathcal{M}^{3,0})\, min\, \{g(z) : \mathcal{A}^3, \mathcal{C}^0\}$ and instead of replacing the constraints (4.1d)-(4.1e) by (4.8a) and (4.8b) respectively, the authors have added them as redundant and they came out a new MILP model $(\mathcal{M}^{3,0,1})\, min\, \{g(z) : \mathcal{A}^3, \mathcal{C}^0, \mathcal{C}^1\}$ for CDAP. That resulting MILP model $\mathcal{M}^{3,0,1}$ is too weak in term of processing time consumption. This is may be due to the big number of constraints, but it performs the same LP relaxation lower bound as $\mathcal{M}^{2,1}$.

The following of this section provides the Lagrangian Relaxation approach that we have used to exploit the MIP model $\mathcal{M}^{2,1}$ so as to provide new lower bound to optimal solution value found so far. Lagrangian Relaxation is a well-known optimization method that has been significantly investigated to solve many optimization problems. It consists of reducing problem complexity by introducing hard constraints associated with their respective penalties as a part of objective function (Geoffrion; 1972, 1974). Lagrangian Relaxation has been used in Held and Karp (1970, 1971) to solve Traveling Salesman Problem (TSP).

The integral MILP model $\mathcal{M}^{2,1}$ to which we apply Lagrangian Relaxation approach is defined as follows :

$$
(\mathcal{M}^{2,1}) \begin{cases}
min\ g(z) = & \displaystyle\sum_{m\in M}\sum_{i\in I}\sum_{n\in N}\sum_{j\in J} d_{i,j} f_{m,n} z_{m,i,n,j} \quad (4.9a) \\
\text{subject to :} & \\
\displaystyle\sum_{m\in M} s_m z_{m,i,n,j} \leq S_i y_{n,j}, & \forall i \in I, n \in N, j \in J \quad (4.9b) \\
\displaystyle\sum_{n\in N} r_n z_{m,i,n,j} \leq R_j x_{m,i}, & \forall j \in J, m \in M, i \in I. \quad (4.9c) \\
\displaystyle\sum_{i\in I} x_{m,i} = 1, & \forall m \in M \quad (4.9d) \\
\displaystyle\sum_{j\in J} y_{n,j} = 1, & \forall n \in N \quad (4.9e) \\
\displaystyle\sum_{i\in I} z_{m,i,n,j} = y_{n,j}, & \forall m \in M, n \in N, j \in J \quad (4.9f) \\
\displaystyle\sum_{j\in J} z_{m,i,n,j} = x_{m,i}, & \forall n \in N, m \in M, i \in I \quad (4.9g) \\
x_{m,i}, y_{n,j}, z_{m,i,n,j} \in \{0,1\}, & \forall m \in M, n \in N, i \in I, j \in J. \quad (4.9h) \\
z_{m,i,n,j} \in \{0,1\}, & \forall m \in M, n \in N, i \in I, j \in J. \quad (4.9i)
\end{cases}
$$

Here, we propose a Lagrangian Relaxation procedure that relaxes the derived capacity constraints (4.9b) and (4.9c). For Nassief et al. (2016), in the considered MIP model

$\mathcal{M}^{3,0,1}$, the authors relaxed capacity constraints (4.9b) and (4.9c) together with assignment constraints (4.9f) - (4.9g) in a Lagrangian way and solve Lagrangian dual model using sub-gradient optimization algorithm. The authors showed that the Lagrangian objective function can be decomposed into three sub-problems, each sub-problem in a space of the corresponding decision variable $x_{m,i}$, $y_{n,j}$ and $z_{m,i,n,j}$ respectively. Unfortunately the Lagrangian lower bound found is worse. In fact, from the results of computational experiments, the lower bound value of Lagrangian dual is smaller than that of corresponding LP relaxation for all instances.

In our case of study, the relaxation of the constraints (4.9b) and (4.9c) in a Lagrangian fashion has been motivated by the following findings : *i*) some preliminary computational experiments showed that the optimal solution value when capacity constraints (4.9b) and (4.9c) are dropped is the same as the LP relaxation lower bound of $(\mathcal{M}^{2,0})\, min\,\{g(z) : \mathcal{A}^2, \mathcal{C}^0\}$ and the model is faster than LP relaxation of $\mathcal{M}^{2,0}$; *ii*) the LP relaxation lower bound of $\mathcal{M}^{2,1}$ is better than that of $\mathcal{M}^{2,0}$ for all instances, see e.g., Table A.5.

We define $\lambda_{i,n,j}, \gamma_{j,m,i} \geq 0$ that stand for the Lagrange multipliers associated to each capacity constraints (4.9b) and (4.9c), respectively.

The Lagrangian Relaxation model can be stated as follows :

$$
\mathcal{LR}(\lambda,\gamma)\begin{cases}
min\ l(\lambda,\gamma) = & \displaystyle\sum_{m\in M}\sum_{i\in I}\sum_{n\in N}\sum_{j\in J} a_{m,i,n,j} z_{m,i,n,j} - \sum_{n\in N}\sum_{j\in J} b_{n,j} y_{n,j} \\
& \qquad - \displaystyle\sum_{m\in M}\sum_{i\in I} c_{m,i} x_{m,i} \\
\textbf{subject to :} & \\
\displaystyle\sum_{i\in I} x_{m,i} = 1, & \forall m \in M \\
\displaystyle\sum_{j\in J} y_{n,j} = 1, & \forall n \in N \\
\displaystyle\sum_{i\in I} z_{m,i,n,j} = y_{n,j}, & \forall m \in M, n \in N, j \in J \\
\displaystyle\sum_{j\in J} z_{m,i,n,j} = x_{m,i}, & \forall n \in N, m \in M, i \in I \\
x_{m,i}, y_{n,j} \in \{0,1\}, & \forall m \in M, n \in N, i \in I, j \in J \\
z_{m,i,n,j} \geq 0, & \forall m \in M, n \in N, i \in I, j \in J
\end{cases}
$$

where :

- $a_{m,i,n,j} = d_{i,j} f_{m,n} + \lambda_{i,n,j} s_m + \gamma_{j,m,i} r_n$

- $b_{n,j} = \sum_{i \in I} \lambda_{i,n,j} S_i$
- and $c_{m,i} = \sum_{j \in J} \gamma_{j,m,i} R_j$

4.5.1 Solving Lagrangian Dual model

We let $v(\mathcal{P})$ denotes the optimal value of a minimization optimization problem $\mathcal{P}$ and $v(\overline{\mathcal{P}})$ denotes the optimal value given by the usual LP relaxation of the same problem. For any Lagrange multiplier μ, let $\mathcal{LR}_\mu$ denotes the Lagrangian Relaxation program for problem $\mathcal{P}$. It is well known that $\mathcal{LR}_\mu$ provides a lower bound $v(\mathcal{LR}_\mu)$ on the optimal solution value for the original linear problem. The goal is to find an optimal value μ that provides the best lower bound value by solving Lagrangian dual model related to the Lagrangian program $\mathcal{LR}_\mu$. Let z_D be the Lagrangian dual model associated to $\mathcal{LR}_\mu$. Theoretically, $v(\overline{\mathcal{P}}) \leq v(z_D) \leq v(\mathcal{P})$.

In our case of study, the best possible lower bound is obtained by solving the Lagrangian dual program $(\mathcal{D})$ below :

$$(\mathcal{D}) \begin{cases} max\ z = & v(\mathcal{LR}(\lambda, \gamma)) \\ \textbf{subject to :} & \lambda, \gamma \geq 0 \end{cases}$$

$$v(\overline{\mathcal{M}^{2,1}}) \leq v(\mathcal{D}) \leq v(\mathcal{M}^{2,1}),\ \text{for all}\ \ \lambda,\ \gamma \geq 0$$

where :

- $v(\overline{\mathcal{M}^{2,1}})$ is the LP relaxation lower bound of the model $\mathcal{M}^{2,1}$
- $v(\mathcal{D})$ is the lower bound obtained by solving dual model $\mathcal{D}$
- $v(\mathcal{M}^{2,1})$ is the optimal solution value of the model $\mathcal{M}^{2,1}$

The sub-gradient optimization method, see e.g., Shor (1968); Schirotzek (1986); Fisher (1981), is one of the existing iterative algorithms to find good values for Lagrange multipliers. For a certain number of iterations, the sub-gradient optimization algorithm adjusts iteratively the value of Lagrange multipliers to the solution value that is the best or near the best lower bound. To solve the Lagrangian dual program $\mathcal{D}$, we use the sub-gradient optimization algorithm, see e.g., Held et al. (1974); Fisher (1981). At each iteration, the sub-gradient optimization method depicted in algorithm 4.1 solves Lagrangian dual and updates Lagrange multipliers $\lambda_{i,n,j}$ and $\gamma_{j,m,i}$ to move Lagrangian lower bound value $v(\mathcal{LR}(\lambda, \gamma))$ in direction of optimal solution value of original problem $\mathcal{M}^{2,1}$.

Algorithm 4.1 Sub-gradient Optimization Algorithm

Parameters :

$LB \leftarrow -\infty$; $UB \leftarrow$ Upper bound on $v(\mathcal{M}^{2,1})$;

$\sigma^0 \leftarrow 2$; $\epsilon \leftarrow 10^{-5}$; $t \leftarrow 0$ $N_{count} \leftarrow 0$; N_{max}

$(\lambda^{\mathbf{0}}, \gamma^{\mathbf{0}}) \leftarrow (\mathbf{0}, \mathbf{0})$**{**$\lambda, \gamma$ **start by zero}**

$(\lambda^{\mathbf{0}}, \gamma^{\mathbf{0}}) \leftarrow (\mathbf{d}$(4.9b)$, \mathbf{d}$(4.9c)$)$**{**$\lambda, \gamma$ **start by dual value of associated LP}**

repeat

  Solve Lagrangian model $\mathcal{LR}(\lambda^t, \gamma^t)$

  $l(\lambda^t, \gamma^t) \leftarrow \mathcal{LR}(\lambda^t, \gamma^t)$ $\{l(\lambda^t, \gamma^t)$ is optimal value of $\mathcal{LR}(\lambda, \gamma)$ model at iteration $t\}$

  if $(l(\lambda^t, \gamma^t) > LB)$ **then**

    $LB \leftarrow l(\lambda^t, \gamma^t)$

    $N_{count} \leftarrow 0$

  else

    $N_{count} \leftarrow N_{count} + 1$

  end if

  Compute the sub-gradient $\mathcal{G}(\lambda^t, \gamma^t)$ of $\mathcal{LR}(\lambda, \gamma)$

  Compute stepsize $\varphi^t \leftarrow \frac{\sigma^t(UB - l(\lambda^t, \gamma^t)))}{||\mathcal{G}(\lambda^t, \gamma^t)||^2}$

  Updates Lagrange multipliers $(\lambda^{t+1}, \gamma^{t+1}) \leftarrow Max\{0, (\lambda^t, \gamma^t) + \varphi^t.\mathcal{G}(\lambda^t, \gamma^t)\}$

  if $N_{count} \geq N_{max}$ **then** {if no progress in more than N_{max} iterations}

    $\alpha \leftarrow random(0, 1)$

    if $\alpha\sigma^t \leq \epsilon$ **then**

      $\sigma^{t+1} \leftarrow \sigma^0$ {reset σ^{t+1} to its starting value}

    else

      $\sigma^{t+1} \leftarrow \alpha\sigma^t$

    end if

    $N_{count} \leftarrow 0$

  else

    $\sigma^{t+1} \leftarrow \sigma^t$

  end if

  $t \leftarrow t + 1$

until terminate()

Return LB

Let (x^*, y^*, z^*) be an optimal solution of Lagrangian Relaxation $\mathcal{LR}(\lambda^t, \gamma^t)$ obtained at iteration t, the sub-gradient of the associated function $v(\mathcal{LR}(\lambda^t, \gamma^t))$ at that iteration t is $\mathcal{G}(\lambda^t, \gamma^t) = (\mathcal{G}^{\lambda^t}, \mathcal{G}^{\gamma^t})$ where:

$$\begin{cases} \mathcal{G}^{\lambda^t}_{i,n,j} = \sum_{m \in M} s_m z^*_{m,i,n,j} - S_i y^*_{n,j}, & \forall i \in I, n \in N, j \in J \\ \mathcal{G}^{\gamma^t}_{i,m,j} = \sum_{n \in N} r_n z^*_{m,i,n,j} - R_j x^*_{m,i}, & \forall i \in I, m \in M, j \in J \end{cases}$$

From sub-gradient algorithm 4.1 below, the stopping criteria are a maximum number of iterations, a maximum processing time to solve each instance and if the sub-gradient $\mathcal{G}(\lambda^t, \gamma^t)$ equal to zero. If one of these criteria is met, the sub-gradient algorithm terminates (line 29) returning the best lower bound found so far (line 30). In this algorithm, LB stands for the lower bound on optimal solution value of $\mathcal{M}^{2,1}$. After a number of consecutive iterations $nMax$ without improvement of LB, the parameter $\sigma^t \in [0, 2]$ starting with $\sigma^0 = 2$ is decreased by a random value $\alpha \in]0, 1[$ and σ^t is reset to its initial value 2 if it attains ϵ. The lines 3 and 4 define that sub-gradient algorithm starts by zero or by dual values of associated LP for Lagrange multipliers.

The upper bound UB is generated by a constructive heuristic. This heuristic begins by assigning in sequential way all origins on inbound dock doors. Afterwards, regarding origins assignment, each destination is then assigned in a greedy way. After this initial assignment, origins are removed from dock doors where they are assigned and are reassigned in a greedy way regarding destinations assignment.

4.6 Computational results

All tests presented in this section were conducted on a personal computer Intel(R) Core(TM) with i7-6700HQ 2.60GHz CPU and 16GB of RAM, running Windows 10 OS. To solve the MIP formulations and Lagrangian model we have used CPLEX 12.6.3 solver and the sub-gradient algorithm has been coded in Java IDE. The MIP formulations are compared in terms of the quality of the upper bounds they provide, and the CPU time consumed by CPLEX to solve an instance with a time limit set to 2 hours (7200 seconds). Lagrangian Relaxation and LP relaxation lower bounds are compared as well as the CPU time consumed. For sub-gradient algorthmin 4.1 one of the two criteria, 2 hours or 2000 iterations has to be fulfilled for the algorithm to end. For testing purposes, 50 benchmark

instances [a] proposed in Guignard et al. (2012) were used. The authors have generated this data set in the following way. They filled the flow matrix $(f_{m,n})$, for all $m \in M, n \in N$ with randomly generated integer values between 10 and 50 until 25% of the flow matrix was filled. It is assumed that a destination n will receive a flow of at least $f_{m,n}$ from one origin m which will send at least flow $f_{m,n}$ to one destination n. The process is repeated until all $|M|$ origins and all $|N|$ destinations are accommodated assuming $|M| = |N|$. To generate the distance matrix, the I-shape cross-docking facility is assumed to have an equal number of inbound and outbound dock doors, i.e., $|I| = |J|$. Most of the applications instances have cross-docks with a width of 90 feet and dock doors with a width of 12 feet, which corresponds approximately to the proportion of 8 to 1. Therefore, in all instances distances range from 8 to $8+|I|-1$. In addition, the I-shaped cross-dock has a rectangular shape, with receiving dock doors on one side and outbound dock doors on the other side. Therefore, rectilinear distances may accurately simulate distances traversed by the forklifts following clearly marked lanes, see e.g., Guignard et al. (2012). This means that all instances are generated to correspond to a realistic situation. The capacity of each dock door is set to be equal to the total flow coming from all origins divided by the total number of inbound dock doors, plus the quotient of $p\%$ of the slackness of the total flow, where $p \in \{5, 10, 15, 20, 30\}$. More precisely, the dock door capacity is calculated using the following formula :

$$Slack = \frac{\sum_{m \in M} s_m}{|I|} * p\%$$

$$Dock\ door\ capacity = \frac{\sum_{m \in M} s_m}{|I|} + Slack = \frac{\sum_{m \in M} s_m}{|I|}(1 + p\%)$$

The number of origins/destinations in the instances ranges from 8 to 20, while the number of inbound/outbound dock doors is between 4 and 10.

For the MIP formulations, the computational results are divided into two parts. In the first part we test the MIP models where integrality requirements on the variables $z_{m,i,n,j}$ are relaxed, while in the second part we keep the integrality requirements. We identify models with relaxed integrality requirements by denoting them as $\mathcal{M}^{k',h}$ where $\mathcal{M}^{k,h}$ is the corresponding model with the integrality requirement intact.

a. https://tinyurl.com/yb6l6vmz

4.6.1 Comparison of models - with integrality requirement on variables $z_{m,i,n,j}$ relaxed

In Tables 4.2 and 4.3, we provide a summary of the results obtained by the 11 MIP models (with a relaxed integrality requirement on the variables $z_{m,i,n,j}$). By the convention that "**solving** " an instance means that a feasible solution is found, Table 4.2 provides summary results in terms of the number of instances solved (row ''`# instances`''), the number of instances solved to optimality (row ''`# optimal`''), the average optimality gap attained by CPLEX (row ''`gap`''), the average CPU time used by CPLEX to solve an instance (row ''`CPU time`'') and the average number of nodes processed (row ''`# nodes`''). Table 4.3 provides some detailed results for each class of instances for models that succeed in solving all instances. Instances with the same number of origins/destinations and inbound/outbound dock doors form a class. The number of origins/destinations and inbound/outbound dock doors in each class is given in the first column of the Table 4.3 in the form $|N| \times |I|$. The remaining columns of the table report for each method the average solution value (column ''`value`''), the average CPU time (column ''`CPU time`'') and the average optimality gap (column ''`gap`'') attained by CPLEX on each class.

	$\mathcal{M}^{0',0}$	$\mathcal{M}^{0',1}$	$\mathcal{M}^{1',0}$	$\mathcal{M}^{1',1}$	$\mathcal{M}^{2',0}$	$\mathcal{M}^{2',1}$	$\mathcal{M}^{3',0}$	$\mathcal{M}^{3',1}$	$\mathcal{M}'^{0',0}$	$\mathcal{M}'^{0',1}$	$\mathcal{M}'^{1',1}$
# instances	38	50	50	44	50	49	50	47	41	50	50
# optimal	21	29	38	28	45	34	45	34	29	29	34
gap	0.143	0.213	0.020	0.030	0.010	0.037	0.010	0.022	0.029	0.193	0.033
CPU time	3546.10	3449.20	2069.87	3102.73	745.35	2489.73	768.48	2406.56	2325.72	3275.19	2801.96
# nodes	403590.47	919624.10	21718.80	3423.85	6909.22	3721.92	7254.68	4107.19	2115087.02	1522937.64	5134.08

TABLE 4.2 – Comparison of models - integrality requirement on variables $z_{m,i,n,j}$ relaxed

$\|N\| \times \|I\|$	$\mathcal{M}^{0',1}$			$\mathcal{M}^{1',0}$			$\mathcal{M}^{2',0}$			$\mathcal{M}^{3',0}$			$\mathcal{M}'^{0',1}$			$\mathcal{M}'^{1',1}$		
	value	time	gap	value	time	gap	value	time	gap	value	time	gap	value	time	gap	value	time	gap
8x4	5120.8	2.92	0.000	5120.8	1.15	0.000	5120.8	0.22	0.000	5120.8	0.23	0.000	5120.8	1.69	0.000	5120.8	3.08	0.000
9x4	5978.2	5.82	0.000	5978.2	1.53	0.000	5978.2	0.20	0.000	5978.2	0.38	0.000	5978.2	3.56	0.000	5978.2	4.70	0.000
10x4	6319.8	28.28	0.000	6319.8	2.76	0.000	6319.8	0.34	0.000	6319.8	0.53	0.000	6319.8	15.58	0.000	6319.8	11.99	0.000
10x5	6427.8	297.31	0.000	6427.8	8.73	0.000	6427.8	0.77	0.000	6427.8	1.18	0.000	6427.8	111.90	0.000	6427.8	97.44	0.000
11x5	7555.6	1600.02	0.000	7555.6	14.52	0.000	7555.6	1.40	0.000	7555.6	1.94	0.000	7555.6	572.83	0.000	7555.6	673.42	0.000
12x5	7972.8	5838.02	0.109	7970.2	61.64	0.000	7970.2	2.75	0.000	7970.2	3.58	0.000	7978.8	5791.21	0.107	7970.2	749.89	0.000
12x6	10452.4	5119.59	0.093	10449.8	413.18	0.000	10449.8	12.10	0.000	10449.8	13.75	0.000	10474.8	4655.11	0.056	10452.4	4879.05	0.015
15x6	13819.6	7200.00	0.500	13756.4	5878.42	0.001	13756.4	61.56	0.000	13756.4	128.25	0.000	13849.4	7200.00	0.452	13842.6	7200.00	0.040
15x7	14786.2	7200.00	0.524	14705.8	7200.00	0.028	14688.8	174.13	0.000	14688.8	334.93	0.000	14761.8	7200.00	0.446	14836.0	7200.00	0.061
20x10	29869.4	7200.00	0.902	29904.0	7200.00	0.174	29602.4	7200.00	0.101	29641.4	7200.00	0.101	29638.2	7200.00	0.873	33157.2	7200.00	0.216

TABLE 4.3 – Comparison of models on each instance class - integrality requirement on variables $z_{m,i,n,j}$ relaxed

From the reported results we observe that only models $\mathcal{M}^{0',1}$, $\mathcal{M}^{1',0}$, $\mathcal{M}^{2',0}$, $\mathcal{M}^{3',0}$, $\mathcal{M}'^{0',1}$, and $\mathcal{M}'^{1',1}$, enable us to solve all the 50 instances using CPLEX. Among them, models $\mathcal{M}^{2',0}$ and $\mathcal{M}^{3',0}$ are the best two, both yielding the best optimality gap (0.010%), solving the largest number (45) of instances to optimality and consuming the least CPU time on the average. Their superiority over the other models is also confirmed by a 95% confidence interval plot of the optimality gap (see Figure 4.1). The MIP formulation $\mathcal{M}^{2',0}$ needed 745.35 seconds on average, while $\mathcal{M}^{3',0}$ consumed 768.48 seconds to solve an instance. These values are about three times less than the average CPU time consumed by the next fastest formulation $\mathcal{M}^{1',0}$. On the other hand, the two worst models, in terms of the number of solved instances, turn out to be models $\mathcal{M}^{0',0}$ and $\mathcal{M}'^{0',0}$ for which CPLEX was only able to solve 38 and 41 instances, respectively. In addition, we observe that all models $\mathcal{M}^{0',1}$, $\mathcal{M}^{1',0}$, $\mathcal{M}^{2',0}$, $\mathcal{M}^{3',0}$, $\mathcal{M}'^{0',1}$, and $\mathcal{M}'^{1',1}$ are capable of optimally solving instances with up to 11 origins/destinations and 5 inbound/outbound dock doors. However, only models $\mathcal{M}^{2',0}$ and $\mathcal{M}^{3',0}$ succeed in optimally solving each instance with up to 15 origins/destinations and 7 inbound/outbound dock doors. On the largest class of instances, model $\mathcal{M}^{2',0}$ exhibits slightly better performance in terms of solution value than $\mathcal{M}^{3',0}$ (see Appendix A).

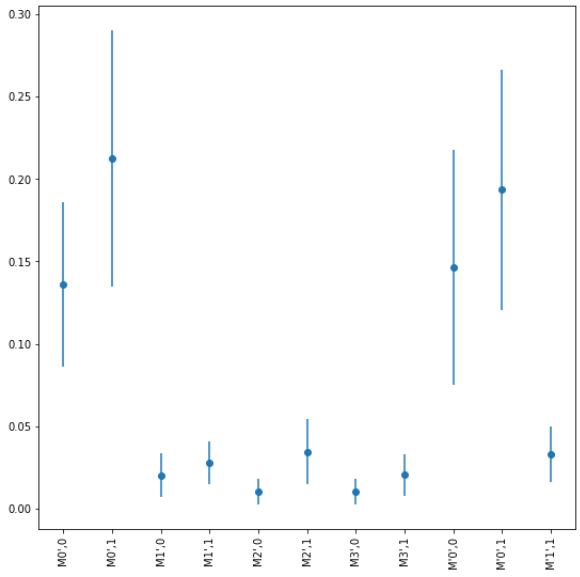


FIGURE 4.1 – 95% confidence interval plot of the optimality gap-integrality requirement on variables $z_{m,i,n,j}$ relaxed

To further assess the performance of models $\mathcal{M}^{0',1}$, $\mathcal{M}^{1',0}$, $\mathcal{M}^{2',0}$, $\mathcal{M}^{3',0}$, $\mathcal{M}'^{0',1}$, and $\mathcal{M}'^{1',1}$ which enable CPLEX to provide a solution for all instances considered, we use performance profiles as suggested in Dolan and Moré (2002). For each method two perfor-

mance profiles are generated : one with respect to the best upper bounds found and the another with respect to the CPU times consumed. We denote the best upper bound by $U_{\mathcal{M}}$ and denote the CPU time consumed in solving an instance by $T_{\mathcal{M}}$. Then, to compare $U_{\mathcal{M}}$ or $T_{\mathcal{M}}$ for different models, we compute the ratio $R^M_{\mathcal{M}} = M_{\mathcal{M}}/\min_{\mathcal{M}' \in \overline{\mathcal{M}}}\{M_{\mathcal{M}'}\}$, where $M_{\mathcal{M}}$ stands for $U_{\mathcal{M}}$ or $T_{\mathcal{M}}$ and $\overline{\mathcal{M}}$ is the set of models to be compared. Therefore, the performance profile of model $\mathcal{M}$ with respect to metric $R^M_{\mathcal{M}}$ measured over each instance s in a set S is simply the graph of the cumulative distribution function, defined as :

$$F^M_{\mathcal{M}}(r) = |\{s \in S \,|\, R^M_{\mathcal{M}} \leq r\}|/|S|.$$

In the graph, $R^M_{\mathcal{M}}$ values are given on the x-axis, while $F^M_{\mathcal{M}}$ values are given on y-axis.

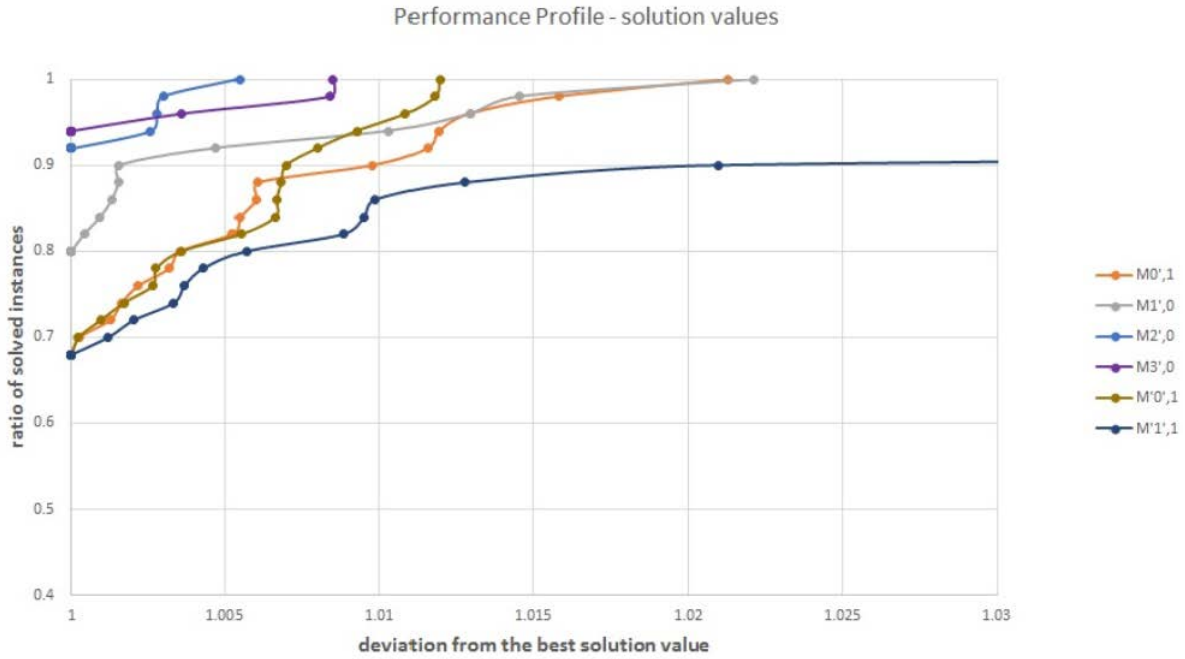


FIGURE 4.2 – Performance profile-solution values : integrality requirement on variables $z_{m,i,n,j}$ relaxed

From the performance profiles presented in Figures 4.2 and 4.3 we may conclude that models $\mathcal{M}^{2',0}$ and $\mathcal{M}^{3',0}$ clearly dominate all the others. The average optimally gaps presented in Table 4.2 were indicative of this advantage, but this is now confirmed by the upper bound and CPU time performance profiles, where we see the graphs of $\mathcal{M}^{2',0}$ and $\mathcal{M}^{3',0}$ on top of the others. If we compare the upper bound performance profiles of $\mathcal{M}^{2',0}$ and $\mathcal{M}^{3',0}$ we see that they cross once in the interval $[1, 1.005]$. Namely, the upper bound performance profile of $\mathcal{M}^{3',0}$ dominates that of $\mathcal{M}^{2',0}$ in the interval $[1, 1.0025]$, which means that $\mathcal{M}^{3',0}$ finds an upper bound within 0.25% of the best upper bound for more

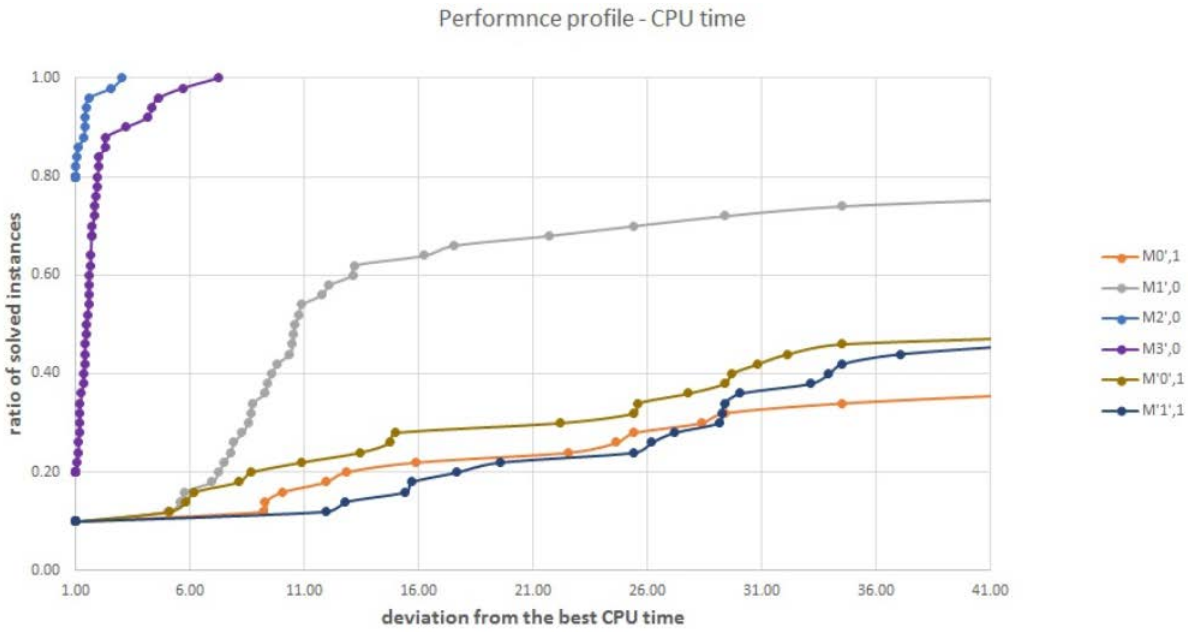


FIGURE 4.3 – Performance profile-CPU times : integrality requirement on variables $z_{m,i,n,j}$ relaxed

instances than $\mathcal{M}^{2',0}$. Starting from the crossing point, the upper bound performance profile of $\mathcal{M}^{2',0}$ starts to dominate that of $\mathcal{M}^{3',0}$. In addition, we observe that the largest deviation from the best solution value attained by model $\mathcal{M}^{2',0}$ is about 0.25% less than the largest deviation from the best solution value attained by model $\mathcal{M}^{3',0}$. However, the difference between models $\mathcal{M}^{2',0}$ and $\mathcal{M}^{3',0}$ is not statistically significant, in terms of the optimality gap, as can be observed from the 95% confidence interval plot of the optimality gap (see Figure 4.1). On the other hand, if we compare CPU times in the performance profiles of $\mathcal{M}^{2',0}$ and $\mathcal{M}^{3',0}$, we observe that the model $\mathcal{M}^{2',0}$ clearly outperforms the model $\mathcal{M}^{3',0}$. The superiority of $\mathcal{M}^{2',0}$ over $\mathcal{M}^{3',0}$ in terms of CPU time consumed is established by a Wilcoxon signed rank test, refer to e.g., Wilcoxon (1945), which yields a p-value < 0.0001 (i.e., $p = 5.3e^{-6}$). In view of these observations we may say that the model $\mathcal{M}^{2',0}$ is better than any other model compared, especially if a high quality solution is sought in a short time.

4.6.2 Comparison of models - with integrality requirement on variables $z_{m,i,n,j}$ imposed

Similary to Tables 4.2 and 4.3, in Tables 4.4 and 4.5 we again compare the preceding models but now with the integrality requirement imposed on the variables $z_{m,i,n,j}$

	$\mathcal{M}^{0,0}$	$\mathcal{M}^{0,1}$	$\mathcal{M}^{1,0}$	$\mathcal{M}^{1,1}$	$\mathcal{M}^{2,0}$	$\mathcal{M}^{2,1}$	$\mathcal{M}^{3,0}$	$\mathcal{M}^{3,1}$	$\mathcal{M}'^{0,0}$	$\mathcal{M}'^{0,1}$	$\mathcal{M}'^{1,1}$
# instances	46	47	50	47	50	49	50	49	41	48	50
# optimal	21	28	45	39	45	39	45	38	29	27	34
gap	0.268	0.239	0.015	0.016	0.011	0.024	0.012	0.026	0.158	0.248	0.026
CPU time	4324.69	3544.65	1036.82	1818.23	841.08	1891.40	1177.04	2053.82	2284.94	3525.66	2693.76
# nodes	416648.28	394352.79	16222.62	386.43	10639.40	384.55	24638.02	606.80	2113500.12	450466.65	1040.50

TABLE 4.4 – Comparison of models - integrality requirement on variables $z_{m,i,n,j}$ imposed

$\|N\| \times \|I\|$	$\mathcal{M}^{0,1}$			$\mathcal{M}^{2,0}$			$\mathcal{M}^{3,0}$			$\mathcal{M}'^{1,1}$		
	value	time	gap	value	time	gap	value	time	gap	value	time	gap
8x4	5120.8	0.70	0.000	5120.8	0.15	0.000	5120.8	0.35	0.000	5120.8	2.49	0.000
9x4	5978.2	1.25	0.000	5978.2	0.21	0.000	5978.2	0.54	0.000	5978.2	4.64	0.000
10x4	6319.8	1.67	0.000	6319.8	0.38	0.000	6319.8	0.93	0.000	6319.8	16.38	0.000
10x5	6427.8	6.31	0.000	6427.8	1.00	0.000	6427.8	3.17	0.000	6427.8	98.49	0.000
11x5	7555.6	6.04	0.000	7555.6	1.64	0.000	7555.6	4.24	0.000	7555.6	210.63	0.000
12x5	7970.2	14.40	0.000	7970.2	3.67	0.000	7970.2	14.32	0.000	6280.4	898.74	0.000
12x6	10449.8	87.18	0.000	10449.8	26.09	0.000	10449.8	80.48	0.000	10449.8	4106.28	0.004
15x6	13756.4	648.34	0.000	13756.4	132.93	0.000	13756.4	622.78	0.000	13867.6	7200.00	0.035
15x7	14688.8	2402.35	0.000	14688.8	1044.73	0.000	14688.8	3843.62	0.000	14965.0	7200.00	0.065
20x10	30165.4	7200.00	0.145	29828.2	7200.00	0.105	30004.2	7200.00	0.124	31067.2	7200.00	0.155

TABLE 4.5 – Comparison of models on each instance class - integrality requirement on variables $z_{m,i,n,j}$ imposed

The results presented in Table 4.4 show that only 4 of 11 models enable CPLEX to provide a solution for each instance in the data set. These four models are : $\mathcal{M}^{1,0}$, $\mathcal{M}^{2,0}$, $\mathcal{M}^{3,0}$ and $\mathcal{M}'^{1,1}$. Of these, model $\mathcal{M}'^{1,1}$ enabled 34 instances to be solved to optimality, while the remaining three enabled 45 instances to be solved. More precisely, just on the class containing the largest instances, models $\mathcal{M}^{1,0}$, $\mathcal{M}^{2,0}$, $\mathcal{M}^{3,0}$ failed to find optimal solutions and the best performance in terms of solution quality is exhibited by model $\mathcal{M}^{2,0}$ (see Table 4.5). Further, if we compare the average optimality gap attained by using these 4 models, we see that the least average optimality gap is provided by $\mathcal{M}^{2,0}$ (0.011%), while model $\mathcal{M}'^{1,1}$ yields the largest average optimality gap (0.024%). From the 95% confidence interval plot of the optimality gaps in Figure 4.4, we observe that there is no significant difference among these four models. Comparing the average CPU time consumed to solve an instance, model $\mathcal{M}^{2,0}$ yields the least average CPU time consumed (841.08) which is significantly less than the average CPU time consumed when using model $\mathcal{M}^{1,0}$ (1036.82), the second best of the models by this criterion. To further verify that $\mathcal{M}^{2,0}$ is best in terms of solution quality and solution time performance, in Figures 4.5 and 4.6 we draw the upper bound and CPU time performance profiles of models $\mathcal{M}^{1,0}$, $\mathcal{M}^{2,0}$, $\mathcal{M}^{3,0}$ and $\mathcal{M}'^{1,1}$ using the approach described in the preceding section. These figures show that the graphs representing the upper bound and CPU time performance profiles of model $\mathcal{M}^{2,0}$ lie above the others. The superiority of model $\mathcal{M}^{2,0}$ over models $\mathcal{M}^{1,0}$ and $\mathcal{M}^{3,0}$, the closest competitors in terms of CPU time consumption is confirmed by the Wilcoxon signed rank test which yields p-values of $5.48e^{-8}$ and $5.18e^{-9}$ by comparing $\mathcal{M}^{2,0}$ and $\mathcal{M}^{1,0}$, and $\mathcal{M}^{2,0}$ and $\mathcal{M}^{3,0}$, respectively. On the other hand, we recall that, in terms of the number of solved instances, the models $\mathcal{M}^{0,0}$ and $\mathcal{M}'^{0,0}$ were the two worst, enabling CPLEX to solve only 38 and 41 instances, respectively.

The comparison results in the previous tables lead to some interesting observations. We see that after relaxing the integrality restrictions on the variables $z_{m,i,n,j}$, model $\mathcal{M}^{1',0}$ is less efficient(causes CPLEX to perform less efficiently) than the corresponding model $\mathcal{M}^{1,0}$. However, model $\mathcal{M}^{2',0}$ is better than its corresponding model $\mathcal{M}^{2,0}$ regarding both solution quality and CPU time consumed. These observations lead to the conclusion that is difficult to say in the case of certain models whether it is better to relax the integrality requirement for some variables or not. However, it is interesting to observe that the standard linear MIP formulation $\mathcal{M}^{0,0}$ is the weakest. It consumes a substantial amount of CPU time even for the simple instances, and additionally consumes a lot of memory for some instances. The associated MIP formulation $\mathcal{M}^{0',0}$ behaves the same way in terms of

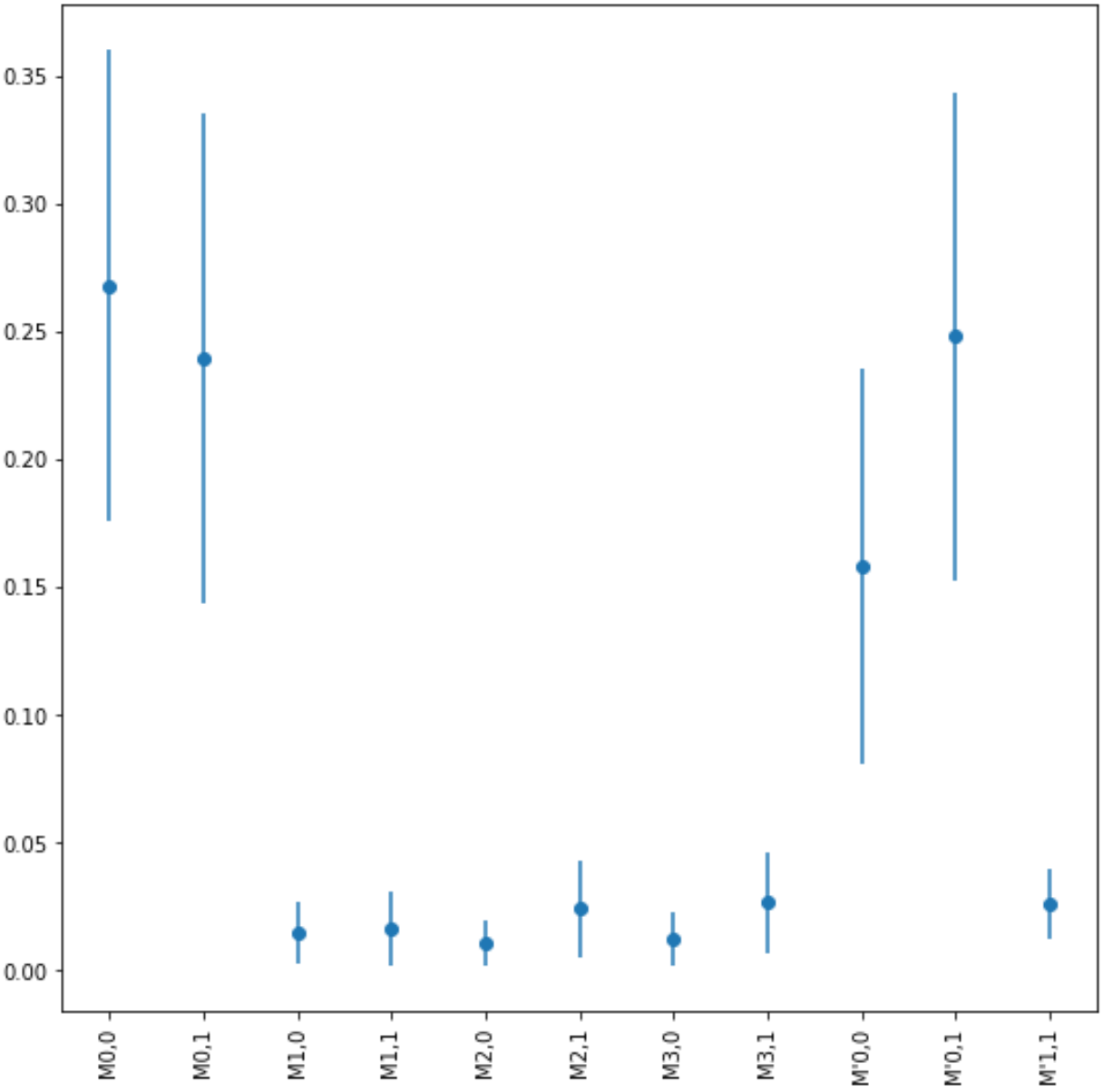


FIGURE 4.4 – 95% confidence interval plot of the optimality gap-integrality requirement on variables $z_{m,i,n,j}$ imposed

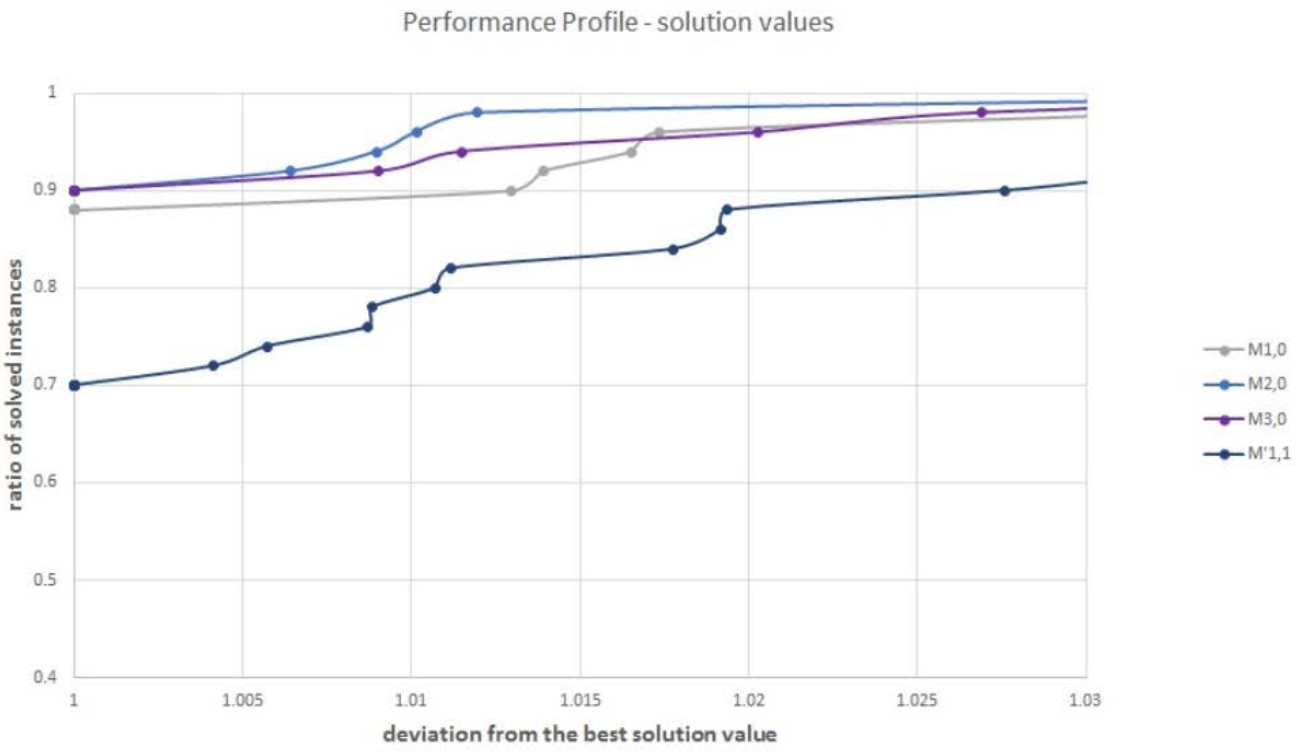


FIGURE 4.5 – Performance profile-solution values : integrality requirement imposed on variables $z_{m,i,n,j}$

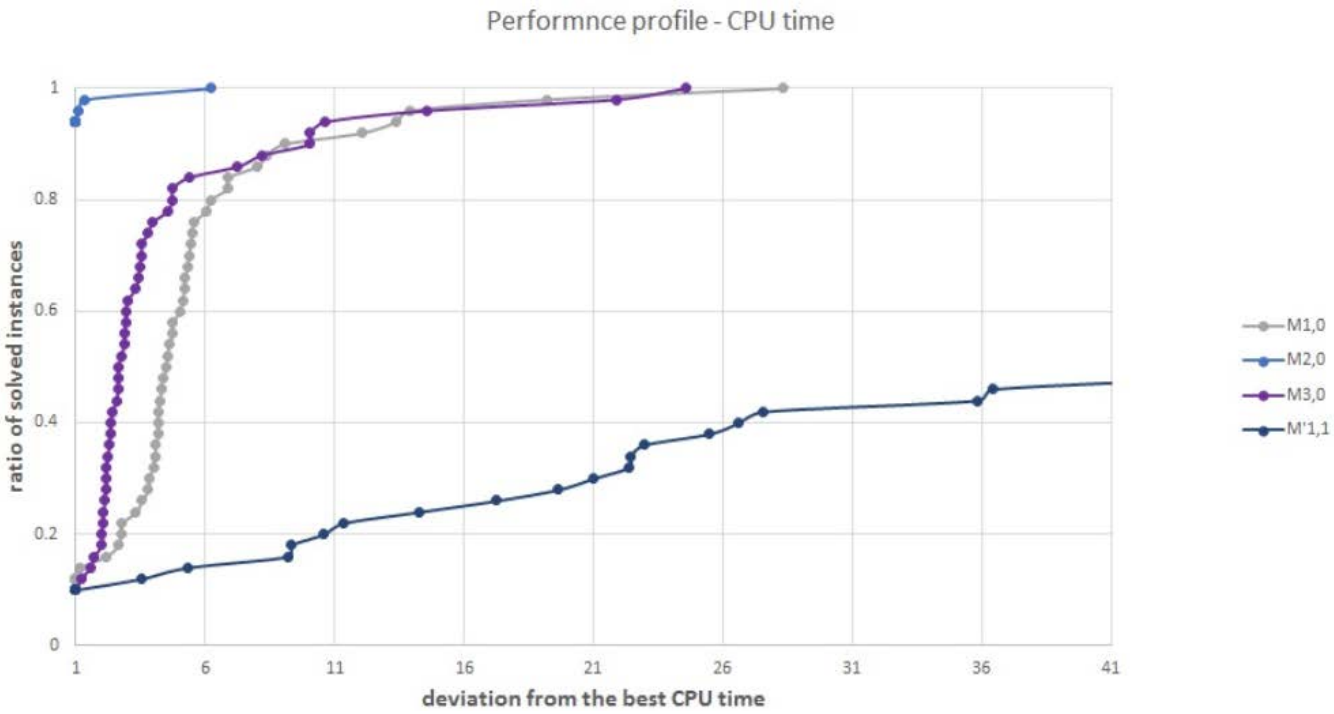


FIGURE 4.6 – Performance profile-CPU times : integrality requirement imposed on variables $z_{m,i,n,j}$

memory consumption but is slightly faster in terms of running times. In sum, we conclude that the model $\mathcal{M}^{2',0}$ is the best among those considered in this chapter 4, both with and without relaxing the integrality requirement. We emphasize once again that to the best of our knowledge, the " winning " models $\mathcal{M}^{2',0}$ and $\mathcal{M}^{2,0}$ are considered here for the first time.

The LP relaxations of models $\mathcal{M}^{0,0}$, $\mathcal{M}^{0,1}$, $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$ are the weakest, yielding an LP relaxation value of zero on all instances. The average LP relaxation values of the remaining models as well as the average CPU times (in second) needed to obtain these values, over entire set of instances, are given in Table 4.6. As we can see the model $\mathcal{M}^{2,0}$ exhibits the best compromise between LP relaxation value and CPU time consumption. This may explain why models $\mathcal{M}^{2',0}$ and $\mathcal{M}^{2,0}$ are the best. In addition, the results reported in Table 4.6 suggest that the behavior of the models detected in this chapter may be very similar to the behavior when some other MIP solver is used. The models $\mathcal{M}^{0,0}$, $\mathcal{M}^{0,1}$, $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$ would be most likely the worst, while the models $\mathcal{M}^{2',0}$, $\mathcal{M}^{2,0}$, $\mathcal{M}^{3,0}$, and $\mathcal{M}^{3',0}$ and $\mathcal{M}^{1,0}$ would be most likely among the best.

	$\mathcal{M}^{1,0}$	$\mathcal{M}^{1,1}$	$\mathcal{M}^{2,0}$	$\mathcal{M}^{2,1}$	$\mathcal{M}^{3,0}$	$\mathcal{M}^{3,1}$	$\mathcal{M}'^{1,1}$
LP value	9627	10006	9627	10028	9627	10028	10004
CPU time	0.67	13.02	0.21	7.37	0.33	7.52	7.88

TABLE 4.6 – Comparison of LP relaxations

Expect the worst MIPs $\mathcal{M}^{0,0}$, $\mathcal{M}^{0,1}$, $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$ which yield LP relaxation lower bound zero for all instances, we see in table 4.6 that all MIPs models $\mathcal{M}^{k,h}, \forall k = 1, ..., 3$ and $h = 1$ yield an average lower bound value slightly higher that the average lower bound of corresponding MIP $\mathcal{M}^{k,h}, \forall k = 1, ..., 3$ and $h = 0$. We can conclude that the two derived sets of constraints (4.8a) and (4.8b) impact improvement of the LP lower bound. Computational results of all 11 MIPs models and their performance are detailed on Appendix A.

4.6.3 Lower bounds comparison for $\mathcal{M}^{2,1}$ and Nassief et al.(2016)

Table 4.7 summarizes the computational results of the Lagrangian Relaxation approach. The first column stands for the name of the instance class. For each class of

instances, the column BKS stands for the average value of the best known solutions, the three columns under $\mathcal{M}^{2,1}$ CPLEX are the average for optimal solution value, gap (in percentage) and CPU time (in second), respectively for the original MILP model $\mathcal{M}^{2,1}$. For each class of instances, the remaining nine columns LB, $dev(\%)$ and Time (in second) under $\mathcal{M}^{2,1}$ LP, $\mathcal{M}^{2,1}$ $LR, (\lambda^0, \gamma^0) = (0, 0)$ and $\mathcal{M}^{2,1}$ $LR, (\lambda^0, \gamma^0) = (d(4.9b), d(4.9c))$ stand for average lower bound value, deviation and CPU time (in second). The deviation expressed in percentage is computed as follows : $dev(\%) = 100(BKS - LB)/BKS$.

The results in the Table 4.7 prove that Lagrangian Relaxation lower bounds are better than those of LP relaxation and that the sub-gradient is sensitive to the starting values of Lagrange multipliers. Indeed, the LR lower bound is better when sub-gradient starts by dual values than when it starts by empty constraints for Lagrange multipliers. However, the associated processing time is more important than that of the LP relaxation. The lower bounds and deduced deviation for $\mathcal{M}^{2,1}$ MILP model are compared with LR lower bound ad deviation in Nassief et al. (2016), i.e., the two columns under heading LR Nassief. Lagrangian lower bound in Nassief et al. (2016) is weak and smaller than the corresponding LP relaxation lower bound. Detailed computational results can be viewed in the appendix table A.8

N x I	BKS	$\mathcal{M}^{2,1}$ CPLEX			$\mathcal{M}^{2,1}$ LP			$\mathcal{M}^{2,1}$ $LR, (\lambda^0, \gamma^0) = (0,0)$			$\mathcal{M}^{2,1}$ $LR, (\lambda^0, \gamma^0) = (d(4.9b), d(4.9c))$			LR Nassief	
		Value	Gap(%)	Time	LB	dev(%)	Time	LB	dev(%)	Time	LB	dev(%)	Time	LB	dev(%)
8x4	5120.80	5120.80	0.00	2.13	4984.05	2.67	0.14	5007.82	2.20	0.40	5020.64	1.95	0.18	4955.62	3.22
9x4	5978.20	5978.20	0.00	4.31	5789.05	3.16	0.24	5864.15	1.90	0.39	5871.43	1.78	0.29	5743.22	3.93
10x4	6319.80	6319.80	0.00	6.11	6078.96	3.79	0.37	6185.08	2.11	0.43	6192.91	1.99	0.34	6015.12	4.80
10x5	6427.80	6427.80	0.00	51.59	6148.77	4.32	1.83	6208.49	3.39	1.22	6225.22	3.13	0.65	6079.96	5.39
11x5	7555.60	7555.60	0.00	270.63	7190.16	4.81	0.25	7313.91	3.17	2.94	7347.72	2.73	1.78	7087.54	6.17
12x5	7970.20	7970.20	0.00	313.63	7603.30	4.60	0.32	7807.48	2.04	7.18	7830.81	1.75	4.59	7505.67	5.83
12x6	10449.80	10453.00	0.03	3590.95	9797.74	6.20	0.55	10034.31	3.94	93.84	10078.59	3.51	40.17	9695.58	7.19
15x6	13756.40	13883.80	0.09	7200.00	12985.44	5.60	0.97	13266.99	3.55	168.79	13341.26	3.01	117.58	12803.30	6.93
15x7	14688.80	15061.60	0.07	7200.00	13659.07	7.00	2.12	13796.78	6.05	557.38	13927.86	5.16	236.00	13440.01	8.49
20x10	29171.40	-	-	-	26038.50	10.73	66.96	25838.99	11.40	4139.04	25872.56	11.29	3305.62	25665.55	12.01
Avg	**10743.88**	**-**	**-**	**-**	**10027.50**	**5.29**	**7.37**	**10132.40**	**3.98**	**497.16**	**10170.90**	**3.63**	**370.72**	**9899.16**	**6.40**

TABLE 4.7 – Lower bounds comparison for $\mathcal{M}^{2,1}$ and Nassief et al. (2016)

4.7 Conclusion

In this chapter, we started the study of the Cross-dock Door Assignment Problem (CDAP) from the standard quadratic formulation of the problem and we have derived 11 nonstandard Mixed Integer Linear Programming (MILP) models for the CDAP. Eight of these 11 proposed MILP models are considered in this chapter for the first time. We prove the equivalence between all these models, with an integrality requirement imposed on the z variables, in the sense of admitting the same feasible and optimal solutions. We also establish results about the integrality properties of these models. These results further imply the equivalence of the models that have relaxed integrality requirement on the z variables.

To detect the best model among these 11, an exhaustive empirical study has been performed on benchmark instances from the literature, applying the CPLEX MIP software to compare the formulations in terms of the number of instances they enable to be solved to optimality, upper bounds they provide, and the CPU time consumed. The results reveal that the best model is one of the eight MILP formulations proposed in this study. In addition, we picked the MILP model that gives good compromise between LP relaxation lower bound and CPU time consumption and apply the Lagrangian Relaxation procedure that relaxes the capacity constraints. The goal was to produce new lower bounds on optimal solution value. Lagrangian dual was solved using the sub-gradient optimization algorithm. The lower bound given by our Lagrangian dual outperforms that given by LP relaxation and that given by Lagrangian dual from literature but the consumed processing time is important specially for large scale instances.

However, the challenge remains to identify an effective solution algorithm and model formulation for handling large scale instances whose solution remains elusive. A possible research direction is to propose a hybrid approach that combines the best model from those identified in this chapter with an existing or newly proposed heuristic algorithm. This heuristic can be combined to the sub-gradient method to solve efficiently Lagrangian dual model.

Chapter 5

Probabilistic Tabu Search for the Cross-dock Door Assignment Problem

Chapter notes : This chapter is a development of an article published into *European Journal of Operational Research (EJOR)*. The preliminary works of that article have been presented into the following conferences :

- 12$^{\text{th}}$ Metaheuristics International Conference (MIC 2017), http://mic2017.upf.edu/
- 19$^{\text{ème}}$ congrès de la société Française de Recherche Opérationnelle et d'Aide à la Décision (ROADEF2018), http://roadef2018.labsticc.fr/wp/

Contents

5.1 Introduction 92

5.2 Main ingredients of the Probabilistic Tabu Search approaches 93

5.2.1 Constructive heuristic to generate an initial solution 95

5.2.2 Neighborhood structures and move evaluation 96

5.3 Probabilistic Tabu Search 103

5.3.1 Probabilistic Tabu Search : Variant 1 104

5.3.2 PTS : Variant 2 106

5.4 Computational Results 108

5.4.1 Comparison of exhaustive and heuristic exploration of swap neighborhood 108

5.4.2 Comparison with methods from the literature 109

5.5 Conclusion 113

The solutions proposed in chapter 4 are still weak for large instances of the Cross-dock Door Assignment Problem (CDAP). In this chapter, we propose and implements two novel heuristics that solve efficiently the CDAP. The proposed heuristics are evaluated on 99 benchmark instances from the literature, disclosing that our heuristics approaches outperform recent state-of-the-art approaches by reaching 45 previous best-known solutions in the literature and discovering 53 new best-known solutions while consuming significantly less CPU time.

5.1 Introduction

The heuristics that we have proposed are applied to an considering I-shape cross-dock. We recall that for an I-shape cross-dock where the inbound dock doors are set at one side and the sets of outbound dock doors are set at the opposite side, fully loaded incoming trucks enter the cross-dock and unload goods at inbound dock doors and unloaded goods are immediately transferred to outbound dock doors to be loaded into outgoing truck the same day. For a broad description, literature review and mathematical formulation of the Cross-dock Door Assignment Problem, let's refer the readers to chapter 3 on section 3.6.

The CDAP is considered as an instance of an assignment problem as shown in Guignard et al. (2012). Assignment problems are well-studied optimization problems that have given rise to numerous proposals for solution algorithms including both metaheuristics and exact methods, see, e.g., Pentico (2007). To briefly indicate some of the more salient contributions, variants of assignment problems that have received attention include : The Generalized Assignment Problem (GAP), see, e.g., Yagiura et al. (2006), the generalized quadratic assignment problem, see, e.g., Pessoa et al. (2010) and the quadratic three-dimensional assignment problem, see, e.g., Hahn, Kim, Stuetzle, Kanthak, Hightower, Samra, Ding and Guignard (2008). In Zhu et al. (2009), the authors observe a relationship between the Generalized Quadratic three-dimensional Assignment Problem (GQ3AP) and the CDAP we study here which discloses that the CDAP can be solved as GQ3AP.

Some Mixed Integer Programming formulations of the studied problem may be found in Guignard et al. (2012); Nassief et al. (2016); Gelareh et al. (2020). According to the computational results reported in the previous chapter, instances with up to 15 origins/destinations and 7 indoors/outdoors may be optimally solved by the CPLEX MIP solver within the time limit of two hours. However, the largest instances remain elusive

for the CPLEX MIP solver and therefore there is a need for heuristic approaches.

In this chapter, we develop two novel heuristics based on Probabilistic Tabu Search (PTS) utilizing a new neighborhood structure applicable both to CDAP and related problems to solve this NP-hard optimization problem.

We recall that Tabu Search (TS) is a metaheuristic introduced by Glover (1986) that cross boundaries of feasibility or local. It guides a local search to explore the solution space beyond local optimality by using adaptive memory to create a flexible search. The two variant of PTS differ from each other in the way they construct a candidate list of solutions and accept new incumbent solutions. In addition, we propose a new extension of the swap neighborhood that allows the exchange of more than two elements and we design an efficient heuristic method to explore it. Extensive testing is performed on benchmark instances from the literature to assess the performance of our proposed approaches, showing that our PTS heuristics outperform the previous state-of-the art approaches by reaching 45 previous best-known solutions and discovering 53 new best-known solutions on a set of 99 instances. In addition, the CPU time consumed by our approaches during exploration these existing solutions and the 53 new best solutions is substantially less than that consumed by the previous state-of-the art methods. We also conduct tests to show that our heuristic exploration yields a good trade-off between solution quality and CPU time in comparison with exhaustive exploration of our new neighborhood structure.

The rest of the chapter is organized as follows. The next section describes the main ingredients of the proposed heuristics based on Probabilistic Tabu Search, including a procedure for constructing an initial solution, as well as the neighborhood structures used and efficient ways of exploring them. Section 5.3 presents the two Probabilistic Tabu Search heuristics built on the ingredients described in the preceding section and Section 5.4 is dedicated to computational experiments to assess the merit of the proposed approaches. Finally, Section 5.5 offers concluding observations.

5.2 Main ingredients of the Probabilistic Tabu Search approaches

In this section we present the main ingredients of our proposed Probabilistic Tabu Search heuristics with multiple neighborhood structures. First, we present the procedure used to generate an initial solution and then we describe neighborhood structures exploited

by our PTS heuristics. In addition, we expose the data structures and updating procedures used in our implementation.

A solution of the CDAP is represented by partitions of M and N denoted by X and Y, respectively. Each element X_i of X is a set containing all origins assigned to the inbound dock door $i \in I$. Similarly, each element Y_j is a set containing all destinations assigned to the outbound dock door $j \in J$. More formally, using two binary variables defined as follows : $x_{m,i}$ and $y_{n,j}$ indicate whether or not an incoming truck $m \in M$ is assigned to inbound dock door $i \in I$, and whether or not an outgoing truck $n \in N$ is assigned to outbound dock door $j \in J$, respectively, the set X_i and Y_j may be expressed as :

$$X_i = \{m \in M : x_{m,i} = 1\} \text{ and } Y_j = \{n \in N : y_{n,j} = 1\}$$

We note that some sets X_i or Y_j can be empty in a feasible solution.

The objective function (3.1a)

$$f(x,y) = \sum_{m \in M} \sum_{i \in I} \sum_{n \in N} \sum_{j \in J} d_{i,j} f_{m,n} x_{m,i} y_{n,j} \text{ (see chapter 3 subsection 3.6.3)}$$

of a solution (X, Y) may be computed as :

$$f(X,Y) = \sum_{i \in I} \sum_{j \in J} \sum_{m \in X_i} \sum_{n \in Y_j} f_{m,n} d_{i,j}$$

Correspondingly, on origin side of I-shape cross-dock that we express by the letter $\tau = o$, the cost incurred by assigning origin $m \in M$ to inbound dock door $i \in I$ for a given partition Y, may be expressed as

$$c^o_{m,i}(Y) = \sum_{j \in J} \sum_{n \in Y_j} f_{m,n} d_{i,j} \tag{5.1a}$$

and on destination side of I-shape cross-dock that we express by the letter $\tau = d$, the cost of assigning destination $n \in N$ to outbound dock door $j \in J$ for a given partition X, may be expressed as

$$c^d_{n,j}(X) = \sum_{i \in I} \sum_{m \in X_i} f_{m,n} d_{i,j} \tag{5.2a}$$

The amount of capacity $S(X_i)$ (respectively $R(Y_j)$) consumed at each inbound (respectively outbound) dock door with respect to the solution (X, Y) is expressed as :

$$S(X_i) = \sum_{m \in X_i} s_m \ , \qquad \forall i \in I$$

$$R(Y_j) = \sum_{n \in Y_j} r_n \ , \qquad \forall j \in J$$

5.2.1 Constructive heuristic to generate an initial solution

We use the following procedure described in **Algorithm 5.1** to build an initial solution. First, the procedure sorts the origins so that their numbers of pallets, s_m, are in descending order of size, and then assigns these origins to the inbound dock doors in a random fashion (line 6) respecting the capacity constraint of these dock doors (line 4). The destinations are then assigned to the outbound dock doors using a greedy procedure in which the destinations are similarly sorted in descending order according to the total number of pallets r_n they receive (line 10). After that, the destinations are assigned one by one to the outbound dock doors following the established order. This latter assignment is accomplished by assigning a destination n to a dock door j associated with the minimum assignment cost $c^d_{n,j}(X)$, where $c^d_{n,j}(X)$ depends on the given assignment of origins (lines 11-17). We have found that sorting the origins and destinations in this simple manner greatly enhances the algorithm's ability to find a feasible initial solution that satisfies the dock doors' capacities, although of course there is no guarantee that the solution will be feasible. Namely, some origins (destinations) may remain non-assigned to inbound (outbound) dock doors. If this happens, non-assigned origins (destinations) are assigned to inbound (outbound) dock doors in a greedy way so that the violation of the capacity constraints is minimized. To measure the violation of the capacity constraints the following function is used :

$$g(X,Y) = \sum_{i \in I} max\{0, S(X_i) - S_i\} + \sum_{j \in J} max\{0, R(Y_j) - R_j\}$$

After that, in order to attain feasibility, we launch a PTS algorithm, whose steps are given in Section 5.3. In this case, the PTS considers $g(X,Y)$ as the objective function and may accept also infeasible solutions. Once a feasible solution is found, it is used as an initial solution for the PTS which works only with feasible solutions and uses the CDAP objective function, $f(X,Y)$ (see Section 5.3 for more details). Starting from this point, a candidate list $\mathcal{N}(X,Y)$ is forced to contain only feasible solutions at each subsequent iteration. The procedure is depicted in **Algorithm 5.1**, it checks the capacity constraints of dock doors.

Algorithm 5.1 Constructive heuristic

Create empty solution : $X_i = \phi$ for all $i \in I$, and $Y_j = \phi$ for all $j \in J$;
Sort the origins $m \in M$ in descending order of their s_m values ;
for each $m \in M$ **do**
  Let $I' = \{i \in I : S(X_i) + s_m \leq S_i\}$ be the set of inbound dock doors that can receive the origin $m \in M$;
  if $I' \neq \phi$ **then**
    Select randomly an inbound dock door $i \in I'$;
    $X_i = X_i \cup \{m\}$;
  end if
end for
Sort the destinations $n \in N$ in descending order of their r_n values ;
for each $n \in N$ **do**
  Let $J' = \{j \in J : R(Y_j) + r_n \leq R_j\}$ be the outbound dock doors that can receive the destination $n \in N$;
  if $J' \neq \phi$ **then**
    Let $j = argmin\{c^d_{n,j'}(X) : j' \in J'\}$ be the dock door with the smallest cost $c^d_{n,j'}(X)$ associated ;
    $Y_j = Y_j \cup \{n\}$;
  end if
end for
Return (X, Y).

5.2.2 Neighborhood structures and move evaluation

A solution (X, Y) corresponds to a partition of the set of origins M and a partition of set of destinations N, respectively. The moves that define the neighborhood structure consist of transferring a truck from one dock door to another, and of exchanging two subsets of trucks between two dock doors. Hence, we define the neighborhood structures of the current solution $\mathcal{N}^{\tau}(X, Y)$ that affect either the origins (side $\tau = o$) or the destinations (side $\tau = d$). For each side $\tau \in \{o, d\}$, we denote by $\bar{\tau}$ the opposite side of τ, i.e., if $\tau = o$ then $\bar{\tau} = d$ and vice-versa. Specifically, we divide the moves into the following two types *Shift* moves and *Swap* moves. It is worth mentioning that we consider only feasible moves when defining the neighborhood structure. However, in the exceptional case where the

solution returned by the initial solution procedure is not feasible, the algorithm accepts only moves that decrease infeasibility until a feasible solution is found. Then, only feasible moves are performed.

5.2.2.1 Shift moves

For a given side (origin side $\tau = o$ or destination side $\tau = d$), a shift move transfers a selected truck (origin or destination) from one dock door to another (inbound or outbound).

For the origin side $\tau = o$, a solution that is a neighbor of the current solution (X, Y) is obtained by shifting an origin $m \in M$ from its current inbound dock door i ($m \in X_i$) to another inbound dock door $i^* \in I - \{i\}$ selected randomly among the k best inbound dock doors, that is, dock doors having the smallest costs $c^o_{m,i^*}(Y)$. More precisely, for each origin $m \in M$, we re-index the inbound dock doors $i' \in I - \{i\}$ so that $c^o_{m,1}(Y) \leq c^o_{m,2}(Y) \leq ... \leq c^o_{m,|I|-1}(Y)$ and let $I^k_m = \{1, ..., k\}$ be the set identifying the inbound dock doors $i' \in I - \{i\}$ with the k smallest values $c^o_{m,i'}(Y)$. A neighboring solution $(X', Y') \in \mathcal{N}^{o,k}_{Shift}(X, Y)$ is obtained by setting $Y' = Y$ (partition Y still fixed) and selecting randomly $i^* \in I^k_m$ and for all $i' \in I$ setting

$$X'_{i'} = \begin{cases} X_i - \{m\} & if \quad i' = i \\ X_{i^*} + \{m\} & if \quad i' = i^* \\ X_i & otherwise \end{cases} \tag{5.3a}$$

Analogously, for the destination side $\tau = d$, a solution in the neighborhood of the current solution (X, Y) is obtained by shifting a destination $n \in N$ from its current outbound dock door j ($n \in Y_j$) to another outbound dock door $j^* \in J - \{j\}$ selected randomly among the k best outbound dock doors, that is, dock doors having the smallest costs c^d_{n,j^*}. For the sake of completeness, we provide definitions of these moves as well: for each destination $n \in N$, we re-index the outbound dock doors $j' \in J - \{j\}$ so that $c^d_{n,1}(X) \leq c^d_{n,2}(X) \leq ... \leq c^d_{n,|J|-1}(X)$ and let $J^k_n = 1, ..., k$ be the set identifying the outbound dock doors $j' \in J - \{j\}$ with the k smallest values $c^d_{n,j'}(X)$. A neighboring solution $(X', Y') \in \mathcal{N}^{d,k}_{Shift}(X, Y)$ is obtained by setting $X' = X$ (partition X still fixed) and selecting randomly $j^* \in J^k_n - \{j\}$ and for all $j' \in J$ setting

$$Y'_{j'} = \begin{cases} Y_j - \{n\} & if \quad j' = j \\ Y_{j^*} + \{n\} & if \quad j' = j^* \\ Y_j & otherwise \end{cases} \tag{5.4a}$$

Remark 1: if $k = 1$, the origin m (respectively the destination n) is transferred to the best i^* inbound dock door (respectively to j^* outbound dock door) , while if $k = |I| - 1$ (respectively $k = |J| - 1$), the origin m (respectively the destination n) is transferred to a randomly selected i^* inbound dock door (respectively j^* outbound dock door).

5.2.2.2 Swap moves

A swap move consists of exchanging trucks between two different dock doors. In our implementation, we consider two groups of swap moves : *elementary swap moves* and *multiple swap moves*. An elementary swap move consists of exchanging two different trucks between two different dock doors, while a multiple swap move consists of exchanging two subsets of trucks between two different dock doors.

Formally, for the origin side $\tau = o$, a neighborhood solution $(X', Y') \in \mathcal{N}^{o,p,q}_{Swap}(X, Y)$ is obtained by setting $Y' = Y$ (partition Y still fixed), choosing $i, i' \in I$ with $i \neq i'$, selecting $P \subseteq X_i$ such that $|P| = p$ and $Q \subseteq X'_i$ such that $|Q| = q$ and for all $h \in I$ setting

$$X'_h = \begin{cases} X_h - P \;\; if \;\; h = i \\ X_h + Q \;\; if \;\; h = i' \\ X_h \;\; otherwise \end{cases} \tag{5.5a}$$

Similarly, for the destination side $\tau = d$, a neighboring solution $(X', Y') \in \mathcal{N}^{d,p,q}_{Swap}(X, Y)$ is obtained by setting $X' = X$ (partition X still fixed), choosing $j, j' \in J$ with $j \neq j'$, selecting $P \subseteq Y_j$ such that $|P| = p$ and $Q \subseteq Y'_j$ such that $|Q| = q$ and for all $h \in J$ setting

$$Y'_h = \begin{cases} Y_h - P \;\; if \;\; h = j \\ Y_h + Q \;\; if \;\; h = j' \\ Y_h \;\; otherwise \end{cases} \tag{5.6a}$$

Remark 2: An elementary swap moves can be derived from the above definition by choosing $p = 1$ and $q = 1$.

The set of neighboring solutions generated by swap moves that affect sets X_i and $X_{i'}$ on origin side (respectively Y_j and $Y_{j'}$ on destination side) will be denoted by $\mathcal{N}^{o,p,q}_{X_{Swap}}(X_i, X_{i'})$ (resp. $\mathcal{N}^{d,p,q}_{Y_{Swap}}(Y_j, Y_{j'})$). Using these definitions, on origin side ($\tau = o$), we have :

$$\mathcal{N}^{o,p,q}_{Swap}(X, Y) = \bigcup_{i,i' \in I, i \neq i'} \mathcal{N}^{o,p,q}_{X_{Swap}}(X_i, X_{i'})$$

and similarly, on destination side ($\tau = d$), we have :

$$\mathcal{N}^{d,p,q}_{Swap}(X,Y) = \bigcup_{j,j' \in J, j \neq j'} \mathcal{N}^{d,p,q}_{Y_{Swap}}(Y_j, Y_{j'})$$

In Section 5.2.2.4, we describe an efficient procedure to explore the swap neighborhoods.

5.2.2.3 Data structures to evaluate and update moves

To efficiently evaluate each move presented in the preceding section we use auxiliary data structures. By move evaluation we mean the change in the objective function caused by executing a certain move on a current solution. Here we present only a method to evaluate efficiently the shift moves, since each swap move (elementary or multiple) can be easily transformed into a set of shift moves.

From equation (3.1a) the objective function value of a solution (X, Y) can be expressed as

$$f(X,Y) = \sum_{i \in I} \sum_{j \in J} \sum_{m \in X_i} \sum_{n \in Y_j} f_{m,n} d_{i,j}$$

Using equation (5.1a), this can be rewritten as

$$f(X,Y) = \sum_{i \in I} \sum_{m \in X_i} c^o_{m,i}(Y) \tag{5.7a}$$

Or equivalently by equation (5.2a) :

$$f(X,Y) = \sum_{j \in J} \sum_{n \in Y_j} c^d_{n,j}(X) \tag{5.8a}$$

Again, we differentiate shift moves that affect origin - inbound dock door assignments ($\tau = o$) and those that affect destination - outbound door assignments ($\tau = d$).

First consider a shift move on origin side ($\tau = o$), that transfers an origin $m \in M$ from its current inbound dock door i ($m \in X_i$) to another inbound dock door $i^* \in I^k_m - \{i\}$. The objective function change produced by this shift move is given by

$$\Delta^o(m, i, i^*) = f(X', Y) - f(X, Y)$$

Using the expressions (5.7a) and (5.8a) we obtain

$$\Delta^o(m, i, i^*) = c^o_{m,i^*}(Y) - c^o_{m,i}(Y) \tag{5.9a}$$

Next consider a shift move on the destination side $\tau = d$, that transfers a destination $n \in N$ from its current inbound dock door j ($n \in Y_j$) to another inbound door $j^* \in J_n^k - \{j\}$. The objective function change produced by this shift move is given by

$$\Delta^o(n, j, j^*) = f(X, Y') - f(X, Y)$$

Similarly, using the expressions (5.7a) and (5.8a) we obtain

$$\Delta^d(n, j, j^*) = c^d_{n,j^*}(X) - c^d_{n,j}(X) \tag{5.10a}$$

As a consequence of the expressions (5.9a) and (5.10a), a shift move can be evaluated in constant time $O(1)$, if we make reference to the two matrices $c^o_{m,i}(Y)$ and $c^d_{n,j}(X)$. Hence, to achieve this constant time computation of the objective function change $\Delta^o(m, i, i^*)$ and $\Delta^d(n, j, j^*)$, we need to update the two matrices $c^o_{m,i}(Y)$ and $c^d_{n,j}(X)$ after each shift move.

Let $c'^o_{m,i}(Y)$ (respectively $c'^d_{n,j}(X)$) be the value at entry (m, i) (respectively (n, j)) in the matrix $c^o_{m,i}(Y)$ (respectively $c^d_{n,j}(X)$) after a shift move. From equations (5.1a) and (5.2a) that provide the definitions of $c^o_{m,i}(Y)$ and $c^d_{n,j}(X)$ respectively, we observe that the execution of a shift move on the origin side, $\tau = o$, affects the matrix $c^d_{n,j}(X)$ and vice versa. More precisely, after a shift move on the origin side, $\tau = o$, we have

$$c'^d_{n,j}(X') = \sum_{i \in I} \sum_{m \in X'_i} d_{i,j} f_{m,n}$$

Using expression (5.3a), we obtain

$$c'^d_{n,j}(X') = c^d_{n,j}(X) + f_{m,n}(d_{i^*,j} - d_{i,j}) \tag{5.11a}$$

Similarly, after a shift move on the side $\tau = d$, we have

$$c'^o_{m,i}(Y') = \sum_{j \in J} \sum_{n \in Y'_j} d_{i,j} f_{m,n}$$

Using expression (5.4a), we obtain

$$c'^o_{m,i}(Y') = c^o_{m,i}(Y) + f_{m,n}(d_{i,j^*} - d_{i,j}) \tag{5.12a}$$

As a consequence, the complexity of updating Δ^o after a shift move on the origin side $\tau = d$ is $O(|N| \times |J|)$ and the complexity of updating Δ^d after a shift move on the destination side $\tau = o$ is $O(|M| \times |I|)$.

5.2.2.4 Efficient exploration of the swap neighborhood

The complexity of the neighborhood $\mathcal{N}^{o,p,q}_{Swap}(X,Y)$ is $O(\sum_{i.i'\in I, i\neq i'} \binom{|X_i|}{p}\binom{|X_{i'}|}{q})$. Consequently, the exhaustive exploration of the union of neighborhoods $\mathcal{N}^{o,p,q}_{Swap}(X,Y), 1 \leq p \leq |X_i|$ and $1 \leq q \leq |X_{i'}|$, which we denote by $\mathcal{N}^{o}_{Swap}(X,Y)$, has the complexity $O(\sum_{i,i'\in I, i\neq i'} 2^{|X_i|+|X_{i'}|})$. However, if each solution in $\mathcal{N}^{o}_{Swap}(X,Y)$ is feasible, then the best solution in this neighborhood can be found by an exploration of smaller complexity, as we demonstrate in the following proposition.

Proposition 5.2.1 *The best solution within the union of swap neighborhoods $\mathcal{N}^{o}_{Swap}(X,Y)$ can be determined with time complexity $O(\sum_{i.i'\in I, i\neq i'} |X_i| + |X_{i'}|)$ if all solutions in $\mathcal{N}^{o}_{Swap}(X,Y)$ are feasible.*

Proof. Consider two sets X_i and $X_{i'}$ and define $X_i^{imp} = \{m \in X_i : \Delta^o(m,i,i') < 0\}$ and $X_{i'}^{imp} = \{m' \in X_{i'} : \Delta^o(m',i',i) < 0\}$. By these definitions the best multiple swap move that affects sets X_i and $X_{i'}$ is the one that exchanges sets X_i^{imp} and $X_{i'}^{imp}$. Denote the solution obtained from such a swap move by $(X^{i,i'}, Y)$. The generation of this solution requires $O(|X_i| + |X_{i'}|)$ operations, since sets X_i^{imp} and $X_{i'}^{imp}$ may be generated in linear time complexity $O(|X_i|)$ and $O(|X_{i'}|)$, respectively. Consequently, the best solution in the neighborhood $\mathcal{N}^{o}_{Swap}(X,Y)$ i.e $(X^*,Y^*) = argmin\{f(X^{i.i'},Y) : i,i' \in I, i \neq i'\}$ may be found with complexity $O(\sum_{i.i'\in I, i\neq i'} |X_i| + |X_i'|)$.□

The preceding result does not hold if there is an infeasible solution in the neighborhood $\mathcal{N}^{o}_{Swap}(X,Y)$. This can be demonstrated by a small example involving only 3 incoming trucks m_1, m_2 and m_3 with loads $s_{m_1} = 3$, $s_{m_2} = 5$ and $s_{m_3} = 8$, respectively. Suppose we have only two incoming dock doors i_1 and i_2 both with capacity $S_{i_1} = S_{i_2} = 10$. Further, in the solution (X,Y), assume trucks m_1 and m_2 with loads 3 and 5 are assigned to the first incoming door i_1 and truck m_3 with load 8 is assigned to the door i_2. Then the neighborhood $\mathcal{N}^{o}_{Swap}(X,Y)$ contains both feasible and infeasible solutions with respect to the capacity constraints. Let $\Delta^o(m_1,i_1,i_2) > 0$, $\Delta^o(m_2,i_1,i_2) < 0$ and $\Delta^o(m_3,i_2,i_1) < 0$. Then if we use the procedure from the preceding proposition, only the move that exchanges trucks m_2 and m_3 between dock doors will be considered as a potential improving move, but this move is infeasible. Consequently, the current solution (X,Y) would be the best solution. However, in the case that $\Delta^o(m_1,i_1,i_2) + \Delta^o(m_2,i_1,i_2) + \Delta^o(m_3,i_2,i_1) < 0$, a swap move that exchanges trucks m_1 and m_2 from one side with a truck m_3 from the other side is an improving move. So, the procedure used in the preceding proposition may fail to reach the best solution if there is an infeasible solution in the neighborhood $N^{o}_{Swap}(X,Y)$.

However, to avoid an exhaustive exploration of the neighborhood $\mathcal{N}^o_{Swap}(X,Y)$, which may be time consuming due to its large cardinality, but to be still able to find near best solution, we propose the following heuristic exploration of the neighborhood $\mathcal{N}^o_{Swap}(X,Y)$. We consider two sets X_i and $X_{i'}$ and sort the origins in X_i (respectively $X_{i'}$) in ascending order with respect to $\Delta^o(m,i,i')$ (respectively $\Delta^o(m',i',i)$). Represent the established order by $X_i = \{m_1, m_2, ..., m_{|X_i|}\}$ and $X'_i = \{m'_1, m'_2..., m'_{|X_{i'}|}\}$, respectively. Then the procedure tries to find the best improving move by exchanging sets $L = \{m_1, m_2, ..., m_p\}, 1 \leq p \leq |X_i|$ and $L' = \{m'_1, m'_2, ..., m'_q\} 1 \leq q \leq |X_{i'}|$. The steps of the procedure are given in **Algorithm 5.2**. As will be shown in the computational results section, this procedure is able to find a solution which is the best solution or close to the best solution, while taking much smaller CPU time than an exhaustive exploration. Henceforth, when we speak of the swap neighborhood we refer to the set of solutions inspected by the procedure in Algorithm 5.2.

Algorithm 5.2 Exploration of Swap Neighbourhood $(o, X_i, X_{i'})$

Sort the origins $m \in X_i$ in ascending order of the values $\Delta^o(m,i,i')$;
Sort the origins $m' \in X_{i'}$ in ascending order of the values $\Delta^o(m',i',i)$;
Set $\mathcal{N}^o_{X_{Swap}}(X_i, X_{i'}) = \phi$; and $L = \phi$;
for each $m \in X_i$ **do**
  $L = L + \{m\}$; $L' = \phi$
  for each $m' \in X_{i'}$ **do**
    $L' = L' + \{m'\}$;
    if $S(X_i) + S(L') - S(L) \leq S_i$ and $S(X_{i'}) + S(L) - S(L') \leq S_{i'}$ **then**
      $(X', Y') = (X, Y)$;
      $X'_i = X'_i + L' - L$;
      $X'_{i'} = X'_{i'} + L - L'$;
      $\mathcal{N}^o_{X_{Swap}}(X_i, X_{i'}) = \mathcal{N}^o_{X_{Swap}}(X_i, X_{i'}) + (X', Y')$;
    end if
  end for
end for
Return $\mathcal{N}^o_{X_{Swap}}(X_i, X_{i'})$;

Hence, the complexity of the procedure that explores the entire swap neighborhood of a solution (X,Y) on the side $\tau = o$ is $O(\sum_{i,i' \in I, i \neq i'} |X_i||X_{i'}| + |X_i| log|X_i| + |X_{i'}| log|X_{i'}|)$.

Remark 3: If there is no infeasible solution in the swap neighborhood of the current

solution the heuristic procedure described in Algorithm 5.2 and the exhaustive exploration procedure return the same solution.

Analogous results hold for the exploration of the neighborhood $\mathcal{N}^d_{Swap}(X, Y)$ and we will not bother to describe them.

5.3 Probabilistic Tabu Search

In this section we present the Probabilistic Tabu Search approaches we use to solve the CDAP. Probabilistic Tabu Search is a variant of the metaheuristic Tabu Search introduced in Glover (1986). The main steps of our PTS procedure for solving the CDAP are presented in Algorithm 5.3. Starting from an initial solution, PTS is run until a predefined stopping criterion is met. The procedure presented in Algorithm 5.1 is used to generate an initial solution and afterwards it uses the following function to evaluate the visited solutions. At each iteration, our PTS approach constructs a candidate list $\mathcal{N}(X, Y)$, selects a solution from it to be the new incumbent solution, updates the tabu list TL, the auxiliary data structures $c^o_{m,i}(Y)$ and $c^d_{n,j}(X)$ (as explained in the preceding section) and the best solution found so far. To construct a candidate list $\mathcal{N}(X, Y)$ and select a new incumbent solution we propose two approaches which lead to two different variants of PTS which we denote PTS1 and PTS2. In both variants the tabu list (TL) (referred to as short term memory in the original tabu search approach) is managed in the simplest way. The old incumbent solution (X, Y) is added to the tabu list and if the size of the list is greater than l, the oldest solution in TL, added before the l most recent iterations, is deleted.

Algorithm 5.3 Probabilistic Tabu Search : General Framework

1: Generate an initial solution (X, Y) using the procedure in Algorithm 5.1
2: Assign any non-assigned trucks in a greedy way using the function $g(X, Y)$;
3: Set $(X^*, Y^*) = (X, Y)$; $TL = \phi$;
4: **while** a stopping criterion is not met **do**
5: **if** (X^*, Y^*) is feasible **then**
6: $F(X, Y) = f(X, Y)$;
7: **else**
8: $F(X, Y) = g(X, Y)$;
9: **end if**

$\mathcal{N}(X,Y) = Construct_candidate_list(X,Y,TL)$;
$(X,Y) = Select_solution(\mathcal{N}(X,Y), F(X,Y))$;
Update matrices $c^o_{mi}(Y)$ and $c^d_{nj}(X)$;
Update tabu list TL;
$(X'',Y'') = argmin\{F(X',Y') : (X',Y') \in \mathcal{N}(X,Y)\}$;
$(X^*,Y^*) = argmin\{F(X'',Y''), F(X^*,Y^*)\}$;
end while
Return (X^*,Y^*);

5.3.1 Probabilistic Tabu Search : Variant 1

The first PTS variant, denoted PTS1, constructs a candidate list $\mathcal{N}(X,Y)$ using Algorithm 5.4. The procedure first selects side $\tau \in \{o,d\}$ at random. After that, it constructs a candidate list of size μ, selecting half of the solutions from the shift neighborhoods $\mathcal{N}^{\tau,k}_{Shift}(X,Y)$ and half of the solutions from the swap neighborhood $\mathcal{N}^{\tau}_{Z_{Swap}}(Z,Z')$ (where $\mathcal{N}^{\tau}_{Z_{Swap}}(Z,Z')$ corresponds either to $\mathcal{N}^{o}_{X_{Swap}}(X_i,X_{i'})$ or $\mathcal{N}^{d}_{Y_{Swap}}(Y_i,Y_{i'})$ depending on the chosen side τ). Solutions from the neighborhoods are chosen based on a random variable p generated in $[0,1]$: if $p \in [0,0.6]$ the best solution from the neighborhood is chosen, if $p \in]0.6,0.8]$ a solution among the b best ones is chosen, and finally if $p \in]0.8,1]$ a random solution is chosen. The procedure considers solutions to be admissible only if they are not in the tabu list TL (see i.e. Algorithm 5.4)

Algorithm 5.4 Candidate list construction in PTS1

Function $Construct_candidate_list((X,Y),\mu,b,TL)$
$\mathcal{N}(X,Y) = \phi$;
Select a side $\tau \in \{o,d\}$ at random;
for 1 to $\frac{\mu}{2}$ **do**
  $p = random(0,1)$;
  if $p \in [0,0.6]$ **then**
    $k^* = 1$
  end if
  if $p \in [0.6,0.8]$ **then**
    $k^* = b$
  end if

11: **if** $p \in [0.8, 1]$ and $\tau = o$ **then**
12: $k^* = |I|$
13: **end if**
14: **if** $p \in [0.8, 1]$ and $\tau = d$ **then**
15: $k^* = |J|$
16: **end if**
17: Select a random solution $(X', Y') \in \mathcal{N}^{\tau,k^*}_{Shift}(X, Y) - TL$;
18: $\mathcal{N}(X, Y) = \mathcal{N}(X, Y) + (X', Y')$;
19: **end for**
20: **for** 1 to $\frac{\mu}{2}$ **do**
21: **if** $\tau = o$ **then**
22: $(Z, Z') = (X_i, X_{i'}), i \neq i', (X_i, X_{i'})$ chosen at random;
23: **end if**
24: **if** $\tau = d$ **then**
25: $(Z, Z') = (Y_j, Y_{j'}), j \neq j', (Y_j, Y_{j'})$ chosen at random;
26: **end if**
27: $p = random(0, 1)$;
28: **if** $p \in [0, 0.6]$ **then**
29: Select the best solution $(X', Y') \in \mathcal{N}^{\tau}_{Z_{Swap}}(Z, Z') - TL$;
30: **end if**
31: **if** $p \in [0.6, 0.8]$ **then**
32: Among b best select a random $(X', Y') \in \mathcal{N}^{\tau}_{Z_{Swap}}(Z, Z') - TL$;
33: **end if**
34: **if** $p \in [0.8, 1]$ **then**
35: Select a random solution $(X', Y') \in \mathcal{N}^{\tau}_{Z_{Swap}}(Z, Z') - TL$;
36: **end if**
37: $\mathcal{N}(X, Y) = \mathcal{N}(X, Y) + (X', Y')$;
38: **end for**
39: **Return** $\mathcal{N}(Y, Y)$;

In our implementation, we do not use any auxiliary data structure or procedure to avoid repetition of solutions in the candidate list. The reason is that the size of the candidate list is chosen to be much smaller than the size of the pool of candidate solutions (see computational results in Section 5.4) and therefore the probability of having repeated

solutions is very small. Moreover, the use of an auxiliary data structure or procedure would slow down the proposed heuristics.

To choose a new incumbent solution, PTS1 uses the procedure in Algorithm 5.5, which simply selects the best solution in the candidate list as the new incumbent.

Algorithm 5.5 Solution Selection in PTS1

Procedure $Select_solution(\mathcal{N}(X,Y), F(X,Y))$
$(X'',Y'') = argmin\{F(X',Y') : (X',Y') \in \mathcal{N}(X,Y)\}$;
Return (X'',Y'');

5.3.2 PTS : Variant 2

The second PTS variant, denoted PTS2, constructs a candidate list $\mathcal{N}(X,Y)$ according to Algorithm 5.6. The procedure first chooses a side $\tau \in \{o,d\}$, at random, as a basis for building the candidate list. Then it adds to the candidate list solutions from the neighborhood $\mathcal{N}^{\tau,1}_{Shift}(X,Y)$. If there is no improving solution available to be added, it proceeds by adding solutions from the neighborhood $\mathcal{N}^{\tau,1,1}_{Swap}(X,Y)$. If still no improving solutions exist to be added, it adds to the candidate list the best solutions from the swap neighborhood $\mathcal{N}^{\tau}_{Swap}(Z,Z')$, which corresponds either to $\mathcal{N}^{o}_{Swap}(X_i,X_{i'})$ or $\mathcal{N}^{d}_{Swap}(Y_i,Y_{i'})$ depending on the chosen side τ. As in the first variant, the procedure considers solutions to be admissible only if they are not in the tabu list TL.

Algorithm 5.6 Candidate list construction in PTS2

Procedure $Construct_candidate_list(X,Y,TL)$
$\mathcal{N}(X,Y) = \phi$;
Select a side $\tau \in \{o,d\}$ at random;
$\mathcal{N}(X,Y) = \mathcal{N}^{\tau,1}_{Shift}(X,Y) - TL$;
if no improving solution is available in $\mathcal{N}(X,Y)$ **then**
  $\mathcal{N}(X,Y) = \mathcal{N}(X,Y) + \mathcal{N}^{\tau,1,1}_{Swap}(X,Y) - TL$;
end if
if no improving solution is available in $\mathcal{N}(X,Y)$ **then** {***multiple swap moves***}
  if $\tau = o$ **then**
    $L = \{(X_i,X_{i'}) : i,i' \in I, i \neq i'\}$;
  end if
  if $\tau = d$ **then**

12: $L = \{(Y_j, Y_{j'}) : j, j' \in J, j \neq j'\}$;
13: **end if**
14: **for** each pair $(Z, Z') \in L$ **do**
15: Select the best solution $(X', Y') \in \mathcal{N}^{\tau}_{Z_{Swap}}(Z, Z') - TL$;
16: $\mathcal{N}(X, Y) = \mathcal{N}(X, Y) + \{(X', Y')\}$;
17: **end for**
18: **end if**
19: **Return** $\mathcal{N}(X, Y)$;

To select a new incumbent solution, PTS2 uses Algorithm 5.7. The procedure first sorts the neighboring solutions of the solution (X, Y) in increasing order with respect to the objective function (line 1). Then if the set of improving neighboring solutions $\mathcal{N}^*(X, Y)$ is not empty we set $\eta^* = min(|\mathcal{N}^*(X, Y)|, b)$, otherwise $\eta^* = b$. After this, in line 7 the procedure selects at random one of the η^* best solutions in the candidate list to be new incumbent solution. This means that in the case where improving solutions exist the choice is made among at most b best improving solutions. On the other hand, if there is no improving solutions, the choice is made among exactly the b best (non-improving solutions) in the candidate list.

Algorithm 5.7 Solution Selection in PTS2

Procedure *Select_solution*$(\mathcal{N}(X, Y), F(X, Y), b)$
1: Sort solutions in $\mathcal{N}(X, Y)$ in increasing order with respect to the function $F(X, Y)$, i.e., $F(X'^1, Y'^1) \leq F(X'^2, Y'^2) \leq ... \leq F(X'^{\eta}, Y'^{\eta})$, where $\eta = |\mathcal{N}(X, Y)|$;
2: Let $\mathcal{N}^*(X, Y) = \{(X'^k, Y'^k) : F(X'^k, Y'^k) < F(X, Y)\}$
3: **if** $\mathcal{N}^*(X, Y) \neq \phi$ **then** $\eta^* = min(|\mathcal{N}^*(X, Y)|, b)$;
4: **else**
5: $\eta^* = b$;
6: **end if**
7: Select a solution (X'', Y'') randomly from the set $(X'^1, Y'^1), (X'^2, Y'^2), ..., (X'^{\eta^*}, Y'^{\eta^*})$
8: **Return** (X'', Y'');

5.4 Computational Results

In this section, we first compare the results of exhaustive and heuristic exploration of the swap neighborhood. The goal is to show that our heuristic exploration yields a good trade-off between solution quality and CPU time in comparison with exhaustive exploration. Following this, we compare our methods with the state-of-the art methods from the literature. Our approaches were implemented in Java and executed on a PC with 16 GB of RAM and using an Intel Xeon E3-1505M v5 processor with 2.80 GHz. For testing purposes, we have used the same benchmark data sets of the chapter 4 that are introduced in Guignard et al. (2012).

Later, the authors generated a new set of large-scale instances in the same manner. In the newly generated instances, the number of origins/destinations is chosen from $\{25, 50, 75, 100\}$ and the number of indock doors/outdock doors is chosen from $\{10, 20, 30, 43\}$. The first set of test problems is referred to as "SetA" and contains 50 instances, while the second (large-scale) set is denoted "SetB" and contains 49 instances. The name of each instance has the format 00x00S00, where the first 00 refers to the number of origins/destinations, the second 00 after x refers to the number of inbound/outbound dock doors and the last 00 after S is the slack. For example, the instance name 8x4S30 refers to an instance with eight origins, eight destinations, four inbound dock doors, four outbound dock doors and slack equal to 30%.

5.4.1 Comparison of exhaustive and heuristic exploration of swap neighborhood

In order to highlight the advantage of using our proposed heuristic exploration of the swap neighborhood presented in Algorithm 5.2, as contrasted to exhaustive exploration, we perform the following test. On each test instance we generate an initial feasible solution using Algorithm 5.1 and perform heuristic and exhaustive exploration of the swap neighborhood starting from this solution.

For comparison purposes we store the best solution value found, the CPU time consumed (in milliseconds) and the number of solutions evaluated by both approaches. Table 5.1 presents the average values obtained over the SetA and SetB instances (Columns 'CPU', 'value' and '#solutions'). In addition, we report the average percentage deviations of solution values found by heuristic exploration from those found by exhaustive exploration

(Column '% dev.'), and the number of instances in each data sets where heuristic and exhaustive exploration reach the same value (Column '#same.').

The outcomes show that the heuristic exploration is significantly faster than the exhaustive one as a result of evaluating significantly fewer solutions (as expressed in the proposition of Section 5.2.2.4). Despite evaluating fewer solutions, it is able to find solutions with a quality only slightly worse than that obtained with an exhaustive exploration, as evidenced by the fact that the average percentage deviations are 0.02% and 0.008% on setA and setB, respectively. In addition, it should be emphasized that on 71 out of 99 instances these two approaches return the same solution as final.

Data Set	Heuristic			Exhaustive			%dev	#same
	CPU(ms)	Value	#solutions	CPU(ms)	Value	#solutions		
SetA	0.01	11419.04	65.84	0.40	11416.80	331.76	0.020	33/50
SetB	3.27	562505.60	2542.98	11960.76	562459.90	51537734.50	0.008	38/49

TABLE 5.1 – Heuristic vs. Exhaustive exploration of swap neighborhood

5.4.2 Comparison with methods from the literature

As a basis for comparison, we refer to the following four leading heuristics from the literature : two local search based heuristics, named LS1 and LS2, the Convex Hull Relaxation (CHR) heuristic proposed by Guignard et al. (2012) and the Lagrangian Relaxation (LR) heuristic proposed by Nassief et al. (2016).

After some tuning, the parameters of our algorithms are set in the following way. Both PTS1 and PTS2, use a stopping criterion that limits the number of iterations performed. For both methods, the limiting number is set to 10^5 on SetA and to 2×10^5 on SetB. The parameter b of the selection procedures is set to 3 and the size of the tabu list is set to $\frac{|M|+|N|+|I|+|J|}{16}$. For PTS1, the size μ of the neighborhood $N(X, Y)$ is set to $\frac{|M|+|N|}{2}$. On each instance, our PTS heuristics are executed 10 times using different random seeds.

In Tables 5.2 and 5.3, we compare the results of PTS1 and PTS2 on SetA instances with the Best Known Solution (BKS) values reported in Guignard et al. (2012) and Nassief et al. (2016). The BKS values in Table 5.2 are found by the LS1, LS2, CHR, CPLEX and LR heuristics, while the BKS values in Table 5.3 are found by LS1 and LS2. Tables 5.2 and 5.3 provide summary results over test classes while detailed results may be found in the Appendix C. By convention, the test class is formed by instances with the same number of

$\|N\| \times \|I\|$	BKS	$\mathcal{M}^{3,0}$	$\mathcal{M}^{2,0}$	PTS1			PTS2		
		CPU(s)	CPU(s)	Best	Avg.	CPU(s)	Best	Avg.	CPU(s)
8x4	5120.8	0.35	0.15	5120.8	5120.8	3.50	5120.8	5120.8	1.10
9x4	5978.2	0.54	0.21	5978.2	5978.2	4.04	5978.2	5978.2	1.03
10x4	6319.8	0.93	0.38	6319.8	6319.8	4.61	6319.8	6319.8	1.10
10x5	6427.8	3.17	1.00	6427.8	6427.8	4.20	6427.8	6427.8	1.46
11x5	7555.6	4.24	1.64	7555.6	7555.6	4.73	7555.6	7555.9	1.55
12x5	7970.2	14.32	3.67	7970.2	7970.2	5.29	7970.2	7970.2	1.53
12x6	10449.8	80.48	26.09	10449.8	10449.8	4.84	10449.8	10453.1	2.06
15x6	13756.4	622.78	132.93	13756.4	13756.4	6.55	13756.4	13773.0	2.29
15x7	14688.8	3843.62	1044.73	14688.8	14688.8	6.13	14688.8	14703.0	2.81
20x10	29171.4	7200.00	7200.00	**29151.2**	**29157.8**	7.89	**29151.2**	29342.0	5.17
Avg	**10743.88**	**768.48**	**745.35**	**10741.86**	**10742.53**	**5.18**	**10741.86**	**10764.39**	**2.01**

TABLE 5.2 – Summary results on "SetA" instances

origins/destinations and inbound/outbound doors. Therefore, the headings of Tables 5.2 and 5.3 are defined as follows. The number of origins/destinations and inbound/outbound dock doors in each class is given in the first column in the form $|N| \times |I|$. The second column is dedicated to BKS values. In columns three and four in Table 5.2 we present the CPU time needed for CPLEX to solve the recent best MIP formulations $\mathcal{M}^{2,0}$ and $\mathcal{M}^{3,0}$ for CDAP (see chapter 4, section 4.6 of computational results), where column three (Column '$\mathcal{M}^{3,0}$') is taken from Nassief et al. (2016) and column four (Column '$\mathcal{M}^{2,0}$') is taken from chapter 4, section 4.6. Remaining columns report the results of our heuristics. On each instance our heuristics were executed 10 times recording the best solution value and the average solution value found in 10 runs, and the average CPU time spent in solving the instance. The averages of these values over the instances from the same test class are reported in Columns 'Best', 'Avg.', and 'CPU', respectively.

In Table 5.4, we report the total number of instances, over each data set, where the first approach in the comparison provides better, equal or worse solutions than the second approach in the comparison. For example, under the header PTS1 vs BKS, we provide the total numbers of instances where PTS1 offers better (Columns 'Best'), equal (Columns 'Equal'), and worse (Columns 'Worse') solution than BKS.

From the results presented for SetA instances in appendix Table B.1, we see that both

$\|N\| \times \|I\|$	BKS	PTS1			PTS2		
		Best	Avg.	CPU(s)	Best	Avg.	CPU(s)
25x10	48446.0	**48268.0**	48287.1	21.68	48280.0	48341.0	12.58
25x20	51741.0	**51533.0**	**51618.4**	19.81	51562.0	52334.2	28.51
50x10	187945.4	**187395.0**	187551.5	72.99	187469.0	188446.8	26.26
50x20	230622.2	**229566.2**	**230233.4**	48.71	231038.0	233009.4	49.41
50x30	264322.3	**262510.0**	**263745.3**	46.43	265606.0	268292.8	84.85
50x43	330661.0	**330285.0**	**332378.3**	47.46	335036.0	341233.4	121.07
75x10	431150.2	**429874.2**	**430538.9**	172.98	430845.2	432162.1	44.84
75x20	513604.6	**511545.2**	**512406.4**	89.49	514356.6	518236.2	73.03
75x30	608476.0	**605108.0**	**606868.6**	83.37	611928.0	616832.8	111.24
100x10	756508.0	**754670.6**	**755528.7**	352.63	755725.0	757630.9	68.74
100x20	933612.6	**929704.0**	**931873.7**	153.61	934695.4	939120.4	103.30
100x30	1113857.0	**1102169.2**	**1105648.0**	130.49	1114188.4	1122055.4	143.05
Avg	**504253.51**	**501137.59**	**502307.05**	**117.41**	**504369.71**	**507363.03**	**70.51**

TABLE 5.3 – Summary results on "SetB" instances

Data Set	PTS1 vs BKS			PTS2 vs BKS			PTS1 vs PTS2		
	Better	Equal	Worse	Better	Equal	Worse	Better	Equal	Worse
SetA	**4**	45	1	**4**	45	1	**0**	50	0
SetB	**49**	0	0	**27**	0	22	**43**	3	3
All	**53**	**45**	**1**	**31**	**45**	**23**	**43**	**53**	**3**

TABLE 5.4 – Comparison of methods in terms of solution quality

algorithms PTS1 and PTS2 only fail to reach the best-known solution value previously reported in the literature on a single instance (i.e., 20x10S15), while they both establish new best-known solution values for four instances. Regarding CPU-time, we observe that PTS2 is more than 2 times faster on average than PTS1. However, the average solution values found by PTS1 are in general better than those of PTS2 and sometimes better than the previously found best-known solution values (see Table B.1 in the Appendix B, for the instances 20x10S5, 20x10S10, 20x10S20, 20x10S30). Furthermore, the results reported (chapter 4) and Nassief et al. (2016) show that instances with up to 15 origins/destinations and 7 indoors/outdoors are optimally solved by the CPLEX MIP solver within the maximum of 492 seconds. On these instances, the PTS algorithms were able to reach all optimal solutions in less than 8 seconds. Moreover, for instances with 20

origins/destinations and 10 indoors outdoors, CPLEX did not reach optimal solutions in two hours while the PTS algorithms provide better results in less than 10 seconds. This comparison with CPLEX results indicates the merit of using probabilistic tabu search for solving hard optimization problem such as CDAP.
On the other hand, on SetB, PTS1 outperforms the state-of-the-art methods, LS1 and LS2, in finding the best-found solution. On several instances even the average solution values reported by PTS1 are better than the best solution values found by LS1 and LS2 (see Table B.2 in the Appendix C). Comparing the best solutions found by PTS1 and PTS2, we see that PTS2 is better than PTS1 on 3 instances, ties with PTS1 on 3 instances, while on the remaining 43 instances PTS1 is better than PTS2. Comparing the average solution values of PTS1 and PTS2, we see that PTS1 outperforms PTS2 on 48 out of 49 instances (see Table B.2 in the Appendix C). We also see that PTS2 performs very well on instances with 10 indoors/outdoors where it obtains results very close to those of PTS1, while consuming very little CPU time compared to PTS1. In addition, compared to the BKS method, PTS2 provides better solutions on 27 instances out of 49 instances.

Previous findings indicate that PTS1 outperforms the other approaches in terms of solution quality. In order to check if this superior performance is significant or not, we perform the Wilcoxon signed rank test, see e.g., Wilcoxon (1945). The resulting p-values of 1.1101×10^{-9}, 1.1101×10^{-7}, and 6.7951×10^{-9} when comparing PTS1 vs LS1, PTS1 vs LS2, and PTS1 vs PTS2, respectively, reveal the significance of the differences that establish the superiority of PTS1.

Comparing the running times of PTS1 and PTS2, we see that : PTS2 is better on instances with 10 dock doors (instances with reduced neighborhood structure) ; the methods are very similar on instances with 20 dock doors (where sometimes PTS1 is better than PTS2) ; PTS1 in general outperforms PTS2 in terms of running time on instances with 30 doors and on one instance with 43 doors, even though the neighborhood considered in PTS1 is much larger than the neighborhood considered in PTS2. On the other hand, the running times of our methods are much better than those of LS1 and LS2 (see Table B.2 in the Appendix C).

In the light of these results, we conclude that PTS2 is more suitable for small problems, while PTS1 is more suitable for large problems. We conjecture that the good performance of PTS1 on large instances may be explained by the fact that it explores a smaller neighborhood than PTS2, which enables it to achieve a better tradeoff between intensification and diversification than PTS2 within the same amount of time.

5.5 Conclusion

In this chapter, two Probabilistic Tabu Search heuristics have been presented to tackle the Cross-dock Door Assignment Problem (CDAP) that has important applications in supply chain management. Their main differences are embodied in the ways they generate candidate lists and accept solutions in each iteration. Our methods are implemented in an innovative manner using a new large swap neighborhood structure for solving either the CDAP or similar problems. In addition, a supporting heuristic is proposed to explore the large swap neighborhood efficiently. The merit of our proposed algorithms is assessed by comparing them with the most effective methods from the literature on two benchmark data sets, likewise from the literature. Computational tests disclose that our approaches significantly outperform the previously proposed methods by reaching 45 previous best-known solutions and establishing 53 new best-known solutions over the full set of 99 instances. In addition, the CPU time consumed by our approaches is substantially less than consumed by the previous state-of-the art methods.

Chapter 6

Conclusions and Future works

Conclusions

All along this PhD thesis, our works focused on an NP-hard combinatorial optimization problem referred to "Cross-dock Door Assignment Problem (CDAP)" also known as "Truck-to-dock Door Assignment Problem (TDAP)".

Numerous studies have been performed and improved in order to solve the CDAP. Because of the NP-hard character of the CDAP, most of researchers have turned to the implementation of heuristics based algorithm. According to the best of our knowledge, very few exact methods and mathematical formulations for CDAP have been proposed until now.

First, we focused on a broad analysis of two standard mathematical formulations proposed in the literature for CDAP, the first one is a standard quadratic formulation and the second one is the standard linear model. Further, other related mathematical formulations have been analyzed. However, those models remain too weak to solve the problem even for small instances. Our main idea has been to exploit those two standard models to mathematically formulate new linear models for the CDAP. Using mathematical theories of reformulation and linearization, we have mathematically formulated eight new nonstandard MILP models for the CDAP that we have implemented and solved using CPLEX solver. We then showed equivalence between those new nonstandard linear models as well as their equivalence with those in literature. An exhaustive empirical study has been conducted on benchmark data sets from literature. The results of computational experiments have proven that one of those eight new nonstandard linear models for

CDAP is the best even compared to the those in literature. Afterwards, we carried out a Lagrangian Relaxation (LR) procedure with the aim to produce new lower bounds on optimal value. We selected the model for Lagrangian Relaxation on the basis of a good compromise between LP lower bound and CPU time consumption given by that model against MIP models performing the same LP lower bound. The Lagrangian dual is solved using the sub-gradient optmization algorithm. Computational experiments proved that the Lagrangian dual improved significantly LP lower bound and the lower bound given by Lagrangian Relaxation from literature.

Even though some of the proposed new linear models for the CDAP are good and our Lagrangian algorithm gives good lower bounds, those methods remain weak for large scale instances. Thereby, we proposed and implemented a heuristic based on Tabu Search (TS) to efficiently solve large scale instances of CDAP. We focused more precisely on Probabilistic Tabu Search (PTS). Depending on how candidate list of new solutions is built and the way a new incumbent solution is selected from the candidate list of solutions, we have proposed two approaches that leaded to two different variants of PTS that we denoted by PTS1 and PTS2 . The merit of our proposed PTS is assessed by comparing PTS1 and PTS2 with the most effective and recent algorithms from literature review on two sets of benchmark data set. Computational experiments demonstrated the efficiency of our algorithms.

Future works directions

The MIP models presented in the chapter 4 are applicable to pure cross-docks with fixed mode of dock doors in a pre-distribution environment without arrival and departure restrictions. Hence, a possible future research direction is to consider a less restrictive model that also takes into account availability of trucks, stochasticity of arrivals, uncertainty in contents of trucks.

Future works may also include adapting the models that we have presented in the chapter 4 to handle other cross-docking facility shapes and to carry out associated theoretical and empirical analysis.

Considering the PTS proposed in chapter 5, future work will examine the ways to exploit advantages of each of the proposed PTS heuristics by combining their best features within one scheme to yield other variants for solving CDAP. We envision that benefits

will accrue from future works that study the impact of cross-docking facilities on vehicle scheduling in a supply chain by combining CDAP with the vehicle routing problem.

To benefit the advantages of models formulation and algorithms, a possible future research direction is to propose a hybrid approach that combines the best MIP model from those identified in chapter 4 with our newly proposed PTS and the combine PTS with the sub-gradient optimization method to tackle big instances in a Lagrangian fashion.

Appendix A

Detailed results of computational experiments on new MIP Models and Lagrangian Relaxation for the CDAP

In this annex, we provide detailed results of computational experiments carried out on entire data sets referred to "SetA" for all mathematical models and Lagrangian Relaxation presented in the chapter 4. The mathematical models are run for several cases of test, once when integrality requirements on the 3 variables of the MIP models are imposed (see e.g.,Tables A.1, A.2, A.6a and A.6b), second case when the used linearization variable $z_{m,i,n,j}$ is the only one relaxed (see e.g., Tables A.3, A.4, A.7a and A.7b) and the last case we tested the models for LP relaxations (see e.g., Tables A.5). For Lagrangian Relaxation, detailed results are presented in Table A.8. The sign '-' in the Tables A.1, A.2, A.3, A.4 and A.8 implies that CPLEX could not provide a feasible solution within the imposed time limit. In the Tables A.6 and A.7 we perform Wilcoxon signed rank statistical tests for all of the models that are able to provide a feasible solution for tested instance. The tests are carried out to compare the superiority between models for each pair of models both for the solution quality and runtime.

A.1 Comparison of models - Integrality requirement on variables $x_{m,i}$, $y_{n,j}$ and $z_{m,i,n,j}$ imposed

A.1.1 Quadratic model Q and Linear Models $\mathcal{M}^{0,0}$, $\mathcal{M}^{0,1}$, $\mathcal{M}^{1,0}$, $\mathcal{M}^{1,1}$, $\mathcal{M}^{2,0}$ for each class of instances and general average

Instances	Q			$\mathcal{M}^{0,0}$			$\mathcal{M}^{0,1}$			$\mathcal{M}^{1,0}$			$\mathcal{M}^{1,1}$			$\mathcal{M}^{2,0}$		
	Solution	Time	Gap	Solution	Time	Gap	Solution	Time	Gap	Solution	Time	Gap	Solution	Time	Gap	Solution	Time	Gap
8x4S30	5063	1.016	0.000	5063	8.140	0.000	5063	4.590	0.000	5063	0.760	0.000	5063	4.688	0.000	5063	0.160	0.000
8x4S20	5086	1.281	0.000	5086	7.040	0.000	5086	1.968	0.000	5086	0.560	0.000	5086	2.156	0.000	5086	0.140	0.000
8x4S15	5112	1.484	0.000	5112	1.281	0.000	5112	1.812	0.000	5112	0.610	0.000	5112	3.094	0.000	5112	0.140	0.000
8x4S10	5169	1.703	0.000	5169	1.320	0.000	5169	2.078	0.000	5169	0.720	0.000	5169	4.250	0.000	5169	0.140	0.000
8x4S5	5174	1.891	0.000	5174	1.150	0.000	5174	1.204	0.000	5174	0.830	0.000	5174	3.690	0.000	5174	0.160	0.000
8x4	**5120.8**	**1.475**	**0.000**	**5120.8**	**3.786**	**0.000**	**5120.8**	**2.330**	**0.000**	**5120.8**	**0.696**	**0.000**	**5120.8**	**3.576**	**0.000**	**5120.8**	**0.148**	**0.000**
9x4S30	5904	3.000	0.000	5904	60.970	0.000	5904	47.046	0.000	5904	1.170	0.000	5904	11.719	0.000	5904	0.280	0.000
9x4S20	5937	3.750	0.000	5937	22.450	0.000	5937	15.172	0.000	5937	1.310	0.000	5937	5.953	0.000	5937	0.190	0.000
9x4S15	5976	4.031	0.000	5976	3.890	0.000	5976	11.719	0.000	5976	1.450	0.000	5976	5.375	0.000	5976	0.160	0.000
9x4S10	6027	4.266	0.000	6027	4.360	0.000	6027	3.563	0.000	6027	1.280	0.000	6027	9.812	0.000	6027	0.230	0.000
9x4S5	6047	4.484	0.000	6047	2.560	0.000	6047	2.296	0.000	6047	1.050	0.000	6047	9.406	0.000	6047	0.190	0.000
9x4	**5978.2**	**3.906**	**0.000**	**5978.2**	**18.846**	**0.000**	**5978.2**	**15.959**	**0.000**	**5978.2**	**1.252**	**0.000**	**5978.2**	**8.453**	**0.000**	**5978.2**	**0.210**	**0.000**
10x4S30	6193	6.937	0.000	6193	486.840	0.000	6193	1785.390	0.000	6193	1.300	0.000	6193	14.266	0.000	6193	0.280	0.000
10x4S20	6267	10.219	0.000	6267	2226.250	0.000	6267	102.330	0.000	6267	1.480	0.000	6267	20.266	0.000	6267	0.360	0.000
10x4S15	6296	13.031	0.000	6296	200.760	0.000	6296	112.234	0.000	6296	1.520	0.000	6296	22.328	0.000	6296	0.280	0.000
10x4S10	6325	14.890	0.000	6325	46.970	0.000	6325	313.469	0.000	6325	1.720	0.000	6325	32.718	0.000	6325	0.380	0.000
10x4S5	6518	15.266	0.000	6518	5.110	0.000	6518	5.000	0.000	6518	2.340	0.000	6518	47.391	0.000	6518	0.580	0.000
10x4	**6319.8**	**12.069**	**0.000**	**6319.8**	**593.186**	**0.000**	**6319.8**	**463.685**	**0.000**	**6319.8**	**1.672**	**0.000**	**6319.8**	**27.394**	**0.000**	**6319.8**	**0.376**	**0.000**
10x5S30	6308	46.670	0.000	6308	7200	0.144	6308	2342.703	0.000	6308	3.770	0.000	6308	48.406	0.000	6308	0.750	0.000
10x5S20	6342	75.766	0.000	6342	5290.090	0.000	6342	7200	0.053	6342	3.770	0.000	6342	66.375	0.000	6342	0.720	0.000
10x5S15	6397	96.090	0.000	6397	4078.950	0.000	6397	1430.656	0.000	6397	7.140	0.000	6397	113.766	0.000	6397	0.890	0.000

10x5S10	6476	107.516	0.000	6476	921.290	0.000	6476	1692.734	0.000	6476	9.380	0.000	6476	95.156	0.000	6476	1.550	0.000
10x5S5	6616	110.547	0.000	6616	140.640	0.000	6616	69.610	0.000	6616	7.50s	0.000	6616	77.344	0.000	6616	1.090	0.000
10x5	**6427.8**	**87.318**	**0.000**	**6427.8**	**3526.194**	**0.029**	**6427.8**	**2547.141**	**0.011**	**6427.8**	**6.015**	**0.000**	**6427.8**	**80.209**	**0.000**	**6427.8**	**1.000**	**0.000**
11x5S30	7468	1642.047	0.000	7468	7200	0.286	7420	2960.140	0.000	7420	5.380	0.000	7420	84.797	0.000	7420	1.130	0.000
11x5S20	7439	1841.641	0.000	7439	7200	0.215	7439	919.157	0.000	7439	5.660	0.000	7439	45.000	0.000	7439	0.910	0.000
11x5S15	7543	2049.750	0.000	7543	7200	0.141	7542	7200.000	0.094	7535	7.220	0.000	7535	156.671	0.000	7535	1.780	0.000
11x5S10	7572	2074.469	0.000	7572	5300.109	0.000	7572	5090.563	0.000	7572	5.470	0.000	7572	197.234	0.000	7572	1.420	0.000
11x5S5	7812	2078.730	0.000	7812	125.375	0.000	7812	366.187	0.000	7812	6.450	0.000	7812	276.960	0.000	7812	2.980	0.000
11x5	**7566.8**	**1937.327**	**0.000**	**7566.8**	**5405.097**	**0.128**	**7557**	**3307.209**	**0.019**	**7555.6**	**6.036**	**0.000**	**7555.6**	**152.132**	**0.000**	**7555.6**	**1.644**	**0.000**
12x5S30	8017	7200,000	0.140	8017	7200	0.371	7964	7200.000	0.351	7923	15.440	0.000	7923	320.860	0.000	7923	4.640	0.000
12x5S20	7992	6455.703	0.000	7992	7200	0.259	7939	7200.000	0.263	7939	16.360	0.000	7939	266.890	0.000	7939	3.910	0.000
12x5S15	7999	3625.188	0.000	7999	7200	0.249	7939	5336.187	0.000	7939	12.910	0.000	7939	228.734	0.000	7939	3.390	0.000
12x5S10	8008	2209.593	0.000	8008	7200	0.262	7978	1576.938	0.000	7978	12.560	0.000	7978	376.781	0.000	7978	3.000	0.000
12x5S5	8072	96.172	0.000	8072	7200	0.144	8072	452.546	0.000	8072	14.730	0.000	8072	320.937	0.000	8072	3.410	0.000
12x5	**8017.6**	**3917.331**	**0.028**	**8017.6**	**7200**	**0.257**	**7978.4**	**4353.134**	**0.123**	**7970.2**	**14.400**	**0.000**	**7970.2**	**302.840**	**0.000**	**7970.2**	**3.670**	**0.000**
12x6S30	10276	7200	0.232	10276	7200	0.394	10357	7200	0.457	10228	126.490	0.000	10228	1081.235	0.000	10228	9.470	0.000
12x6S20	10396	348.219	0.000	10396	7200	0.419	10452	7200	0.409	10312	42.950	0.000	10312	650.234	0.000	10312	37.750	0.000
12x6S15	10420	7200	0.054	10420	7200	0.293	10413	7200	0.384	10362	62.230	0.000	10362	87.313	0.000	10362	5.170	0.000
12x6S10	10480	7200	0.034	10480	7200	0.243	10456	4273.516	0.000	10456	107.250	0.000	10456	1135.890	0.000	10456	5.590	0.000
12x6S5	10894	760.265	0.000	10894	7200	0.213	10891	877.531	0.000	10891	96.980	0.000	10891	1393.453	0.000	10891	72.480	0.000
12x6	**10493.2**	**4541.697**	**0.064**	**10493.2**	**7200**	**0.313**	**10513.8**	**5350.209**	**0.250**	**10449.8**	**87.180**	**0.000**	**10449.8**	**869.625**	**0.000**	**10449.8**	**26.092**	**0.000**
15x6S30	13712	7200	0.386	13850	7200	0.619	13722	7200	0.618	13567	389.730	0.000	13567	3451.828	0.000	13567	27.980	0.000
15x6S20	-	-	-	14013	7200	0.628	13916	7200	0.569	13720	966.050	0.000	13720	5432.985	0.000	13720	270.550	0.000
15x6S15	13946	7200	0.474	-	-	-	13951	7200	0.566	13765	405.730	0.000	13765	6511.375	0.000	13765	147.780	0.000
15x6S10	13956	7200	0.446	-	-	-	14053	7200	0.648	13803	582.450	0.000	13803	5239.437	0.000	13803	20.560	0.000
15x6S5	13927	7200	0.061	14136	7200	0.631	14129	7200	0.595	13927	897.750	0.000	14042	7200	0.025	13927	197.800	0.000
15x6	**-**	**-**	**-**	**-**	**-**	**-**	**13954.2**	**7200**	**0.599**	**13756.4**	**648.342**	**0.000**	**13779.4**	**5567.125**	**0.005**	**13756.4**	**132.934**	**0.000**
15x7S30	14630	7200	0.583	-	-	-	14657	7200	0.792	14409	2136.470	0.000	14469	7200	0.023	14409	504.840	0.000
15x7S20	14808	7200	0.585	-	-	-	14837	7200	0.655	14514	1795.580	0.000	14723	7200	0.043	14514	680.580	0.000
15x7S15	14916	7200	0.596	14906	7200	0.596	-	-	-	14657	2251.700	0.000	14797	7200	0.045	14657	417.660	0.000
15x7S10	15053	7200	0.584	15087	7200	0.617	-	-	-	14810	3789.730	0.000	14929	7200	0.045	14810	1352.470	0.000
15x7S5	15179	7200	0.169	15355	7200	0.697	-	-	-	15054	2038.280	0.000	15460	7200	0.080	15054	2268.090	0.000
15x7	**14917.2**	**7200**	**0.503**	**-**	**-**	**-**	**-**	**-**	**-**	**14688.8**	**2402.352**	**0.000**	**14875.6**	**7200**	**0.047**	**14688.8**	**1044.728**	**0.000**

20x10S30	29219	7200	0.802	30756	7200	0.954	30522	7200	0.952	29458	7200	0.124	34666	7200	0.257	29081	7200	0.089
20x10S20	29219	7200	0.884	30928	7200	0.970	30769	7200	0.938	29754	7200	0.130	34162	7200	0.240	29609	7200	0.099
20x10S15	29973	7200	0.842	30844	7200	0.979	31278	7200	0.980	30156	7200	0.145	-	-	-	29967	7200	0.130
20x10S10	29766	7200	0.812	31388	7200	1.000	31557	7200	0.953	30039	7200	0.142	-	-	-	29880	7200	0.099
20x10S5	30824	7200	0.836	32442	7200	1.000	33062	7200	0.981	31420	7200	0.186	-	-	-	30604	7200	0.110
20x10	**29800.2**	**7200**	**0.835**	**31271.6**	**7200**	**0.980**	**31437.6**	**7200**	**0.961**	**30165.4**	**7200**	**0.145**	**-**	**-**	**-**	**29828.2**	**7200**	**0.105**
Average	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**10843.28**	**1057.831**	**0.015**	**-**	**-**	**-**	**10809.56**	**841.080**	**0.011**

TABLE A.1 – Quadratic model Q and linear models $\mathcal{M}^{0,0}, \mathcal{M}^{0,1}, \mathcal{M}^{1,0}, \mathcal{M}^{1,1}$ and $\mathcal{M}^{2,0}$

A.1.2 Linear Models $\mathcal{M}^{2,1}$, $\mathcal{M}^{3,0}$, $\mathcal{M}^{3,1}$, $\mathcal{M}'^{0,0}\mathcal{M}'^{0,1}$ and $\mathcal{M}'^{1,1}$ for each class of instances and general average

Instances	$\mathcal{M}^{2,1}$			$\mathcal{M}^{3,0}$			$\mathcal{M}^{3,1}$			$\mathcal{M}'^{0,0}$			$\mathcal{M}'^{0,1}$			$\mathcal{M}'^{1,1}$		
	Solution	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**
8x4S30	5063	1.281	0.000	5063	0.340	0.000	5063	1.250	0.000	5063	1.016	0.000	5063	7.391	0.000	5063	1.485	0.000
8x4S20	5086	0.718	0.000	5086	0.360	0.000	5086	0.735	0.000	5086	0.313	0.000	5086	6.953	0.000	5086	1.594	0.000
8x4S15	5112	1.047	0.000	5112	0.340	0.000	5112	1.172	0.000	5112	0.250	0.000	5112	1.953	0.000	5112	2.750	0.000
8x4S10	5169	1.781	0.000	5169	0.370	0.000	5169	1.672	0.000	5169	0.234	0.000	5169	1.172	0.000	5169	2.937	0.000
8x4S5	5174	1.110	0.000	5174	0.360	0.000	5174	1.328	0.000	5174	0.172	0.000	5174	0.672	0.000	5174	3.672	0.000
8x4	**5120.8**	**1.187**	**0.000**	**5120.8**	**0.354**	**0.000**	**5120.8**	**1.231**	**0.000**	**5120.8**	**0.397**	**0.000**	**5120.8**	**3.628**	**0.000**	**5120.8**	**2.488**	**0.000**
9x4S30	5904	2.094	0.000	5904	0.343	0.000	5904	4.282	0.000	5904	1.047	0.000	5904	23.922	0.000	5904	2.578	0.000
9x4S20	5937	3.546	0.000	5937	0.656	0.000	5937	2.562	0.000	5937	0.390	0.000	5937	10.375	0.000	5937	4.250	0.000
9x4S15	5976	1.954	0.000	5976	0.469	0.000	5976	2.516	0.000	5976	0.328	0.000	5976	9.031	0.000	5976	4.406	0.000
9x4S10	6027	3.125	0.000	6027	0.656	0.000	6027	4.078	0.000	6027	0.359	0.000	6027	4.406	0.000	6027	5.157	0.000
9x4S5	6047	2.968	0.000	6047	0.562	0.000	6047	6.140	0.000	6047	0.266	0.000	6047	3.016	0.000	6047	6.812	0.000
9x4	**5978.2**	**2.737**	**0.000**	**5978.2**	**0.537**	**0.000**	**5978.2**	**3.916**	**0.000**	**5978.2**	**0.478**	**0.000**	**5978.2**	**10.150**	**0.000**	**5978.2**	**4.641**	**0.000**
10x4S30	6193	3.985	0.000	6193	0.578	0.000	6193	3.750	0.000	6193	4.531	0.000	6193	233.594	0.000	6193	7.125	0.000
10x4S20	6267	5.875	0.000	6267	0.719	0.000	6267	4.032	0.000	6267	2.609	0.000	6267	120.312	0.000	6267	20.656	0.000
10x4S15	6296	3.859	0.000	6296	0.985	0.000	6296	4.968	0.000	6296	2.063	0.000	6296	53.484	0.000	6296	15.016	0.000
10x4S10	6325	4.516	0.000	6325	0.594	0.000	6325	6.360	0.000	6325	1.735	0.000	6325	30.516	0.000	6325	19.562	0.000

10x4S5	6518	6.515	0.000	6518	1.750	0.000	6518	8.437	0.000	6518	0.438	0.000	6518	6.250	0.000	6518	19.516	0.000
10x4	**6319.8**	**4.950**	**0.000**	**6319.8**	**0.925**	**0.000**	**6319.8**	**5.509**	**0.000**	**6319.8**	**2.275**	**0.000**	**6319.8**	**88.831**	**0.000**	**6319.8**	**16.375**	**0.000**
10x5S30	6308	15.969	0.000	6308	1.750	0.000	6308	36.813	0.000	6308	24.516	0.000	6308	7200	0.052	6308	72.766	0.000
10x5S20	6342	16.375	0.000	6342	3.279	0.000	6342	43.234	0.000	6342	28.375	0.000	6342	7200	0.045	6342	72.125	0.000
10x5S15	6397	17.672	0.000	6397	2.344	0.000	6397	57.720	0.000	6397	19.062	0.000	6397	1833.047	0.000	6397	115.031	0.000
10x5S10	6476	62.906	0.000	6476	3.328	0.000	6476	104.593	0.000	6476	7.578	0.000	6476	52.797	0.000	6476	127.422	0.000
10x5S5	6616	47.250	0.000	6616	5.156	0.000	6616	72.703	0.000	6616	1.359	0.000	6616	56.468	0.000	6616	105.125	0.000
10x5	**6427.8**	**32.034**	**0.000**	**6427.8**	**3.171**	**0.000**	**6427.8**	**63.013**	**0.000**	**6427.8**	**16.178**	**0.000**	**6427.8**	**3268.462**	**0.019**	**6427.8**	**98.494**	**0.000**
11x5S30	7420	26.593	0.000	7420	2.312	0.000	7420	47.047	0.000	7420	1144.422	0.000	7428	7200	0.056	7420	128.969	0.000
11x5S20	7439	22.204	0.000	7439	3.234	0.000	7439	76.672	0.000	7439	308.235	0.000	7439	770.875	0.000	7439	126.187	0.000
11x5S15	7535	60.594	0.000	7535	3.875	0.000	7535	102.719	0.000	7535	42.140	0.000	7535	376.390	0.000	7535	186.641	0.000
11x5S10	7572	108.859	0.000	7572	4.687	0.000	7572	143.516	0.000	7572	28.172	0.000	7572	154.610	0.000	7572	319.390	0.000
11x5S5	7812	222.980	0.000	7812	7.094	0.000	7812	221.234	0.000	7812	7.610	0.000	7812	185.375	0.000	7812	291.953	0.000
11x5	**7555.6**	**88.246**	**0.000**	**7555.6**	**4.240**	**0.000**	**7555.6**	**118.238**	**0.000**	**7555.6**	**306.116**	**0.000**	**7557.2**	**1737.450**	**0.011**	**7555.6**	**210.628**	**0.000**
12x5S30	7923	113.765	0.000	7923	38.047	0.000	7923	163.375	0.000	7971	7200	0.135	-	-	-	7923	1516.469	0.000
12x5S20	7939	149.703	0.000	7939	7.922	0.000	7939	172.484	0.000	7939	2088.500	0.000	7939	2902.703	0.000	7939	874.234	0.000
12x5S15	7939	163.297	0.000	7939	9.797	0.000	7939	217.344	0.000	7939	3534.954	0.000	7939	1371.703	0.000	7939	559.703	0.000
12x5S10	7978	200.296	0.000	7978	6.875	0.000	7978	221.078	0.000	-	-	-	7978	1454.594	0.000	7978	777.641	0.000
12x5S5	8072	229.891	0.000	8072	8.953	0.000	8072	331.297	0.000	8072	20.328	0.000	8072	5735.812	0.000	8072	765.656	0.000
12x5	**7970.2**	**171.390**	**0.000**	**7970.2**	**14.319**	**0.000**	**7970.2**	**221.116**	**0.000**	**-**	**-**	**-**	**-**	**-**	**-**	**7970.2**	**898.741**	**0.000**
12x6S30	10228	260.750	0.000	10228	95.250	0.000	10228	605.078	0.000	-	-	-	10276	7200	0.336	10228	5770.203	0.000
12x6S20	10312	416.578	0.000	10312	64.609	0.000	10312	719.297	0.000	-	-	-	10388	7200	0.405	10312	7200	0.020
12x6S15	10362	348.047	0.000	10362	37.578	0.000	10362	759.938	0.000	-	-	-	10582	7200	0.392	10362	3408.453	0.000
12x6S10	10456	609.422	0.000	10456	81.469	0.000	10456	1115.046	0.000	-	-	-	10500	7200	0.314	10456	2948.844	0.000
12x6S5	10891	1045.265	0.000	10891	123.484	0.000	10891	1368.704	0.000	10891	11.593	0.000	10891	2624	0.000	10891	1203.891	0.000
12x6	**10449.8**	**536.012**	**0.000**	**10449.8**	**80.478**	**0.000**	**10449.8**	**913.613**	**0.000**	**10891**	**11.593**	**0.000**	**10527.4**	**6285**	**0.289**	**10449.8**	**4106.278**	**0.004**
15x6S30	13567	1516.781	0.000	13567	687.437	0.000	13567	3323.718	0.000	13776	7200	0.524	13809	7200	0.624	13685	7200	0.034
15x6S20	13720	5090.156	0.000	13720	818.453	0.000	13805	7200	0.017	13922	7200	0.468	13815	7200	0.504	13873	7200	0.036
15x6S15	13765	5129.641	0.000	13765	406.375	0.000	13765	7005.172	0.000	13911	7200	0.407	13861	7200	0.545	13844	7200	0.031
15x6S10	13803	4759.063	0.000	13803	449.500	0.000	13803	4475.172	0.000	-	-	-	-	-	-	13860	7200	0.027
15x6S5	13940	7200.000	0.009	13927	752.141	0.000	13983	7200	0.024	13960	7200	0.314	14331	7200	0.625	14076	7200	0.045
15x6	**13759**	**4739.128**	**0.002**	**13756.4**	**622.781**	**0.000**	**13784.6**	**5840.812**	**0.008**	**-**	**-**	**-**	**-**	**-**	**-**	**13867.6**	**7200**	**0.035**
15x7S30	14409	7200	0.004	14409	2711.282	0.000	14510	7200	0.024	-	-	-	14724	7200	0.653	14665	7200	0.055

15x7S20	14679	7200	0.029	14514	3227.375	0.000	14583	7200	0.027	-	-	-	15001	7200	0.712	14792	7200	0.059
15x7S15	14742	7200	0.036	14657	4199.359	0.000	14798	7200	0.050	14855	7200	0.508	15056	7200	0.644	14941	7200	0.067
15x7S10	15113	7200	0.062	14810	4628.078	0.000	14923	7200	0.046	14843	7200	0.446	14867	7200	0.581	14941	7200	0.059
15x7S5	15443	7200	0.068	15054	4452.016	0.000	14381	7200	0.072	15190	7200	0.365	15278	7200	0.665	15486	7200	0.083
15x7	**14877.2**	**7200**	**0.040**	**14688.8**	**3843.622**	**0.000**	**14639**	**7200**	**0.044**	**-**	**-**	**-**	**14985.2**	**7200**	**0.651**	**14965**	**7200**	**0.065**
20x10S30	34095	7200	0.237	29415	7200	0.119	34252	7200	0.248	29344	7200	0.828	30125	7200	0.937	30069	7200	0.135
20x10S20	34886	7200	0.255	29941	7200	0.125	35010	7200	0.259	29346	7200	0.845	30242	7200	0.942	31759	7200	0.177
20x10S15	34353	7200	0.240	29934	7200	0.115	35565	7200	0.267	29666	7200	0.805	31278	7200	0.949	31327	7200	0.164
20x10S10	34563	7200	0.241	30322	7200	0.120	35308	7200	0.258	29527	7200	0.833	31209	7200	0.951	30933	7200	0.147
20x10S5	-	-	-	30409	7200	0.141	-	-	-	-	-	-	32681	7200	0.975	31248	7200	0.151
20x10	**-**	**-**	**-**	**30004.2**	**7200**	**0.124**	**35033.75**	**7200**	**0.258**	**29470.75**	**-**	**-**	**31107**	**7200**	**0.951**	**31067.2**	**7200**	**0.155**
Average	**-**	**-**	**-**	**10827.16**	**1177.042**	**0.012**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**10972,2**	**2693.764**	**0.023**

TABLE A.2 – Linear Models $\mathcal{M}^{2,1}$, $\mathcal{M}^{3,0}$, $\mathcal{M}^{3,1}$, $\mathcal{M}'^{0,0}\mathcal{M}'^{0,1}$ and $\mathcal{M}'^{1,1}$

A.2 Comparison of models - Integrality requirement on variable $z_{m,i,n,j}$ relaxed

A.2.1 MIP Models $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$, $\mathcal{M}'^{1,0}$, $\mathcal{M}'^{1,1}$, $\mathcal{M}'^{2,0}$ and $\mathcal{M}'^{2,1}$class of instances and general average

Instances	$\mathcal{M}^{0',0}$			$\mathcal{M}^{0',1}$			$\mathcal{M}^{1',0}$			$\mathcal{M}^{1',1}$			$\mathcal{M}^{2',0}$			$\mathcal{M}^{2',1}$		
	Solution	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**
8x4S30	5063	9.380	0.000	5063	5.390	0.000	5063	1.422	0.000	5063	6.718	0.000	5063	0.188	0.000	5063	1.516	0.000
8x4S20	5086	7.094	0.000	5086	3.610	0.000	5086	0.922	0.000	5086	5.704	0.000	5086	0.156	0.000	5086	1.125	0.000
8x4S15	5112	1.310	0.000	5112	2.540	0.000	5112	1.109	0.000	5112	8.671	0.000	5112	0.234	0.000	5112	2.063	0.000
8x4S10	5169	1.440	0.000	5169	2.047	0.000	5169	1.157	0.000	5169	10.469	0.000	5169	0.219	0.000	5169	3.390	0.000
8x4S5	5174	1.200	0.000	5174	1.016	0.000	5174	1.140	0.000	5174	14.297	0.000	5174	0.281	0.000	5174	2.563	0.000
8x4	**5120.8**	**4.085**	**0.000**	**5120.8**	**2.921**	**0.000**	**5120.8**	**1.150**	**0.000**	**5120.8**	**9.172**	**0.000**	**5120.8**	**0.216**	**0.000**	**5120.8**	**2.131**	**0.000**
9x4S30	5904	60.650	0.000	5904	12.312	0.000	5904	1.282	0.000	5904	10.828	0.000	5904	0.230	0.000	5904	2.547	0.000
9x4S20	5937	14.310	0.000	5937	8.078	0.000	5937	1.406	0.000	5937	14.375	0.000	5937	0.120	0.000	5937	2.500	0.000
9x4S15	5976	3.890	0.000	5976	3.454	0.000	5976	1.300	0.000	5976	15.219	0.000	5976	0.140	0.000	5976	3.484	0.000
9x4S10	6027	3.760	0.000	6027	3.218	0.000	6027	1.359	0.000	6027	18.016	0.000	6027	0.300	0.000	6027	3.547	0.000

9x4S5	6047	1.920	0.000	6047	2.031	0.000	6047	1.875	0.000	6047	19.812	0.000	6047	0.230	0.000	6047	9.485	0.000
9x4	**5978.2**	**16.906**	**0.000**	**5978.2**	**5.819**	**0.000**	**5978.2**	**1.444**	**0.000**	**5978.2**	**15.650**	**0.000**	**5978.2**	**0.204**	**0.000**	**5978.2**	**4.313**	**0.000**
10x4S30	6193	630.450	0.000	6193	35.562	0.000	6193	2.328	0.000	6193	28.125	0.000	6193	0.300	0.000	6193	3.434	0.000
10x4S20	6267	486.640	0.000	6267	43.078	0.000	6267	2.703	0.000	6267	36.828	0.000	6267	0.310	0.000	6267	4.734	0.000
10x4S15	6296	165.090	0.000	6296	37.188	0.000	6296	2.453	0.000	6296	28.219	0.000	6296	0.250	0.000	6296	4.235	0.000
10x4S10	6325	49.250	0.000	6325	22.156	0.000	6325	2.610	0.000	6325	65.469	0.000	6325	0.330	0.000	6325	5.578	0.000
10x4S5	6518	5.360	0.000	6518	3.422	0.000	6518	3.703	0.000	6518	130.781	0.000	6518	0.530	0.000	6518	12.563	0.000
10x4	**6319.8**	**267.358**	**0.000**	**6319.8**	**28.281**	**0.000**	**6319.8**	**2.759**	**0.000**	**6319.8**	**57.884**	**0.000**	**6319.8**	**0.344**	**0.000**	**6319.8**	**6.109**	**0.000**
10x5S30	6308	7200	0.108	6308	585.235	0.000	6308	8.125	0.000	6308	164.312	0.000	6308	0.500	0.000	6308	21.234	0.000
10x5S20	6342	7200	0.057	6342	501.547	0.000	6342	7.781	0.000	6342	127.420	0.000	6342	0.940	0.000	6342	30.579	0.000
10x5S15	6397	6265.420	0.000	6397	162.531	0.000	6397	8.406	0.000	6397	289.969	0.000	6397	0.780	0.000	6397	47.547	0.000
10x5S10	6476	2341.390	0.000	6476	200.765	0.000	6476	8.250	0.000	6476	632.828	0.000	6476	0.780	0.000	6476	69.812	0.000
10x5S5	6616	70.670	0.000	6616	36.485	0.000	6616	11.063	0.000	6616	2236.328	0.000	6616	0.840	0.000	6616	88.797	0.000
10x5	**6427.8**	**4615.496**	**0.033**	**6427.8**	**297.313**	**0.000**	**6427.8**	**8.725**	**0.000**	**6427.8**	**690.171**	**0.000**	**6427.8**	**0.768**	**0.000**	**6427.8**	**51.594**	**0.000**
11x5S30	7443	7200	0.281	7420	4581.484	0.000	7420	10.220	0.000	7420	436.484	0.000	7420	0.970	0.000	7420	59.078	0.000
11x5S20	7475	7200	0.205	7439	2218.235	0.000	7439	8.047	0.000	7439	279.266	0.000	7439	0.860	0.000	7439	30.797	0.000
11x5S15	7543	7200	0.196	7535	699.515	0.000	7535	14.233	0.000	7535	585.766	0.000	7535	1.640	0.000	7535	62.984	0.000
11x5S10	7572	7200	0.028	7572	341.891	0.000	7572	17.375	0.000	7572	966.000	0.000	7572	1.660	0.000	7572	204.937	0.000
11x5S5	7812	1533.420	0.000	7812	158.970	0.000	7812	22.734	0.000	7812	7200,000	0.039	7812	1.880	0.000	7812	995.375	0.000
11x5	**7569.0**	**6066.684**	**0.142**	**7555.6**	**1600.019**	**0.000**	**7555.6**	**14.522**	**0.000**	**7555.6**	**1893.503**	**0.008**	**7555.6**	**1.402**	**0.000**	**7555.6**	**270.634**	**0.000**
12x5S30	8009	7200	0.408	7923	7200	0.205	7923	150.469	0.000	7923	4174.328	0.000	7923	3.220	0.000	7923	859.704	0.000
12x5S20	7991	7200	0.344	7939	7200	0.090	7939	48.609	0.000	7939	1154.922	0.000	7939	2.240	0.000	7939	108.110	0.000
12x5S15	7990	7200	0.212	7939	7200	0.123	7939	29.891	0.000	7939	626.469	0.000	7939	2.280	0.000	7939	78.328	0.000
12x5S10	8003	7200	0.226	7991	7200	0.126	7978	47.875	0.000	7978	1038.156	0.000	7978	2.730	0.000	7978	115.250	0.000
12x5S5	8072	6253.406	0.000	8072	390,000	0.000	8072	31.344	0.000	8110	7200	0.018	8072	3.300	0.000	8072	406.765	0.000
12x5	**8013.0**	**7010.681**	**0.238**	**7972.8**	**5838.016**	**0.109**	**7970.2**	**61.638**	**0.000**	**7977.8**	**2838.775**	**0.004**	**7970.2**	**2.754**	**0.000**	**7970.2**	**313.631**	**0.000**
12x6S30	10228	7200	0.416	10228	7200	0.315	10228	518.766	0.000	10303	7200	0.027	10228	26.950	0.000	10228	2350.593	0.000
12x6S20	10323	7200	0.390	10325	7200	0.150	10312	419.500	0.000	10377	7200	0.031	10312	9.450	0.000	10312	2745.188	0.000
12x6S15	10462	7200	0.267	10362	4331.140	0.000	10362	313.610	0.000	10388	7200	0.030	10362	7.050	0.000	10362	1220.735	0.000
12x6S10	10534	7200	0.286	10456	6334.672	0.000	10456	458.340	0.000	10579	7200	0.049	10456	12.050	0.000	10456	4438.219	0.000
12x6S5	10891	699.047	0.000	10891	532.125	0.000	10891	461.250	0.000	11091	7200	0.093	10891	4.980	0.000	10907	7200,000	0.030
12x6	**10487.6**	**5899.809**	**0.272**	**10452.4**	**5119.587**	**0.093**	**10449.8**	**434.293**	**0.000**	**10547.6**	**7200,000**	**0.046**	**10449.8**	**12.096**	**0.000**	**10453.0**	**3590.947**	**0.006**
15x6S30	13848	7200	0.670	13638	7200	0.500	13567	3701.510	0.000	13705	7200	0.037	13567	26.390	0.000	13656	7200,000	0.027

15x6S20	13802	7200	0.592	13750	7200	0.513	13720	7200	0.002	13871	7200	0.050	13720	68.500	0.000	13799	7200,000	0.033
15x6S15	14003	7200	0.745	13814	7200	0.441	13765	7200	0.005	13991	7200	0.054	13765	96.470	0.000	13956	7200,000	0.040
15x6S10	-	-	-	13803	7200	0.441	13803	6834,000	0.000	14006	7200	0.049	13803	24.390	0.000	13923	7200,000	0.329
15x6S5	-	-	-	14093	7200	0.604	13927	4457,000	0.000	14452	7200	0.083	13927	92.030	0.000	14085	7200,000	0.036
15x6	**-**	**-**	**-**	**13819.6**	**7200**	**0.500**	**13756.4**	**5878.424**	**0.001**	**14005.0**	**7200,000**	**0.054**	**13756.4**	**61.556**	**0.000**	**13883.8**	**7200,000**	**0.093**
15x7S30	-	-	-	14488	7200	0.480	14415	7200	0.022	14713	7200	0.071	14409	70.950	0.000	14491	7200,000	0.041
15x7S20	-	-	-	14656	7200	0.499	14533	7200	0.023	14973	7200	0.083	14514	62.940	0.000	14760	7200,000	0.041
15x7S15	-	-	-	14745	7200	0.572	14680	7200	0.030	14829	7200	0.074	14657	283.130	0.000	14915	7200,000	0.066
15x7S10	-	-	-	14814	7200	0.438	14824	7200	0.034	15746	7200	0.126	14810	208.470	0.000	14958	7200,000	0.058
15x7S5	-	-	-	15228	7200	0.630	15077	7200	0.031	-	-	-	15054	245.170	0.000	16184	7200,000	0.137
15x7	**-**	**-**	**-**	**14786.2**	**7200**	**0.524**	**14705.8**	**7200**	**0.028**	**-**	**-**	**-**	**14688.8**	**174.132**	**0.000**	**15061.6**	**7200,000**	**0.069**
20x10S30	-	-	-	29158	7200	0.917	29158	7200	0.153	33038	7200	0.221	28943	7200,000	0.091	34546	7200,000	0.254
20x10S20	-	-	-	29695	7200	0.870	29657	7200	0.168	34362	7200	0.247	29314	7200,000	0.094	35563	7200,000	0.270
20x10S15	-	-	-	29594	7200	0.905	29719	7200	0.171	-	-	-	29416	7200,000	0.095	34497	7200,000	0.245
20x10S10	-	-	-	30319	7200	0.926	29827	7200	0.172	-	-	-	29776	7200,000	0.110	-	-	-
20x10S5	-	-	-	30581	7200	0.892	31159	7200	0.205	-	-	-	30563	7200,000	0.114	34211	7200,000	0.231
20x10	**-**	**-**	**-**	**29869.4**	**7200**	**0.902**	**29904.0**	**7200**	**0.174**	**-**	**-**	**-**	**29602.4**	**7200,000**	**0.101**	**-**	**-**	**-**
Average	**-**	**-**	**-**	**10830,26**	**3449.196**	**0.213**	**10818.84**	**2080.296**	**0.020**	**-**	**-**	**-**	**10786.98**	**745.347**	**0.010**	**-**	**-**	**-**

TABLE A.3 – MIP Models $\mathcal{M}'^{0,0}$, $\mathcal{M}'^{0,1}$, $\mathcal{M}'^{1,0}$, $\mathcal{M}'^{1,1}$, $\mathcal{M}'^{2,0}$ and $\mathcal{M}'^{2,1}$

A.2.2 MIPs Models $\mathcal{M}^{3',0}, \mathcal{M}^{3',1}, \mathcal{M}'^{0',0}, \mathcal{M}'^{0',1}$ and $\mathcal{M}'^{1',1}$ for each class of instances and general average

Instances	$\mathcal{M}^{3',0}$			$\mathcal{M}^{3',1}$			$\mathcal{M}'^{0',0}$			$\mathcal{M}'^{0',1}$			$\mathcal{M}'^{1',1}$		
	Solution	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**	**Solution**	**Time**	**Gap**
8x4S30	5063	0.250	0.000	5063	2.297	0.000	5063	0.688	0.000	5063	2.843	0.000	5063	2.266	0.000
8x4S20	5086	0.187	0.000	5086	2.390	0.000	5086	0.218	0.000	5086	2.359	0.000	5086	2.047	0.000
8x4S15	5112	0.219	0.000	5112	3.375	0.000	5112	0.157	0.000	5112	1.391	0.000	5112	2.828	0.000
8x4S10	5169	0.250	0.000	5169	5.454	0.000	5169	0.156	0.000	5169	1.297	0.000	5169	4.188	0.000
8x4S5	5174	0.250	0.000	5174	4.750	0.000	5174	0.110	0.000	5174	0.563	0.000	5174	4.078	0.000
8x4	**5120.8**	**0.231**	**0.000**	**5120.8**	**3.653**	**0.000**	**5120.8**	**0.266**	**0.000**	**5120.8**	**1.691**	**0.000**	**5120.8**	**3.081**	**0.000**
9x4S30	5904	0.422	0.000	5904	3.687	0.000	5904	0.875	0.000	5904	7.094	0.000	5904	3.547	0.000

9x4S20	5937	0.375	0.000	5937	3.328	0.000	5937	0.344	0.000	5937	3.859	0.000	5937	3.609	0.000
9x4S15	5976	0.282	0.000	5976	4.516	0.000	5976	0.297	0.000	5976	3.891	0.000	5976	4.641	0.000
9x4S10	6027	0.391	0.000	6027	10.500	0.000	6027	0.265	0.000	6027	1.671	0.000	6027	5.281	0.000
9x4S5	6047	0.422	0.000	6047	8.672	0.000	6047	0.219	0.000	6047	1.282	0.000	6047	6.437	0.000
9x4	**5978.2**	**0.378**	**0.000**	**5978.2**	**6.141**	**0.000**	**5978.2**	**0.400**	**0.000**	**5978.2**	**3.559**	**0.000**	**5978.2**	**4.703**	**0.000**
10x4S30	6193	0.469	0.000	6193	2.844	0.000	6193	3.969	0.000	6193	27.078	0.000	6193	4.703	0.000
10x4S20	6267	0.531	0.000	6267	4.720	0.000	6267	2.125	0.000	6267	21.937	0.000	6267	8.423	0.000
10x4S15	6296	0.422	0.000	6296	7.390	0.000	6296	1.797	0.000	6296	15.390	0.000	6296	7.282	0.000
10x4S10	6325	0.453	0.000	6325	9.547	0.000	6325	1.421	0.000	6325	9.797	0.000	6325	11.203	0.000
10x4S5	6518	0.781	0.000	6518	23.375	0.000	6518	0.344	0.000	6518	3.688	0.000	6518	28.343	0.000
10x4	**6319.8**	**0.531**	**0.000**	**6319.8**	**9.575**	**0.000**	**6319.8**	**1.931**	**0.000**	**6319.8**	**15.578**	**0.000**	**6319.8**	**11.991**	**0.000**
10x5S30	6308	0.843	0.000	6308	44.578	0.000	6308	22.125	0.000	6308	299.891	0.000	6308	23.438	0.000
10x5S20	6342	0.937	0.000	6342	39.250	0.000	6342	26.063	0.000	6342	64.266	0.000	6342	40.156	0.000
10x5S15	6397	1.094	0.000	6397	54.500	0.000	6397	17.515	0.000	6397	91.000	0.000	6397	66.703	0.000
10x5S10	6476	1.406	0.000	6476	104.344	0.000	6476	6.985	0.000	6476	93.093	0.000	6476	105.453	0.000
10x5S5	6616	1.625	0.000	6616	348.344	0.000	6616	1.078	0.000	6616	11.265	0.000	6616	251.454	0.000
10x5	**6427.8**	**1.181**	**0.000**	**6427.8**	**118.203**	**0.000**	**6427.8**	**14.753**	**0.000**	**6427.8**	**111.903**	**0.000**	**6427.8**	**97.441**	**0.000**
11x5S30	7420	1.422	0.000	7420	75.922	0.000	7420	1055.984	0.000	7420	1683.297	0.000	7420	47.031	0.000
11x5S20	7439	1.344	0.000	7439	102.344	0.000	7439	281.672	0.000	7439	742.485	0.000	7439	42.547	0.000
11x5S15	7535	2.031	0.000	7535	199.906	0.000	7535	37.734	0.000	7535	354.220	0.000	7535	93.218	0.000
11x5S10	7572	1.922	0.000	7572	362.500	0.000	7572	25.610	0.000	7572	42.453	0.000	7572	261.047	0.000
11x5S5	7812	3.000	0.000	7812	1229.484	0.000	7812	7.312	0.000	7812	41.672	0.000	7812	2923.250	0.000
11x5	**7555.6**	**1.944**	**0.000**	**7555.6**	**394.031**	**0.000**	**7555.6**	**281.662**	**0.000**	**7555.6**	**572.825**	**0.000**	**7555.6**	**673.419**	**0.000**
12x5S30	7923	5.984	0.000	7923	597.656	0.000	7965	7200,000	0.139	7944	7200	0.127	7923	1137.219	0.000
12x5S20	7939	3.094	0.000	7939	779.641	0.000	7939	2186.516	0.000	7961	7200	0.127	7939	335.266	0.000
12x5S15	7939	2.547	0.000	7939	233.328	0.000	7939	3488.593	0.000	7939	7200	0.128	7939	151.875	0.000
12x5S10	7978	2.985	0.000	7978	114.125	0.000	-	-	-	7978	7200	0.152	7978	261.656	0.000
12x5S5	8072	3.266	0.000	8072	384.953	0.000	8072	1774.141	0.000	8072	156.062	0.000	8072	1863.422	0.000
12x5	**7970.2**	**3.575**	**0.000**	**7970.2**	**421.941**	**0.000**	**-**	**-**	**-**	**7978.8**	**5791.212**	**0.107**	**7970.2**	**749.888**	**0.000**
12x6S30	10228	8.922	0.000	10228	2756.407	0.000	-	-	-	10296	7200	0.112	10228	3984.922	0.000
12x6S20	10312	6.704	0.000	10312	3082.359	0.000	-	-	-	10369	7200	0.170	10312	3178.234	0.000
12x6S15	10362	8.328	0.000	10362	3290.953	0.000	-	-	-	10362	3208.406	0.000	10362	4246.547	0.000
12x6S10	10456	8.875	0.000	10456	5610.969	0.000	-	-	-	10456	4531.797	0.000	10456	5785.547	0.000

12x6S5	10891	35.922	0.000	10982	7200	0.051	10891	10,000	0.000	10891	1135.360	0.000	10904	7200	0.073
12x6	**10449.8**	**13.750**	**0.000**	**10468.0**	**4388.138**	**0.010**	**-**	**-**	**-**	**10474.8**	**4655.113**	**0.056**	**10452.4**	**4879.050**	**0.015**
15x6S30	13567	149.906	0.000	13679	7200	0.025	13776	7200,000	0.523	13567	7200	0.478	13687	7200	0.038
15x6S20	13720	112.750	0.000	13750	7200	0.022	13922	7200,000	0.468	13869	7200	0.469	13779	7200	0.037
15x6S15	13765	158.266	0.000	13805	7200	0.024	13920	7200,000	0.404	13856	7200	0.421	13793	7200	0.035
15x6S10	13803	112.562	0.000	13843	7200	0.020	-	-	-	13931	7200	0.414	13849	7200	0.035
15x6S5	13927	107.781	0.000	14258	7200	0.058	13927	7200,000	0.060	14024	7200	0.477	14105	7200	0.053
15x6	**13756.4**	**128.253**	**0.000**	**13867.0**	**7200**	**0.030**	**-**	**-**	**-**	**13849.4**	**7200**	**0.452**	**13842.6**	**7200**	**0.040**
15x7S30	14409	306.203	0.000	14634	7200	0.060	-	-	-	14579	7200	0.481	14491	7200	0.046
15x7S20	14514	259.125	0.000	14724	7200	0.051	-	-	-	14630	7200	0.496	14652	7200	0.051
15x7S15	14657	303.813	0.000	14862	7200	0.053	14848	7200,000	0.550	14682	7200	0.398	14711	7200	0.052
15x7S10	14810	313.516	0.000	15141	7200	0.076	-	-	-	14810	7200	0.408	14956	7200	0.065
15x7S5	15054	492.000	0.000	15708	7200	0.104	15128	7200,000	0.492	15108	7200	0.449	15370	7200	0.089
15x7	**14688.8**	**334.931**	**0.000**	**15013.8**	**7200**	**0.069**	**-**	**-**	**-**	**14761.8**	**7200**	**0.446**	**14836.0**	**7200**	**0.061**
20x10S30	29028	7200,000	0.103	34089	7200	0.244	29086	7200,000	0.826	28786	7200	0.861	31927	7200	0.194
20x10S20	29232	7200,000	0.096	34421	7200	0.246	29710	7200,000	0.821	29581	7200	0.881	33059	7200	0.217
20x10S15	29666	7200,000	0.103	-	-	-	29753	7200,000	0.814	29444	7200	0.864	31783	7200	0.183
20x10S10	29687	7200,000	0.089	-	-	-	29916	7200,000	0.805	29889	7200	0.870	34723	7200	0.250
20x10S5	30594	7200,000	0.114	-	-	-	30484	7200,000	0.828	30491	7200	0.890	34294	7200	0.237
20x10	**29641.4**	**7200,000**	**0.101**	**-**	**-**	**-**	**29789.8**	**7200,000**	**0.819**	**29638.2**	**7200**	**0.873**	**33157.2**	**7200**	**0.216**
Average	**10790.88**	**768.478**	**0.010**	**-**	**-**	**-**	-	**-**	**-**	**10810.52**	**3275.188**	**0.193**	**11166.060**	**2801.957**	**0.033**

TABLE A.4 – MIPs Models $\mathcal{M}^{3',0}, \mathcal{M}^{3',1}, \mathcal{M}'^{0',0}, \mathcal{M}'^{0',1}$ and $\mathcal{M}'^{1',1}$

A.3 Comparison of models - LP relaxation

The Table A.5 below reports the experiment results of LP relaxation of all the MIP models presented in chapter 4.

Instances	$\mathcal{M}^{0,0}$		$\mathcal{M}^{0,1}$		$\mathcal{M}^{1,0}$		$\mathcal{M}^{1,1}$		$\mathcal{M}^{2,0}$		$\mathcal{M}^{2,1}$		$\mathcal{M}^{3,0}$		$\mathcal{M}^{3,1}$		$\mathcal{M}'^{0,0}$		$\mathcal{M}'^{0,1}$		$\mathcal{M}'^{1,1}$	
	LB	**Time**	**LB**	**Time**	**LB**	**Time**	**LB**	**Time**	**LB**	**Time**	**LB**	**Time**	**LB**	**Time**	**LB**	**Time**	**LB**	**Time**	**LB**	**Time**	**LB**	**Time**
8x4S30	0	0.078	0	0.078	4824	0.078	4952.130	0.265	4824	0.047	4954.446	0.140	4824	0.062	4954.446	0.172	0	0.063	0	0.063	4952.130	0.125
8x4S20	0	0.047	0	0.047	4824	0.031	4970.630	0.516	4824	0.015	4974.596	0.141	4824	0.031	4974.596	0.156	0	0.016	0	0.015	4970.637	0.078

8x4S15	0	0.047	0	0.047	4824	0.031	4979.640	0.250	4824	0.016	4985.412	0.125	4824	0.016	4985.412	0.156	0	0.015	0	0.016	4979.640	0.094
8x4S10	0	0.047	0	0.031	4824	0.047	4989.929	0.219	4824	0.015	4997.373	0.156	4824	0.016	4997.373	0.157	0	0.016	0	0.015	4989.929	0.093
8x4S5	0	0.047	0	0.047	4824	0.047	4998.932	0.281	4824	0.000	5008.430	0.141	4824	0.015	5008.439	0.140	0	0.015	0	0.016	4998.932	0.109
8x4	**0**	**0.053**	**0**	**0.050**	**4824**	**0.047**	**4978.252**	**0.306**	**4824**	**0.019**	**4984.051**	**0.141**	**4824**	**0.028**	**4984.053**	**0.156**	**0**	**0.025**	**0**	**0.025**	**4978.254**	**0.100**
9x4S30	0	0.047	0	0.047	5584	0.047	5748.583	0.406	5584	0.016	5748.583	0.187	5584	0.016	5748.583	0.312	0	0.016	0	0.031	5748.583	0.110
9x4S20	0	0.047	0	0.047	5584	0.063	5771.989	0.360	5584	0.016	5774.249	0.313	5584	0.016	5774.249	0.328	0	0.031	0	0.016	5771.989	0.140
9x4S15	0	0.046	0	0.047	5584	0.046	5784.607	0.359	5584	0.016	5788.020	0.266	5584	0.016	5788.019	0.328	0	0.016	0	0.016	5784.607	0.141
9x4S10	0	0.032	0	0.047	5584	0.032	5798.000	0.766	5584	0.031	5807.181	0.203	5584	0.031	5807.181	0.266	0	0.016	0	0.015	5798.000	0.141
9x4S5	0	0.047	0	0.062	5584	0.047	5811.281	0.828	5584	0.031	5827.223	0.250	5584	0.016	5827.220	0.297	0	0.000	0	0.031	5811.280	0.171
9x4	**0**	**0.044**	**0**	**0.050**	**5584**	**0.047**	**5782.892**	**0.544**	**5584**	**0.022**	**5789.051**	**0.244**	**5584**	**0.019**	**5789.050**	**0.306**	**0**	**0.016**	**0**	**0.022**	**5782.892**	**0.141**
10x4S30	0	0.046	0	0.047	5960	0.078	6047.390	0.375	5960	0.031	6047.390	0.265	5960	0.046	6047.390	0.297	0	0.015	0	0.016	6047.390	0.172
10x4S20	0	0.063	0	0.031	5960	0.093	6066.899	0.438	5960	0.032	6066.899	0.407	5960	0.032	6066.899	0.578	0	0.016	0	0.016	6066.890	0.188
10x4S15	0	0.047	0	0.047	5960	0.079	6078.504	0.360	5960	0.031	6078.504	0.406	5960	0.031	6078.504	0.297	0	0.015	0	0.031	6078.504	0.156
10x4S10	0	0.047	0	0.047	5960	0.093	6092.304	0.390	5960	0.031	6092.951	0.312	5960	0.031	6092.950	0.281	0	0.016	0	0.016	6092.304	0.219
10x4S5	0	0.062	0	0.047	5960	0.078	6106.285	0.657	5960	0.031	6109.039	0.469	5960	0.031	6109.038	0.265	0	0.016	0	0.015	6106.285	0.219
10x4	**0**	**0.053**	**0**	**0.044**	**5960**	**0.084**	**6078.276**	**0.444**	**5960**	**0.031**	**6078.957**	**0.372**	**5960**	**0.034**	**6078.956**	**0.344**	**0**	**0.016**	**0**	**0.019**	**6078.275**	**0.191**
10x5S30	0	0.094	0	0.047	5968	0.125	6107.830	1.375	5968	0.047	6107.831	1.000	5968	0.032	6107.831	1.063	0	0.016	0	0.047	6107.830	0.812
10x5S20	0	0.094	0	0.062	5968	0.172	6129.560	1.250	5968	0.047	6132.316	2.672	5968	0.031	6132.315	2.000	0	0.016	0	0.047	6129.564	0.844
10x5S15	0	0.110	0	0.063	5968	0.157	6142.928	1.078	5968	0.047	6149.651	2.047	5968	0.031	6149.651	2.319	0	0.031	0	0.031	6142.928	0.969
10x5S10	0	0.078	0	0.078	5968	0.156	6156.277	1.234	5968	0.031	6166.459	1.750	5968	0.031	6166.459	2.000	0	0.015	0	0.047	6156.277	0.703
10x5S5	0	0.110	0	0.078	5968	0.156	6171.533	1.047	5968	0.047	6187.569	1.672	5968	0.031	6187.569	1.907	0	0.032	0	0.031	6171.530	1.015
10x5	**0**	**0.097**	**0**	**0.066**	**5968**	**0.153**	**6141.626**	**1.197**	**5968**	**0.044**	**6148.765**	**1.828**	**5968**	**0.031**	**6148.765**	**1.858**	**0**	**0.022**	**0**	**0.041**	**6141.626**	**0.869**
11x5S30	0	0.093	0	0.078	6968	0.266	7129.435	1.484	6968	0.063	7140.886	0.235	6968	0.047	7140.886	0.218	0	0.015	0	0.047	7129.435	1.282
11x5S20	0	0.077	0	0.078	6968	0.281	7148.967	1.563	6968	0.046	7170.543	0.234	6968	0.063	7170.543	0.266	0	0.047	0	0.047	7148.967	1.312
11x5S15	0	0.094	0	0.063	6968	0.218	7160.937	1.703	6968	0.063	7189.042	0.250	6968	0.062	7189.042	0.281	0	0.047	0	0.031	7160.937	1.313
11x5S10	0	0.110	0	0.078	6968	0.250	7175.424	1.656	6968	0.078	7211.014	0.266	6968	0.047	7211.014	0.266	0	0.031	0	0.047	7174.927	1.218
11x5S5	0	0.140	0	0.078	6968	0.328	7200.141	1.828	6968	0.047	7239.324	0.250	6968	0.063	7239.323	0.297	0	0.032	0	0.031	7191.457	1.078
11x5	**0**	**0.103**	**0**	**0.075**	**6968**	**0.269**	**7162.981**	**1.647**	**6968**	**0.059**	**7190.162**	**0.247**	**6968**	**0.056**	**7190.162**	**0.266**	**0**	**0.034**	**0**	**0.041**	**7161.145**	**1.241**
12x5S30	0	0.078	0	0.094	7408	0.360	7557.605	1.563	7408	0.094	7557.605	0.250	7408	0.046	7557.605	0.265	0	0.031	0	0.079	7557.606	1.000
12x5S20	0	0.078	0	0.078	7408	0.375	7579.290	1.406	7408	0.078	7585.150	0.328	7408	0.063	7585.150	0.313	0	0.047	0	0.046	7579.290	1.079
12x5S15	0	0.110	0	0.078	7408	0.312	7591.732	1.500	7408	0.078	7604.952	0.344	7408	0.062	7604.950	0.375	0	0.031	0	0.063	7591.732	1.000
12x5S10	0	0.125	0	0.094	7408	0.297	7603.407	1.469	7408	0.078	7622.944	0.312	7408	0.094	7622.944	0.328	0	0.031	0	0.047	7603.407	1.015
12x5S5	0	0.140	0	0.109	7408	0.344	7619.232	1.797	7408	0.078	7645.856	0.359	7408	0.078	7645.856	0.359	0	0.031	0	0.047	7619.232	1.078

12x5	**0**	**0.106**	**0**	**0.091**	**7408**	**0.338**	**7590.253**	**1.547**	**7408**	**0.081**	**7603.301**	**0.319**	**7408**	**0.069**	**7603.301**	**0.328**	**0**	**0.034**	**0**	**0.056**	**7590.253**	**1.034**
12x6S30	0	0.204	0	0.156	9472	0.687	9735.083	4.000	9472	0.125	9735.563	0.532	9472	0.125	9735.563	0.485	0	0.047	0	0.078	9735.083	2.157
12x6S20	0	0.156	0	0.110	9472	0.641	9775.843	3.984	9472	0.125	9777.034	0.562	9472	0.141	9777.034	0.562	0	0.063	0	0.062	9775.843	2.859
12x6S15	0	0.156	0	0.110	9472	0.781	9797.628	4.031	9472	0.110	9800.352	0.578	9472	0.110	9800.352	0.578	0	0.047	0	0.078	9797.628	2.859
12x6S10	0	0.125	0	0.125	9472	0.594	9817.316	4.297	9472	0.125	9822.745	0.516	9472	0.110	9822.740	0.579	0	0.047	0	0.094	9817.316	3.094
12x6S5	0	0.156	0	0.110	9472	0.718	9839.826	4.391	9472	0.125	9852.993	0.546	9472	0.125	9852.990	0.546	0	0.046	0	0.063	9839.826	2.828
12x6	**0**	**0.159**	**0**	**0.122**	**9472**	**0.684**	**9793.139**	**4.141**	**9472**	**0.122**	**9797.737**	**0.547**	**9472**	**0.122**	**9797.736**	**0.550**	**0**	**0.050**	**0**	**0.075**	**9793.139**	**2.759**
15x6S30	0	0.282	0	0.218	12544	1.563	12882.315	11.125	12544	0.218	12891.518	0.844	12544	0.203	12891.518	0.922	0	0.063	0	0.125	12882.315	5.438
15x6S20	0	0.250	0	0.188	12544	1.844	12932.098	16.906	12544	0.203	12950.441	0.906	12544	0.235	12950.441	1.000	0	0.078	0	0.140	12930.969	5.797
15x6S15	0	0.281	0	0.187	12544	1.657	12963.356	15.531	12544	0.204	12985.758	0.938	12544	0.203	12985.758	1.016	0	0.063	0	0.125	12959.013	7.187
15x6S10	0	0.281	0	0.203	12544	1.781	12998.252	13.813	12544	0.219	13025.579	1.031	12544	0.203	13025.579	1.078	0	0.062	0	0.141	12990.504	7.547
15x6S5	0	0.281	0	0.250	12544	1.594	13035.214	15.953	12544	0.187	13073.910	1.125	12544	0.203	13073.910	1.172	0	0.078	0	0.109	13027.347	8.875
15x6	**0**	**0.275**	**0**	**0.209**	**12544**	**1.688**	**12962.247**	**14.666**	**12544**	**0.206**	**12985.441**	**0.969**	**12544**	**0.209**	**12985.441**	**1.038**	**0**	**0.069**	**0**	**0.128**	**12958.030**	**6.969**
15x7S30	0	0.438	0	0.344	12992	0.891	13515.606	1.781	12992	0.266	13537.166	1.828	12992	0.313	13537.166	2.000	0	0.110	0	0.141	13514.354	12.797
15x7S20	0	0.406	0	0.313	12992	0.781	13578.144	2.422	12992	0.281	13616.176	2.016	12992	0.281	13616.175	2.156	0	0.110	0	0.141	13576.959	14.000
15x7S15	0	0.406	0	0.265	12992	0.797	13615.824	2.328	12992	0.266	13661.328	2.078	12992	0.328	13661.328	2.313	0	0.078	0	0.156	13613.546	17.875
15x7S10	0	0.407	0	0.266	12992	0.813	13651.894	2.969	12992	0.250	13708.061	2.141	12992	0.266	13708.061	2.547	0	0.094	0	0.140	13649.435	17.812
15x7S5	0	0.328	0	0.281	12992	0.796	13695.068	3.375	12992	0.266	13772.643	2.515	12992	0.297	13772.643	2.500	0	0.094	0	0.141	13689.502	17.891
15x7	**0**	**0.397**	**0**	**0.294**	**12992**	**0.816**	**13611.307**	**2.575**	**12992**	**0.266**	**13659.075**	**2.116**	**12992**	**0.297**	**13659.075**	**2.303**	**0**	**0.097**	**0**	**0.144**	**13608.759**	**16.075**
20x10S30	0	2.203	0	1.156	24552	2.922	25729.454	79.500	24552	1.312	25765.030	43.328	24552	1.562	25765.030	57.109	0	0.328	0	0.578	25725.915	54.922
20x10S20	0	1.515	0	1.141	24552	2.875	25878.510	99.781	24552	1.250	25931.489	62.032	24552	1.281	25931.489	52.047	0	0.359	0	0.703	25870.040	55.062
20x10S15	0	2.172	0	1.266	24552	2.469	25972.320	111.687	24552	1.219	26042.433	67.468	24552	1.375	26042.430	65.906	0	0.375	0	0.578	25961.326	46.437
20x10S10	0	3.735	0	2.734	24552	2.297	26063.120	104.532	24552	1.219	26160.667	70.844	24552	1.391	26160.660	76.484	0	0.359	0	0.579	26050.259	46.079
20x10S5	0	2.062	0	1.016	24552	2.391	26163.443	120.375	24552	1.203	26292.884	91.125	24552	1.406	26292.880	88.704	0	0.360	0	0.656	26142.669	44.562
20x10	**0**	**2.337**	**0**	**1.463**	**24552**	**2.591**	**25961.369**	**103.175**	**24552**	**1.241**	**26038.501**	**66.959**	**24552**	**1.403**	**26038.498**	**68.050**	**0**	**0.356**	**0**	**0.619**	**25950.042**	**49.412**
Average	**0**	**0.363**	**0**	**0.246**	**9627.2**	**0.672**	**10006.234**	**13.02408**	**9627.200**	**0.210**	**10027.504**	**7.375**	**9627.2**	**0.227**	**10027.504**	**7.520**	**0**	**0.072**	**0**	**0.117**	**10004.241**	**7.879**

TABLE A.5 – LP relaxation for all Models

A.4 Wilcoxon signed rank statistical tests for all of the models both for solution quality and runtime

In this part we provide results of Wilcoxon signed rank test applied to each pair of models regarding both solution quality and runtime. The corresponding p-values are provided in Tables A.6 and A.7. The p-value < 0.0001 means that there is significant difference between two models in the comparison, otherwise there is no significant difference. The models included in comparison are these that are able to provide a feasible solution for each test instance in the benchmark set.

	$\mathcal{M}^{1,0}$	$\mathcal{M}^{2,0}$	$\mathcal{M}^{3,0}$	$\mathcal{M}'^{1,1}$
$\mathcal{M}^{1,0}$	-	0.07	0.81	1.20e-03
$\mathcal{M}^{2,0}$	0.07	-	0.31	6.10e-05
$\mathcal{M}^{3,0}$	0.81	0.31	-	6.10e-05
$\mathcal{M}'^{1,1}$	1.20e-03	6.10e-05	6.10e-05	-

(a) p-value in terms of solution value

	$\mathcal{M}^{1,0}$	$\mathcal{M}^{2,0}$	$\mathcal{M}^{3,0}$	$\mathcal{M}'^{1,1}$
$\mathcal{M}^{1,0}$	-	5.48e-08	0.11	5.18e-09
$\mathcal{M}^{2,0}$	5.48e-08	-	5.18e-09	5.18e-09
$\mathcal{M}^{3,0}$	0.11	5.18e-09	-	5.18e-09
$\mathcal{M}'^{1,1}$	5.18e-09	5.18e-09	5.18e-09	-

(b) p-value in terms of CPU time

TABLE A.6 – p-values for all pairs of models in terms of solution value and CPU time - integrality requirement on variables $z_{m,i,n,j}$ imposed

	$\mathcal{M}^{0',1}$	$\mathcal{M}^{1',0}$	$\mathcal{M}^{2',0}$	$\mathcal{M}^{3',0}$	$\mathcal{M}'^{0',1}$	$\mathcal{M}'^{1',1}$
$\mathcal{M}^{0',1}$	-	0.06	4.37e-04	3.2e-03	0.28	0.01
$\mathcal{M}^{1',0}$	0.06	-	2.00e-03	2.00e-03	0.31	4.38e-04
$\mathcal{M}^{2',0}$	4.37e-04	2.00e-03	-	0.81	0.01	4.38e-04
$\mathcal{M}^{3',0}$	3.2e-03	2.00e-03	0.81	-	0.10	4.38e-04
$\mathcal{M}'^{0',1}$	0.25	0.31	0.01	0.10	-	0.10
$\mathcal{M}'^{1',1}$	0.01	4.38e-04	4.38e-04	4.38e-04	0.10	-

(a) p-value in terms of solution value

	$\mathcal{M}^{0',1}$	$\mathcal{M}^{1',0}$	$\mathcal{M}^{2',0}$	$\mathcal{M}^{3',0}$	$\mathcal{M}'^{0',1}$	$\mathcal{M}'^{1',1}$
$\mathcal{M}^{0',1}$	-	1.07e-07	5.18e-09	5.18e-09	4.17e-05	4.6e-03
$\mathcal{M}^{1',0}$	1.07e-07	-	5.18e-09	5.18e-09	2.34e-07	7.74e-08
$\mathcal{M}^{2',0}$	5.18e-09	5.18e-09	-	5.3e-06	5.18e-09	5.18e-09
$\mathcal{M}^{3',0}$	5.18e-09	5.18e-09	5.3e-06	-	5.18e-09	5.18e-09
$\mathcal{M}'^{0',1}$	4.17e-05	2.34e-07	5.18e-09	5.18e-09	-	0.3421
$\mathcal{M}'^{2',1}$	4.6e-03	7.74e-08	5.18e-09	5.18e-09	0.3421	-

(b) p-value in terms of CPU time

TABLE A.7 – p-values for all pairs of models in terms of solution value and CPU time - integrality requirement on variables $z_{m,i,n,j}$ relaxed

A.5 Detailed experiment results of Lagrangian Relaxation approach for the CDAP

The table A.8 provides, on the one hand, detailed solution value, gap and CPU time (in second) of the MILP Model $\mathcal{M}^{2,1}$. On the other hand, it gives lower bounds given by LP relaxation and Lagrangian Relaxation and associated deviation (in percentage) and CPU time (in second). Our lower bounds are compared to the lower bound given by Lagrangian dual of Nassief et al. (2016). The detailed results show that Lagrangian Relaxation improves significantly LP relaxation while still consuming important CPU time and that the lower bound of Nassief et al. (2016) is weak even compared with LP relaxation lower bound.

N x I	BKS	$\mathcal{M}^{2,1}$ CPLEX			$\mathcal{M}^{2,1}$ LP			$\mathcal{M}^{2,1}$ $LR\,(\lambda^0,\gamma^0)=(0,0)$			$\mathcal{M}^{2,1}$ $LR\,(\lambda^0,\gamma^0)=(d(4.9b),d(4.9c))$			LR Nassief	
		Value	Gap(%)	Time	LB	dev(%)	Time	LB	dev(%)	Time	LB	dev(%)	Time	LB	dev(%)
8x4S30	5063.00	5063.00	0.00	1.52	4954.45	2.14	0.14	4980.85	1.62	0.17	4982.94	1.58	0.17	4879.72	3.62
8x4S20	5086.00	5086.00	0.00	1.13	4974.60	2.19	0.141	4999.26	1.71	0.47	5011.69	1.46	0.20	4959.36	2.49

8x4S15	5112.00	5112.00	0.00	2.06	4985.41	2.48	0.125	5006.55	2.06	0.17	5022.61	1.75	0.17	4965.29	2.87
8x4S10	5169.00	5169.00	0.00	3.39	4997.37	3.32	0.156	5020.41	2.87	0.48	5037.99	2.53	0.14	4981.37	3.63
8x4S5	5174.00	5174.00	0.00	2.56	5008.43	3.20	0.141	5032.01	2.74	0.69	5047.99	2.44	0.22	4992.39	3.51
9x4S30	5904.00	5904.00	0.00	2.55	5748.58	2.63	0.187	5820.87	1.41	0.30	5826.53	1.31	0.31	5643.63	4.41
9x4S20	5937.00	5937.00	0.00	2.50	5774.25	2.74	0.313	5847.93	1.50	0.38	5853.43	1.41	0.28	5738.70	3.34
9x4S15	5976.00	5976.00	0.00	3.48	5788.02	3.15	0.266	5856.15	2.01	0.53	5874.10	1.71	0.23	5764.45	3.54
9x4S10	6027.00	6027.00	0.00	3.55	5807.18	3.65	0.203	5889.14	2.29	0.36	5893.13	2.22	0.31	5775.67	4.17
9x4S5	6047.00	6047.00	0.00	9.49	5827.22	3.63	0.25	5906.64	2.32	0.36	5909.98	2.27	0.31	5793.63	4.19
10x4S30	6193.00	6193.00	0.00	3.43	6047.39	2.35	0.265	6151.15	0.68	0.28	6154.81	0.62	0.28	5966.34	3.66
10x4S20	6267.00	6267.00	0.00	4.73	6066.90	3.19	0.407	6170.80	1.53	0.33	6178.29	1.42	0.39	5985.61	4.49
10x4S15	6296.00	6296.00	0.00	4.24	6078.50	3.45	0.406	6191.01	1.67	0.31	6197.10	1.57	0.42	6008.90	4.56
10x4S10	6325.00	6325.00	0.00	5.58	6092.95	3.67	0.312	6193.02	2.09	0.33	6207.74	1.85	0.30	6032.79	4.62
10x4S5	6518.00	6518.00	0.00	12.56	6109.04	6.27	0.469	6219.39	4.58	0.92	6226.63	4.47	0.33	6081.95	6.69
10x5S30	6308.00	6308.00	0.00	21.23	6107.83	3.17	1	6164.76	2.27	1.59	6187.46	1.91	0.55	6035.49	4.32
10x5S20	6342.00	6342.00	0.00	30.58	6132.32	3.31	2.672	6191.81	2.37	0.63	6207.31	2.12	0.61	6052.80	4.56
10x5S15	6397.00	6397.00	0.00	47.55	6149.65	3.87	2.047	6199.78	3.08	0.58	6221.92	2.74	0.59	6065.64	5.18
10x5S10	6476.00	6476.00	0.00	69.81	6166.46	4.78	1.75	6224.86	3.88	1.53	6237.75	3.68	0.73	6106.22	5.71
10x5S5	6616.00	6616.00	0.00	88.80	6187.57	6.48	1.672	6261.26	5.36	1.78	6271.67	5.20	0.78	6139.65	7.20
11x5S30	7420.00	7420.00	0.00	59.08	7140.89	3.76	0.235	7270.78	2.01	4.56	7291.43	1.73	1.74	7010.42	5.52
11x5S20	7439.00	7439.00	0.00	30.80	7170.54	3.61	0.234	7298.66	1.89	2.06	7327.84	1.49	1.58	7067.05	5.00
11x5S15	7535.00	7535.00	0.00	62.98	7189.04	4.59	0.25	7305.86	3.04	2.70	7353.53	2.41	1.22	7085.16	5.97
11x5S10	7572.00	7572.00	0.00	204.94	7211.01	4.77	0.266	7329.57	3.20	2.67	7376.23	2.59	1.50	7116.17	6.02
11x5S5	7812.00	7812.00	0.00	995.38	7239.32	7.33	0.25	7364.67	5.73	2.69	7389.59	5.41	2.86	7158.92	8.36
12x5S30	7923.00	7923.00	0.00	859.70	7557.61	4.61	0.25	7747.05	2.22	2.67	7771.41	1.91	3.56	7424.64	6.29
12x5S20	7939.00	7939.00	0.00	108.11	7585.15	4.46	0.328	7791.00	1.86	3.02	7810.62	1.62	4.47	7480.92	5.77
12x5S15	7939.00	7939.00	0.00	78.33	7604.95	4.21	0.344	7809.33	1.63	10.20	7835.87	1.30	4.81	7507.12	5.44
12x5S10	7978.00	7978.00	0.00	115.25	7622.94	4.45	0.312	7829.89	1.86	7.14	7854.41	1.55	4.31	7532.83	5.58

12x5S5	8072.00	8072.00	0.00	406.77	7645.86	5.28	0.359	7860.14	2.62	12.88	7881.75	2.36	5.81	7582.84	6.06
12x6S30	10228.00	10228.00	0.00	2350.59	9735.56	4.81	0.532	9962.15	2.60	123.63	10002.52	2.20	27.56	9562.16	6.51
12x6S20	10312.00	10312.00	0.00	2745.19	9777.03	5.19	0.562	10010.72	2.92	131.23	10045.43	2.59	14.64	9659.25	6.33
12x6S15	10362.00	10362.00	0.00	1220.74	9800.35	5.42	0.578	10052.92	2.98	36.78	10098.44	2.54	57.45	9702.98	6.36
12x6S10	10456.00	10456.00	0.00	4438.22	9822.75	6.06	0.516	10042.95	3.95	61.16	10096.46	3.44	43.80	9769.04	6.57
12x6S5	10891.00	10907.00	0.03	7200.00	9852.99	9.53	0.546	10102.79	7.24	116.41	10150.12	6.80	57.42	9784.47	10.16
15x6S30	13567.00	13656.00	0.027	7200.00	12891.52	4.98	0.844	13188.39	2.79	77.33	13257.99	2.28	108.88	12648.51	6.77
15x6S20	13720.00	13799.00	0.033	7200.00	12950.44	5.61	0.906	13249.50	3.43	195.16	13327.07	2.86	115.48	12733.53	7.19
15x6S15	13765.00	13956.00	0.04	7200.00	12985.76	5.66	0.938	13293.88	3.42	200.08	13365.96	2.90	126.00	12805.58	6.97
15x6S10	13803.00	13923.00	0.329	7200.00	13025.58	5.63	1.031	13285.84	3.75	171.31	13381.11	3.06	108.27	12883.72	6.66
15x6S5	13927.00	14085.00	0.036	7200.00	13073.91	6.13	1.125	13317.35	4.38	200.05	13374.15	3.97	129.25	12945.15	7.05
15x7S30	14409.00	14491.00	0.041	7200.00	13537.17	6.05	1.828	13715.48	4.81	394.06	13849.00	3.89	320.81	13270.69	7.90
15x7S20	14514.00	14760.00	0.041	7200.00	13616.18	6.19	2.016	13777.99	5.07	646.94	13904.29	4.20	244.17	13380.46	7.81
15x7S15	14657.00	14915.00	0.066	7200.00	13661.33	6.79	2.078	13846.16	5.53	746.77	13924.79	5.00	260.48	13465.39	8.13
15x7S10	14810.00	14958.00	0.058	7200.00	13708.06	7.44	2.141	13802.26	6.80	401.36	13968.66	5.68	142.16	13530.42	8.64
15x7S5	15054.00	16184.00	0.137	7200.00	13772.64	8.51	2.515	13842.01	8.05	597.75	13992.58	7.05	212.39	13553.12	9.97
20x10S30	28541.00	34546.00	0.254	7200.00	25765.03	9.73	43.328	25821.48	9.53	4098.61	25891.09	9.28	2653.09	25293.03	11.38
20x10S20	28963.00	35563.00	0.27	7200.00	25931.49	10.47	62.032	25749.27	11.10	4013.03	25841.37	10.78	4932.11	25539.57	11.82
20x10S15	29134.00	34497.00	0.245	7200.00	26042.43	10.61	67.468	25857.00	11.25	4954.76	25821.88	11.37	2627.17	25719.50	11.72
20x10S10	29286.00	-	-	-	26160.67	10.67	70.844	25908.78	11.53	2567.67	25919.38	11.50	454.47	25853.68	11.72
20x10S5	29933.00	34211.00	0.231	7200.00	26292.88	12.16	91.125	25858.42	13.61	5061.14	25889.07	13.51	5861.25	25921.98	13.40
Avg	**10743.88**	**-**	**-**	**-**	**10027.50**	**5.29**	**7.37**	**10132.40**	**3.98**	**497.16**	**10170.90**	**3.63**	**370.72**	9899.16	6.40

TABLE A.8 – Detailed comparison of lower bounds for $\mathcal{M}^{2,1}$ and Nassief et al.(2016)

Appendix B

Detailed results of computational experiments on PTS1 and PTS2 for the CDAP

In this annex, we provide detailed results on entire data sets "SetA" and on large-scale set of instances referred to "SetB" given by our PTS1 and PTS2 algorithms.

In Tables B.1 and B.2, we compare results found by our heuristic approaches with those reported in the literature by other methods. For PTS1 and PTS2 we report the best solution value (Column 'Best') and the average solution value found in 10 runs (Column 'Avg.') as well the average CPU time (Column 'CPU'). In Table B.1, together with the best known solutions (Column 'BKS'), we report the CPU time needed for CPLEX to solve the recent best MIP formulations for CDAP. The first one (Column '$\mathcal{M}^{3,0}$') is taken from Nassief et al. (2016) and the second one (Column '$\mathcal{M}^{2,0}$') is taken from Gelareh et al. (2020) (See chapiter 4 section 4.5). The results reported show that instances with up to 15 origins/destinations and 7 indoors/outdoors are optimally solved by CPLEX MIP solver within the maximum time limit of 492 seconds. On these instances, PTS algorithms were able to reach all optimal solutions in less than 8 seconds. Moreover, for instances with 20 origins/destinations and 10 indoors/outdoors, CPLEX did not reach optimal solutions in two hours while the PTS algorithms provide better results in less than 10 seconds. This comparison with CPLEX results, underscores the value of using Probabilistic Tabu Search for solving hard optimization problem such as CDAP. In Table B.2, the results are compared with those of the LS1 and LS2 heuristics proposed by Guignard et al. (2012), which are the only two methods executed on the "SetB" instances so far. For LS1 and LS2

we report the best solution value (Column 'Cost') and the normalized CPU time. Since LS1 and LS2 were executed on a machine with an AMD Phenom 9600 processor with 2.31gigahertz, a machine with different characteristics than our machine, we normalize their running times using the approach described in Dongarra (Dongarra (2014)) and data from http://www.cpubenchmark.net/. All comparisons were made according to the Passmark CPU Score (PCPUS). The running times were normalized by using our machine as the reference point, i.e., $\mathbf{Norm.Time(Algo)} = \frac{\mathbf{PCPUS\ (AMD\ Phenom\ 9600)} \times \mathbf{Time(Algo)}}{\mathbf{PCPUS(Intel\ Xeon\ E3-1505M)}}$, where Algo refers to LS1 & LS2. The Passmark CPU Scores of the processors AMD Phenom 9600 and Intel Xeon E3-1505M are 2303 and 8978, respectively. In each table, the boldfaced values correspond to the values that are equal or better than current BKS values, while underlined values denote the new BKS values established by our PTS. The values in italics correspond to the optimal solution values as reported in Gelareh et al. (2020).

Instances	**BKS**	$\mathcal{M}^{3,0}$	$\mathcal{M}^{2,0}$	**PTS1**			**PTS2**		
		CPU(s)	**CPU(s)**	**Best**	**Avg.**	**CPU(s)**	**Best**	**Avg.**	**CPU(s)**
8x4S5	*5174*	0.25	0.28	**5174**	**5174.0**	3.87	**5174**	**5174.0**	1.69
8x4S10	*5169*	0.25	0.22	**5169**	**5169.0**	3.68	**5169**	**5169.0**	1.12
8x4S15	*5112*	0.22	0.23	**5112**	**5112.0**	3.46	**5112**	**5112.0**	0.98
8x4S20	*5086*	0.1	0.16	**5086**	**5086.0**	3.29	**5086**	**5086.0**	0.87
8x4S30	*5063*	0.25	0.19	**5063**	**5063.0**	3.20	**5063**	**5063.0**	0.83
9x4S5	*6047*	0.42	0.23	**6047**	**6047.0**	4.50	**6047**	**6047.0**	1.25
9x4S10	*6027*	0.39	0.3	**6027**	**6027.0**	4.11	**6027**	**6027.0**	1.09
9x4S15	*5976*	0.28	0.14	**5976**	**5976.0**	3.93	**5976**	**5976.0**	0.92
9x4S20	*5937*	0.37	0.12	**5937**	**5937.0**	3.90	**5937**	**5937.0**	0.97
9x4S30	*5904*	0.42	0.23	**5904**	**5904.0**	3.74	**5904**	**5904.0**	0.90
10x4S5	*6518*	0.78	0.53	**6518**	**6518.0**	5.10	**6518**	**6518.0**	1.38
10x4S10	*6325*	0.45	0.33	**6325**	**6325.0**	4.87	**6325**	**6325.0**	1.09
10x4S15	*6296*	0.42	0.25	**6296**	**6296.0**	4.55	**6296**	**6296.0**	0.98
10x4S20	*6267*	0.53	0.31	**6267**	**6267.0**	4.40	**6267**	**6267.0**	0.97
10x4S30	*6193*	0.47	0.3	**6193**	**6193.0**	4.13	**6193**	**6193.0**	1.07
10x5S5	*6616*	1.62	0.84	**6616**	**6616.0**	4.54	**6616**	**6616.0**	1.80
10x5S10	*6476*	1.41	0.78	**6476**	**6476.0**	4.45	**6476**	**6476.0**	1.51
10x5S15	*6397*	1.09	0.78	**6397**	**6397.0**	4.14	**6397**	**6397.0**	1.37
10x5S20	*6342*	0.94	0.94	**6342**	**6342.0**	4.02	**6342**	**6342.0**	1.32
10x5S30	*6308*	0.84	0.5	**6308**	**6308.0**	3.86	**6308**	**6308.0**	1.31
11x5S5	*7812*	3	1.88	**7812**	**7812.0**	5.23	**7812**	**7812.0**	2.12
11x5S10	*7572*	1.92	1.66	**7572**	**7572.0**	4.95	**7572**	**7572.0**	1.61
11x5S15	*7535*	2.03	1.64	**7535**	**7535.0**	4.76	**7535**	**7535.0**	1.36
11x5S20	*7439*	1.34	0.86	**7439**	**7439.0**	4.47	**7439**	**7439.0**	1.29

11x5S30	*7420*	1.42	0.97	**7420**	**7420.0**	4.23	**7420**	7421.6	1.36
12x5S5	*8072*	3.27	3.3	**8072**	**8072.0**	5.81	**8072**	**8072.0**	1.76
12x5S10	*7978*	2.98	2.73	**7978**	**7978.0**	5.53	**7978**	**7978.0**	1.58
12x5S15	*7939*	2.55	2.28	**7939**	**7939.0**	5.36	**7939**	**7939.0**	1.48
12x5S20	*7939*	3.09	2.24	**7939**	**7939.0**	5.06	**7939**	**7939.0**	1.43
12x5S30	*7923*	5.98	3.22	**7923**	**7923.0**	4.68	**7923**	**7923.0**	1.40
12x6S5	*10891*	35.92	4.98	**10891**	**10891.0**	5.36	**10891**	**10891.0**	2.72
12x6S10	*10456*	8.87	12.05	**10456**	**10456.0**	5.13	**10456**	**10456.0**	1.98
12x6S15	*10362*	8.33	7.05	**10362**	**10362.0**	4.89	**10362**	10378.5	1.82
12x6S20	*10312*	6.70	9.45	**10312**	**10312.0**	4.61	**10312**	**10312.0**	1.92
12x6S30	*10228*	8.92	26.95	**10228**	**10228.0**	4.22	**10228**	**10228.0**	1.84
15x6S5	*13927*	107.78	92.03	**13927**	**13927.0**	7.32	**13927**	**13927.0**	2.47
15x6S10	*13803*	112.56	24.39	**13803**	**13803.0**	7.02	**13803**	13810.7	2.20
15x6S15	*13765*	158.27	96.47	**13765**	**13765.0**	6.54	**13765**	13792.3	2.23
15x6S20	*13720*	112.75	68.5	**13720**	**13720.0**	6.17	**13720**	13750.0	2.31
15x6S30	*13567*	149.90	26.39	**13567**	**13567.0**	5.68	**13567**	13585.2	2.22
15x7S5	*15054*	492	245.17	**15054**	**15054.0**	6.90	**15054**	15063.1	3.36
15x7S10	*14810*	313.52	208.47	**14810**	**14810.0**	6.49	**14810**	14843.1	2.70
15x7S15	*14657*	303.81	283.13	**14657**	14657.2	6.16	**14657**	14658.2	2.68
15x7S20	*14514*	259.12	62.94	**14514**	**14514.0**	5.80	**14514**	14537.6	2.72
15x7S30	*14409*	306.20	70.95	**14409**	**14409.0**	5.31	**14409**	14413.2	2.61
20x10S5	29933	7200.00	7200.00	**29907**	**29909.6**	9.16	**29907**	30004.4	6.39
20x10S10	29286	7200.00	7200.00	**29236**	**29253.3**	8.49	**29236**	29567.3	5.01
20x10S15	***29134***	7200.00	7200.00	29135	29135.5	7.77	29135	29345.7	4.81
20x10S20	28963	7200.00	7200.00	**28945**	**28951.2**	7.18	**28945**	29051.5	4.81
20x10S30	28541	7200.00	7200.00	**28533**	**28539.6**	6.85	**28533**	28741.3	4.82
AVG	**10743.88**	**768.48**	**745.35**	**10741.86**	**10742.53**	**5.18**	**10741.86**	**10764.39**	**2.01**

TABLE B.1 – Detailed results on "SetA" instances

Instance	LS1		LS2		PTS1			PTS2		
	Cost	CPU(s)	Cost	CPU(s)	Best	AVG	CPU(s)	Best	AVG	CPU(s)
25x10S5	49144	49.76	49335	50.79	**49013**	**49014.8**	26.11	**49013**	**49013.0**	13.81
25x10S10	48949	29.76	48941	33.09	**48672**	**48699.3**	23.49	**48740**	**48869.5**	12.50
25x10S15	48556	31.29	48504	33.60	**48407**	**48415.6**	20.08	**48407**	**48477.0**	12.10
25x10S20	48215	21.80	48235	23.09	**47934**	**47949.3**	19.11	**47926**	**47977.6**	12.24
25x10S30	47480	19.75	47426	20.78	**47314**	**47356.5**	19.61	**47314**	**47368.1**	12.25
25x20S30	51921	46.43	51741	51.30	**51533**	**51618.4**	19.81	**51562**	52334.2	28.51
50x10S5	191773	1036.07	191788	1153.30	**191160**	**191241.3**	78.29	**191186**	192350.7	26.58
50x10S10	189409	1371.08	189833	1362.87	**189166**	189478.5	66.16	189573	190316.3	26.19
50x10S15	188006	862.66	188264	800.33	**187315**	**187417.9**	69.16	**187377**	188834.9	26.45
50x10S20	186800	988.61	186578	956.55	**186085**	**186246.2**	72.28	**185975**	186788.2	26.24

50x10S30	183961	423.76	184013	401.96	**183249**	**183373.7**	79.04	**183234**	**183943.9**	25.85
50x20S5	238048	1814.59	239673	1874.62	**237656**	238244.5	59.28	239835	241379.3	50.66
50x20S10	235178	2020.58	234807	1985.69	**233341**	**234045.4**	44.56	235326	236932.9	51.59
50x20S15	230758	2004.67	230666	1921.30	**229638**	**230429.2**	45.58	**230375**	233110.0	48.04
50x20S20	227698	1836.91	227883	1879.75	**226448**	**227134.2**	45.65	228332	230201.4	50.93
50x20S30	221892	1831.01	222060	1783.04	**220748**	**221313.8**	48.48	**221322**	223423.5	45.85
50x30S15	275973	1532.68	275121	1782.79	**273166**	**274455.9**	50.52	276938	278798.1	91.23
50x30S20	264199	632.82	263790	783.91	**261725**	**263099.9**	43.55	264729	267802.5	86.29
50x30S30	254056	481.99	254795	503.28	**252639**	**253680.2**	45.23	255151	258277.8	77.02
50x43S30	332318	1789.71	330661	1893.34	**330285**	332378.3	47.46	335036	341233.4	121.07
75x10S5	440248	2337.89	440420	2423.05	**439055**	**439478.3**	158.11	**440181**	441088.8	44.82
75x10S10	435985	2268.11	435970	2354.30	**434275**	**435134.7**	157.91	**435595**	437051.9	44.88
75x10S15	431405	2162.17	431686	2185.77	**430005**	**430781.3**	171.16	**430663**	432183.0	45.28
75x10S20	427468	2250.16	427522	2084.70	**426385**	**427176.8**	179.44	**427168**	428920.9	44.74
75x10S30	420851	1732.51	420660	1704.04	**419651**	**420123.3**	198.28	**420619**	421566.1	44.48
75x20S5	531762	2096.76	532873	2243.49	**529131**	**529964.2**	86.09	532968	535724.6	72.92
75x20S10	523447	2377.65	521970	2479.23	**520127**	**521708.1**	86.09	524001	526829.3	72.91
75x20S15	514760	2508.21	514981	2466.14	**512170**	**512788.7**	87.11	515098	519040.7	73.93
75x20S20	506346	2210.40	506114	2512.06	**504502**	**505141.8**	90.33	506313	511407.4	72.29
75x20S30	493417	2449.47	493977	2527.96	**491796**	**492429.4**	97.83	**493403**	498179.1	73.09
75x30S10	636697	1769.45	634304	1792.02	**630259**	**631982.2**	78.17	637957	644569.3	113.86
75x30S15	620356	1919.00	618688	1921.56	**613001**	**614840.5**	81.81	621149	626365.0	112.66
75x30S20	600422	2230.92	601199	2151.91	**599329**	**601011.5**	84.85	605055	610181.4	114.44
75x30S30	580490	2369.18	582766	2349.43	**577843**	**579640.3**	88.66	583551	586215.3	103.99
100x10S5	773971	1429.05	773498	1369.54	**771172**	**771976.4**	319.05	**771368**	774128.9	68.89
100x10S10	764866	1288.74	763908	1384.93	**762282**	**763320.3**	311.69	**763469**	765036.1	69.55
100x10S15	757159	1496.00	757046	1368.00	**755040**	**755955.7**	348.72	**756236**	758154.4	68.04
100x10S20	750658	1321.06	750394	1323.62	**748611**	**749327.1**	367.73	**750047**	751747.3	68.52
100x10S30	738033	1122.26	737694	1253.34	**736248**	**737063.8**	415.95	**737505**	739087.7	68.69
100x20S5	966474	1262.06	970189	1154.32	**961900**	**964422.8**	142.32	968861	973939.2	104.34
100x20S10	951882	1147.65	949715	1267.19	**945835**	**948003.6**	145.08	952317	958369.8	102.84
100x20S15	935443	1294.64	936227	1284.38	**931525**	**933518.5**	151.57	**935246**	940053.2	102.65
100x20S20	921746	1266.68	922768	1334.65	**916505**	**918597.7**	157.15	**920851**	923581.8	102.52
100x20S30	894685	1424.95	896656	1367.74	**892755**	894825.9	171.95	896202	899657.8	104.15
100x30S5	1170457	722.86	1167044	700.03	**1154077**	**1159790.8**	121.82	**1164617**	1174570.3	141.40
100x30S10	1145700	1065.31	1142881	1104.81	**1127161**	**1130487.3**	126.65	**1140965**	1149830.1	145.87
100x30S15	1113552	1149.70	1119040	1184.08	**1103176**	**1105138.4**	130.40	1116441	1123978.2	146.01
100x30S20	1093126	1264.11	1096146	1232.82	**1081933**	**1086086.7**	133.10	1093174	1100436.0	143.75
100x30S30	1052682	1292.33	1057544	1271.81	**1044499**	**1046736.6**	140.48	1055745	1061462.5	138.23
AVG	**504253.51**	**1388.88**	**504448.86**	**1410.05**	**501137.59**	**502307.05**	**117.41**	**504369.71**	**507363.03**	**70.51**

TABLE B.2 – Detailed results on "SetB" instances

Bibliography

Adams, W. P., Guignard, M., Hahn, P. M. and Hightower, W. L. (2007). A level-2 reformulation linearization technique bound for the quadratic assignment problem, *European Journal of Operational Research* **180**(3) : 983–996.

Adams, W. P. and Sherali, H. D. (1990). Linearization strategies for a class of zero-one mixed integer programming problems, *Operations Research* **38**(9) : 217–226.

Agustina, D., Lee, C. K. M. and Piplani, R. (2010). A review : mathematical modles for cross docking planning, **2** : 13.

Alpan, G., Ladier, A.-L., Larbi, R. and Penz, B. (2011). Heuristic solutions for transshipment problems in a multiple door cross docking warehouse, *Computers & Industrial Engineering* **61**(2) : 402–408.

Bartholdi III, J. J. and Gue, K. R. (2000). Reducing labor costs in an ltl crossdocking terminal, *Operations Research* **48**(6) : 823–832.

Bartholdi, J. J. and Gue, K. R. (2004). The best shape for a crossdock, *Transportation Science* **38** : 235–244.

Bartholdi, J. J. and Hackman, S. T. (2011). Warehouse and Distribution Science, Release 0.94, *Technical report.*

Bellanger, A., Hanafi, S. and Wilbaut, C. (2013). Three-stage hybrid-flowshop model for cross-docking, *Computers and Operations Research* **40**(4) : 1109–1121.

Bermudez, R. and Cole, M. H. (2001). A genetic algorithm approach to door assignments in breakbulk terminals.

Birim, Ş. (2016). Vehicle routing problem with cross docking : A simulated annealing approach, *Procedia-Social and Behavioral Sciences* **235** : 149–158.

Bodnar, P., de Koster, R. and Azadeh, K. (2015). Scheduling Trucks in a Cross-Dock with Mixed Service Mode Dock Doors, *Transportation Science* **51**(1) : 112–131.

Boysen, N. and Fliedner, M. (2010). Cross dock scheduling : Classification, literature review and research agenda, *Omega* **38**(6) : 413–422.

Bozer, Y. and J. Carlo, H. (2008). Optimizing inbound and outbound door assignments in less-than-truckload crossdocks, **40** : 1007–1018.

Brown, A. M. (2003). *Improving the efficiency of hub operations in a less-than-truckload distribution network*, PhD Thesis, Virginia Tech.

Buijs, P., Vis, I. F. and Carlo, H. J. (2014). Synchronization in cross-docking networks : A research classification and framework, **239**(3) : 593–608.

Chu, P. C. and Beasley, J. E. (1997). A genetic algorithm for the generalised assignment problem, *Computers and Operations Research* **24**(1) : 17–23.

Cohen, Y. and Keren, B. (2008). A simple heuristic for assigning doors to trailers in cross-docks, *International Conference on Industrial Logistics. Tel Aviv, Israel*, pp. 1–14.

Cohen, Y. and Keren, B. (2009). Trailer to door assignment in a synchronous cross-dock operation, *International Journal of Logistics Systems and Management* **5**(5) : 574–590.

Cordeau, J.-F., Gaudioso, M., Laporte, G. and Moccia, L. (2006). A memetic heuristic for the generalized quadratic assignment problem, *INFORMS Journal on Computing* **18**(4) : 433–443.

Dıaz, J. A. and Fernández, E. (2001). A tabu search heuristic for the generalized assignment problem, *European Journal of Operational Research* **132**(1) : 22–38.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles, *Mathematical programming* **91**(2) : 201–213.

Dongarra, J. (2014). Performance of various computers using standard linear equations software, university of tennessee, *Computer Science Department* .

Fisher, M. L. (1981). The lagrangian relaxation method for solving integer programming problems, *Management science* **27**(1) : 1–18.

Fortet, R. (1960). Applications de l'algèbre de boole en recherche opérationelle, *Revue Française de Recherche Opérationelle* **4**(14) : 17–26.

French, A. P. and Wilson, J. M. (2007). An LP-based heuristic procedure for the generalized assignment problem with special ordered sets, *Computers and operations research* **34**(8) : 2359–2369.

Garey, M. R. and Johnson, D. S. (1983). Crossing number is np-complete, *SIAM Journal on Algebraic Discrete Methods* **4**(3) : 312–316.

Gelareh, S., Glover, F., Guemri, O., Hanafi, S., Nduwayo, P. and Todosijević, R. (2020). A comparative study of formulations for a cross-dock door assignment problem, *Omega* **91** : 102015.

Gelareh, S., Monemi, R. N., Semet, F. and Goncalves, G. (2016). A branch-and-cut algorithm for the truck dock assignment problem with operational time constraints, *European Journal of Operational Research* **249**(3) : 1144–1152.

Geoffrion, A. M. (1972). Lagrangean relaxation and its uses in integer programming, *Technical Report 195*, Defense Technical Information Center, Fort Belvoir, VA.

Geoffrion, A. M. (1974). *Lagrangean relaxation for integer programming*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 82–114.

Glover, F. (1986). Future paths for integer programming and links to artificial intelligence, *Computers and Operations Research* **13**(5) : 533–549.

Gue, K. and Kang, K. (2001). Staging queues in material handling and transportation systems, Vol. 2, IEEE, pp. 1104–1108.

Gue, K. R. (1999). The effects of trailer scheduling on the layout of freight terminals, *Transportation Science* **33**(4) : 419–428.

Guignard, M., Hahn, P. M., Pessoa, A. A. and Cardoso da Silva, D. (2012). Algorithms for the Cross-dock Door Assignment Problem, *in Proceedings of the XLI Brazilian Symposium on Operations Research*.

Habazin, J., Glasnović, A. and Bajor, I. (2017). Order Picking Process in Warehouse : Case Study of Dairy Industry in Croatia, *PROMET - Traffic and Transportation* **29**(1) : 57.

Hahn, P. M., Kim, B.-J., Guignard, M., Smith, J. M. and Zhu, Y.-R. (2008). An algorithm for the generalized quadratic assignment problem, *Computational Optimization and Applications* **40**(3) : 351–372.

Hahn, P. M., Kim, B.-J., Stuetzle, T., Kanthak, S., Hightower, W. L., Samra, H., Ding, Z. and Guignard, M. (2008). The quadratic three-dimensional assignment problem : Exact and approximate solution methods, *European Journal of Operational Research* **184**(2) : 416–428.

Held, M. and Karp, R. M. (1970). The traveling-salesman problem and minimum spanning trees, *Operations Research* **18**(6) : 1138–1162.

Held, M. and Karp, R. M. (1971). The traveling-salesman problem and minimum spanning trees : Part ii, *Mathematical programming* **1**(1) : 6–25.

Held, M., Wolfe, P. and Crowder, H. P. (1974). Validation of subgradient optimization, *Mathematical programming* **6**(1) : 62–88.

Hermel, D., Hasheminia, H., Adler, N. and Fry, M. J. (2016). A solution framework for the multi-mode resource-constrained cross-dock scheduling problem, *Omega* **59** : 157–170.

Jeet, V. and Kutanoglu, E. (2007). Lagrangian relaxation guided problem space search heuristics for generalized assignment problems, *European Journal of Operational Research* **182**(3) : 1039–1056.

Küçükoğlu, I. (2016). The effects of crossdock shapes on material handling costs, *International Journal of Computational Engineering Research* **06** : 1–05.

Küçükoğlu, I. and Öztürk, N. (2017). A mathematical model for truck-door assignment and product placement problem in cross-docking center, *Global Journal of Business, Economics and Management : Current Issues* **7**(1) : 135.

Ladier, A.-L. (2014). *Scheduling cross-docking operations : Integration of operational uncertainties and resource capacities*, PhD thesis, Université de Grenoble.

Ladier, A.-L. and Alpan, G. (2016a). Cross-docking operations : Current research versus industry practice, *Omega* **62** : 145–162.

Ladier, A.-L. and Alpan, G. (2016b). Robust cross-dock scheduling with time windows, *Computers and Industrial Engineering* **99** : 16–28.

Laguna, M., Kelly, J. P., González-Velarde, J. and Glover, F. (1995). Tabu search for the multilevel generalized assignment problem, *European journal of operational research* **82**(1) : 176–189.

Lim, A., Ma, H. and Miao, Z. (2006). Truck dock assignment problem with time windows and capacity constraint in transshipment network through crossdocks, *International Conference on Computational Science and Its Applications*, Springer, pp. 688–697.

Liu, Y. Y. and Wang, S. (2015). A scalable parallel genetic algorithm for the generalized assignment problem, *Parallel Computing* **46** : 98–119.

Lorena, L. A., Narciso, M. G. and Beasley, J. (2002). A constructive genetic algorithm for the generalized assignment problem, *Evolutionary Optimization* **5** : 1–19.

Mateus, G. R., Resende, M. G. and Silva, R. M. (2011). Grasp with path-relinking for the generalized quadratic assignment problem, *Journal of heuristics* **17**(5) : 527–565.

McKendall, A. and Li, C. (2017). A tabu search heuristic for a generalized quadratic assignment problem, *Journal of Industrial and Production Engineering* **34**(3) : 221–231.

Miao, Z., Cai, S. and Xu, D. (2014). Applying an adaptive tabu search algorithm to optimize truck-dock assignment in the crossdock management system, *Expert Systems with Applications* **41**(1) : 16–22.

Miao, Z., Lim, A. and Ma, H. (2009). Truck dock assignment problem with operational time constraint within crossdocks, *European Journal of Operational Research* **192**(1) : 105–115.

Miao, Z., Yang, F. and Fu, K. (2009). Transshipment problem with penalty in crossdocking distribution networks, *Service Systems and Service Management, 2009. ICSSSM'09. 6th International Conference on*, IEEE, pp. 280–285.

Nassief, W. (2017). *Cross-dock door assignments : models, algorithms and extensions*, PhD thesis, Concordia University.

Nassief, W., Contreras, I. and As' ad, R. (2016). A mixed-integer programming formulation and lagrangean relaxation for the cross-dock door assignment problem, *International Journal of Production Research* **54**(2) : 494–508.

Nassief, W., Contreras, I. and Jaumard, B. (2018). A comparison of formulations and relaxations for cross-dock door assignment problems, *Computers and Operations Research* **94** : 76–88.

Nguyen, T. T. H. (2017). Wal-mart's successfully integrated supply chain and the necessity of establishing the triple-a supply chain in the 21st century, *Journal of Economics & Management* **29** : 102–117.

Oh, Y., Hwang, H., Cha, C. N. and Lee, S. (2006). A dock-door assignment problem for the korean mail distribution center, *Computers and Industrial Engineering* **51**(2) : 288–296.

Pentico, D. W. (2007). Assignment problems : A golden anniversary survey, *European Journal of Operational Research* **176**(2) : 774–793.

Pessoa, A. A., Hahn, P. M., Guignard, M. and Zhu, Y.-R. (2010). Algorithms for the generalized quadratic assignment problem combining lagrangean decomposition and the reformulation-linearization technique, *European Journal of Operational Research* **206**(1) : 54–63.

Ramaa, A., Subramanya, K. and Rangaswamy, T. (2012). Impact of warehouse management system in a supply chain, *International Journal of Computer Applications* **54**(1).

Ross, G. T. and Soland, R. M. (1975). A branch and bound algorithm for the generalized assignment problem, *Mathematical programming* **8**(1) : 91–103.

Sahni, S. and Gonzalez, T. (1976). P-complete approximation problems, *Journal of the ACM (JACM)* **23**(3) : 555–565.

Schirotzek, W. (1986). Shor, nz, minimization methods for non-differentiable functions. berlin-heidelberg-new york-tokyo, springer-verlag 1985. viii, 162 s., dm 84,—. isbn 3-540-12763-1 (springer series in computational mathematics 3)—translation from the russian, *Zeitschrift Angewandte Mathematik und Mechanik* **66** : 575–575.

Shakeri, M., Low, M. Y. H. and Li, Z. (2008). A generic model for crossdock truck scheduling and truck-to-door assignment problems, *Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on*, IEEE, pp. 857–864.

Shakeri, M., Low, M. Y. H., Turner, S. J. and Lee, E. W. (2012). A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock, *Computers and Operations Research* **39**(11) : 2564–2577.

Shor, N. (1968). The rate of convergence of the generalized gradient descent method, *Cybernetics and Systems Analysis* **4**(3) : 79–80.

Shuib, A. and Fatthi, W. N. A. W. A. (2012). A review on quantitative approaches for dock door assignment in cross-docking, **2**(5) : 370–374.

Stalk, G., Evans, P. and Shulman, L. E. (1992). Competing on Capabilities : The New Rules of Corporate Strategy, p. 13.

Stephan, K. and Boysen, N. (2011). Cross-docking, *Journal of Management Control* **22**(1) : 129–137.

Tarhini, A. A., Yunis, M. M. and Chamseddine, M. (2016). Natural Optimization Algorithms for the Cross-Dock Door Assignment Problem, *IEEE Transactions on Intelligent Transportation Systems* **17**(8) : 2324–2333.

Ting, C.-J. and Rodríguez López, A. G. (2012). Dock assignment and truck scheduling problems at cross-docking terminals.

Tsui, L. Y. and Chang, C.-H. (1990). A microcomputer based decision support tool for assigning dock doors in freight yards, *Computers and Industrial Engineering* **19**(1) : 309–312.

Tsui, L. Y. and Chang, C.-H. (1992). An optimal solution to a dock door assignment problem, *Computers and Industrial Engineering* **23**(1) : 283–286.

Van Belle, J., Valckenaers, P. and Cattrysse, D. (2012). Cross-docking : State of the art, **40**(6) : 827–846.

Wen, M., Larsen, J., Clausen, J., Cordeau, J.-F. and Laporte, G. (2009). Vehicle routing with cross-docking, *Journal of the Operational Research Society* **60**(12) : 1708–1718.

Wilcoxon, F. (1945). Individual comparisons by ranking methods, *Biometrics bulletin* **1**(6) : 80–83.

Wilson, J. (1997). A genetic algorithm for the generalised assignment problem, *Journal of the Operational Research Society* **48**(8) : 804–809.

Woodcock, A. J. and Wilson, J. M. (2010). A hybrid tabu search/branch & bound approach to solving the generalized assignment problem, *European journal of operational research* **207**(2) : 566–578.

Yagiura, M., Ibaraki, T. and Glover, F. (2004). An ejection chain approach for the generalized assignment problem, *INFORMS Journal on Computing* **16**(2) : 133–151.

Yagiura, M., Ibaraki, T. and Glover, F. (2006). A path relinking approach with ejection chains for the generalized assignment problem, *European journal of operational research* **169**(2) : 548–569.

Yagiura, M., Iwasaki, S., Ibaraki, T. and Glover, F. (2004). A very large-scale neighborhood search algorithm for the multi-resource generalized assignment problem, *Discrete Optimization* **1**(1) : 87–98.

Yang, K. K., Balakrishnan, J. and Cheng, C. H. (2010). An analysis of factors affecting cross docking operations, **31**(1) : 121–148.

Yu, V. F., Sharma, D. and Murty, K. G. (2008). Door allocations to origins and destinations at less-than-truckload trucking terminals.

Zeinebou, Z. and Abdellatif, B. (2013). Development of a model of decision support for optimization of physical flows in a container terminal, *International Conference on Advanced Logistics and Transport*, IEEE, pp. 421–426.

Zhang, L. (2016). *Optimization and simulation of a cross-docking terminal*, PhD thesis, Automatic Control Engineering. Ecole Centrale de Lille.

Zhu, Y. R., Hahn, P. M., Liu, Y. and Guignard, M. (2009). New approach for the cross-dock door assignment problem, *in Proceedings of the XLI Brazilian Symposium on Operations Research*.