SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Pedro Belin Castellucci

Consolidation problems in freight transportation systems: mathematical models and algorithms

Thesis submitted to the Institute of Mathematics and Computer Sciences (ICMC-USP) and to the School of Mathematics and Statistics (The University of Melbourne) in accordance with the requirements of the Computer and Mathematical Sciences Graduate Program, for the degree of Doctor in Science (ICMC-USP) and for the degree of Doctor of Philosophy (The University of Melbourne). *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dra Franklina Maria Bragion de Toledo (USP)

Advisor: Prof. Dr. Alysson Machado Costa (Unimelb)

Co-advisor: Prof. Dr. Russell Thompson (Unimelb)

USP – São Carlos

September 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

C348c Castellucci, Pedro Belin
Consolidation problems in freight transportation
systems: mathematical models and algorithms / Pedro
Belin Castellucci; orientadora Franklina Toledo;
coorientador Alysson Costa. -- São Carlos, 2019.
144 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2019.

1. Freight transporation. 2. Optimisation. 3.
Container loading. 4. Cross-docking. 5.
Collaborative logistics. I. Toledo, Franklina ,
orient. II. Costa, Alysson, coorient. III. Título.

Pedro Belin Castellucci

**Problemas de consolidação em sistemas de transportes:
modelos matemáticos e algoritmos**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação (ICMC-USP) e à Escola de Matemática e Estatística (Universidade de Melbourne), como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional (ICMC-USP) e para o título de Doutor em Filosofia (Universidade de Melbourne). *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Prof. Dra Franklina Maria Bragion de Toledo (USP)

Orientador: Prof. Dr. Alysson Machado Costa (Unimelb)

Coorientador: Prof. Dr. Russell Thompson (Unimelb)

**USP – São Carlos
Setembro de 2019**

ACKNOWLEDGEMENTS

Although I am the only one getting a PhD, it was a journey of many. From the many (many, many, ..., many) hours of paperwork to define an agreement between the Universidade de São Paulo (USP) and the University of Melbourne (Unimelb) to the research itself. I thank the offices from USP and Unimelb for making this agreement possible. Specially, Carmella Prideaux for her effort in ensuring everything followed the right path.

In this journey, I was lucky to take courses with all (but my supervisor) current researchers from the *Laboratório de Otimização (LOt/USP)*: Eduardo Fontoura, Elias Helou and Marina Andretta and Maristela dos Santos. They each had an impact in forming my professional character and in my research as well. Definitely the one influencing me the most were my supervisors Franklina Toledo and Alysson Costa. I am sure they did not sign up for the amount of paperwork that goes with supervising an intercontinental PhD. Nevertheless, they were always available and cheerful to discuss my research and provided insightful feedback – even in the late nights due to worst possible time zone differences. I also thank Russell Thompson from Unimelb for co-supervising me, specially during my time in Melbourne. Without them, this PhD would never have reached its present state.

I thank all the students who were part of LOt/USP and the ones which I knew by the time I was in Melbourne, you have made this experience even more enjoyable.

I thank the members of the judging committee Prof. Dr. Regina Berreta, Prof. Dr. Flávio Miyazawa and Prof. Dr. André Carvalho for their valuable contributions to this thesis.

Finally, most deep gratitude to Lívea Coda, who was with me before this all started and I hope she will continue to be.

I also thank the financial support from the State of São Paulo Research Foundation (FAPESP), process numbers, 2017/01097-9, 2015/15024-8, CEPID 2013/07375-0.

ABSTRACT

CASTELLUCCI, P. B. **Consolidation problems in freight transportation systems: mathematical models and algorithms**. 2019. 144 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Freight distribution systems are under stress. With the world population growing, the migration of people to urban areas and technologies that allow purchases from virtually anywhere, efficient freight distribution can be challenging. An inefficient movement of goods may lead to business not being economically viable and also has social and environmental negative effects. An important strategy to be incorporated in freight distribution systems is the consolidation of goods, i.e., group goods by their destination. This strategy increases vehicles utilisation, reducing the number of vehicles and the number of trips required for the distribution and, consequently, costs, traffic, noise and air pollution. In this thesis, we explore consolidation in three different contexts (or cases) from an optimisation point of view. Each context is related to optimisation problems for which we developed mathematical programming models and solution methods.

The first case in which we explore consolidation is in container loading problems (CLPs). CLPs are a class of packing problems which aims at positioning three-dimensional boxes inside a container efficiently. The literature has incorporated many practical aspects into container loading solution method (e.g. restricting orientation of boxes, stability and weight distribution). However, to the best of our knowledge, the case considering more dynamic systems (e.g. cross-docking) in which goods might have a schedule of arrival were yet to be contemplated by the literature. We define an extension of CLP which we call Container Loading Problem with Time Availability Constraints (CLPTAC), which considers boxes are not always available for loading. We propose an extension of a CLP model that is suitable for CLPTAC and solution methods which can also handle cases with uncertainty in the schedule of the arrival of the boxes.

The second case is a more broad view of the network, considering an open vehicle routing problem with cross-dock selection. The traditional vehicle routing problem has been fairly studied. Its open version (i.e. with routes that start and end at different points) has not received the same attention. We propose a version of the open vehicle routing problem in which some nodes of the network are consolidation centres. Instead of shippers sending goods directly to their consumers, they must send to one of the available consolidation centres, then, goods are resorted and forwarded to their destination. For this problem, we propose a mixed integer linear programming model for cost minimisation and a solution method based on the Benders decomposition framework.

A third case in which we explored consolidation is in collaborative logistics. Particularly, we focus on the shared use of the currently available infrastructure. We defined a hub

selection problem in which one of the suppliers is selected as a hub. In a hub facility, other suppliers might meet to exchange their goods allowing one supplier to satisfy the demand from others. For this problem, we propose a mixed integer linear programming model and a heuristic based on the model. Moreover, we compared a traditional distribution strategy, with each supplier handling its demand, against the collaborative one.

In this thesis, we explore these three cases which are related to consolidation for improving the efficiency in freight distribution systems. We extend some problems (e.g. versions of CLP) to apply them to a more dynamic setting and we also define optimisation problems for networks with consolidation centres. Furthermore, we propose solution methods for each of the defined problems and evaluate them using randomly generated instances, benchmarks from the literature and some cases based on real-world characteristics.

Keywords: Freight transportation, Optimisation, Container loading, Cross-docking, Collaborative logistics.

RESUMO

CASTELLUCCI, P. B. **Problemas de consolidação em sistemas de transportes: modelos matemáticos e algoritmos**. 2019. 144 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Sistemas de distribuição de carga possuem uma demanda muito alta. Com a população mundial crescendo, a migração em direção às áreas urbanas e as tecnologias que permitem compras de virtualmente qualquer lugar, a distribuição eficiente de mercadorias pode ser um desafio. Uma movimentação ineficiente de mercadorias pode tornar negócios economicamente inviáveis além de ter um impacto social e ambiental negativos. Uma estratégia importante para se incorporar em sistemas de distribuição é a consolidação de cargas, isto é, agrupar cargas de acordo com seus destinos. Essa estratégia aumenta a utilização dos veículos, reduzindo o número de veículos e viagens necessários para a distribuição e, conseqüentemente, custos, tráfego, poluição sonora e do ar. Nesta tese, é explorada a técnica de consolidação em três casos diferentes de um ponto de vista de otimização. Cada caso é relacionado a problemas de otimização para os quais são propostos modelos de programação matemática e métodos de solução.

O primeiro caso em que é explorada a consolidação é em Problemas de Carregamento de Contêineres (PCCs). PCCs pertencem a uma classe de problemas de empacotamento que visa posicionar caixas tridimensionais dentro de contêineres eficientemente. A literatura tem incorporado diversos aspectos práticos em procedimentos de solução dos PCCs (por exemplo, restringir a orientação das caixas, estabilidade e distribuição de peso). No entanto, o caso que considera sistemas logísticos mais dinâmicos (como *cross-docking*), nos quais mercadorias podem ter uma agenda de chegada ainda não havia sido contemplados. É definida uma extensão de PCC chamada de Problema de Carregamento de Contêineres com Restrições de Disponibilidade Temporal (PCCRDT). Também, propõem-se modelos e métodos de solução para o PCCRDT que são capazes de lidar com incerteza na chegada das mercadorias.

O segundo caso utiliza uma visão mais abrangente da rede de distribuição, considerando um problema de roteamento de veículos em rede aberta com seleção de *cross-dock*. O problema tradicional de roteamento de veículos é bastante estudado. A sua versão aberta (com rotas que começam e terminam em pontos diferentes) não tem recebido tanta atenção. É proposta uma versão do roteamento de veículos em rede aberta em que alguns nós da rede são centros de consolidação. Os fornecedores, ao invés de enviar as mercadorias diretamente para os consumidores, enviam-nas para um dos centros de consolidação disponíveis, então, as mercadorias são reorganizadas (em diferentes veículos) e encaminhadas para os seus destinos. Para esse problema, é proposto um modelo de programação linear inteira

mista para a minimização de custo e um método de solução baseado no arcabouço de decomposição de Benders.

Um terceiro caso em que foi explorada a consolidação de mercadorias é o de logística colaborativa. Particularmente, se concentrou no uso compartilhado de infra-estrutura já disponível na rede de distribuição. É definido um problema de seleção de seleção de um dos fornecedores como *hub*. No *hub*, outros fornecedores podem se encontrar para trocar suas mercadorias, permitindo que um fornecedor satisfaça a demanda de outro. Para esse problema, é proposto um modelo de programação linear inteira mista e uma heurística baseada no modelo. Ainda, é comparada uma estratégia de distribuição convencional (com cada fornecedor responsável pela sua própria demanda) com uma estratégia colaborativa.

Nesta tese, são explorados esses três casos que se relacionam com consolidação para melhorar a eficiência de sistemas de distribuição de carga. São estendidos alguns problemas (como o PCC) para que se possa aplicá-los em cenários mais dinâmicos e também são definidos problemas de otimização em redes com centros de consolidação. Além disso, são propostos métodos de solução para cada um dos casos. Os métodos são avaliados em instâncias geradas aleatoriamente, instâncias da literatura e, em alguns casos, instâncias baseadas em cenários reais.

Palavras-chave: Transporte de carga, Otimização, Carregamento de contêineres, Cross-docking, Logística colaborativa.

I declare that:

This thesis comprises only my original work towards the PhD except where indicated. Due acknowledgement has been made in the text to all other material used. The thesis is fewer than 100,000 words in length, exclusive of tables, maps, bibliographies and appendices.

Pedro Belin Castellucci

LIST OF FIGURES

Figure 1 – Representation of one, two and three-dimensional packing problems. . .	23
Figure 2 – Schematic representation of a cross-dock.	24
Figure 3 – Scheme of a typical cross-dock. In the facility ,there are inbound docks to receive freights from the inbound trucks (suppliers S1, S2 and S3). These freights are moved from the input docks to the output docks. In the output docks, there is an outbound vehicle for each consumer or group of consumers (C1, C2, C3 and C4). This figure was adapted from Belle, Valckenaers and Cattrysse (2012).	29
Figure 4 – Scheme with typical cross-dock networks connecting suppliers and consumers. In a <i>few to many</i> scenario, few suppliers feed a larger number of consumers. The opposite happens in <i>many to few</i> network. In a <i>many to many</i> case there is an equivalent number of suppliers and consumers. It is also possible to classify the networks according to the number of cross-docks that may be visited by freight to go from a supplier to a consumer. In <i>single layer</i> networks only on cross-dock is visited whilst more than one may be visited in <i>hub and spoke</i> systems. This figure was adapted from Buijs, Vis and Carlo (2014).	30
Figure 5 – Summary of the typology for cutting-stock and packing problems (Wäscher, Haußner and Schumann, 2007).	40
Figure 6 – Classification scheme for container loading problems based on heterogeneity of cargo and boxes. This figure was inspired on (Wäscher, Haußner and Schumann, 2007).	41
Figure 7 – Schematic representation of centralised (a) and decentralised collaboration (b) systems.	58
Figure 8 – Value of the probability of $\mathcal{B}_1$ arriving at $i \in \{1, \dots, 9\}$ for a reliability index of $\alpha = 0.7$	67
Figure 9 – Example of a scenario with $\mathcal{T} = \{1, 2, 3\}$. It is a graphical example of equation (3.7). The dashed red lines show possible next states for $\mathcal{X}_1$. The full blue lines show the states which need to be evaluated at state 010 at time $t = 2$	68
Figure 10 – Profile of empty volume (dashed line) and operation time (full line) related to the reliability index (α) for $\mu \in \{1, 2, 4\}$. The values were averaged across all instances.	69

Figure 11 – Average of dispatching objective (η , circles) and number of trucks (γ , triangles) related to values of α^{vol} and α^{tr} . Values of α^{vol} are shown in the horizontal axis and $\alpha^{tr} \in \{0.1, 0.2, \dots, 1.7\}$ are coded in the transparency of the curves, the higher the transparency the higher the α^{tr} value. Note that some combinations lead to the same curves. The corresponding arrows indicate the increasing direction α^{tr}	76
Figure 12 – Comparison cases 0, 1, 2 and 3 for $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$ for the dispatching objective (η) different probabilities of arrival (β).	80
Figure 13 – Comparison cases 0, 1, 2 and 3 for $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$ for the number of trucks objective (γ) different probabilities of arrival (β).	80
Figure 14 – Scheme of a logistic network with cross-docks (consolidation centres). Suppliers $o_1, \dots, o_{ \mathcal{O} }$ must send their shipments (set of goods) to one of the consolidation centres ($f_1, \dots, f_{ \mathcal{F} }$). Then, goods are loaded into trucks and dispatched to their final destination ($d_1, \dots, d_{ \mathcal{D} }$).	87
Figure 15 – Scheme of the decomposition algorithm for the network scheduling problem. Solving a master problem determines the set of shipments $\mathcal{K}_{fd_1}, \dots, \mathcal{K}_{fd_{ \mathcal{D} }}$ sent to facility $f \in \mathcal{F} = \{f_1, \dots, f_{ \mathcal{F} }\}$ which is equivalent to defining the set of boxes $\mathcal{I}_{fd_1}, \dots, \mathcal{I}_{fd_{ \mathcal{D} }}$. Set $\mathcal{I}_{fd}$ is the set of boxes sent to facility $f \in \mathcal{F}$ for destination $d \in \mathcal{D} = \{d_1, \dots, d_{ \mathcal{D} }\}$	91
Figure 16 – Example of distribution without collaboration. Each large node (red square, blue star, green circle) represents a supplier, providing the associated type of good. Each cell represents a consumer which has demand from the type of items, indicated by the drawing inside the cell. In this non-collaborative case, each supplier is responsible for routing its own deliveries.	104
Figure 17 – Example with collaboration with supplier hub being the square node. Each large node (red square, blue star, green circle) represents a supplier. Each cell represents a consumer which has demand from the type of items indicated by the drawing inside the cell. In this case, after some exchanges happen with the hub (indicated by the dotted black two-way arrows), a supplier can satisfy demands for different sets of consumers.	105
Figure 18 – Graphical representation of the relax-and-fix strategy. At each iteration $1, \dots, T $, only the correspondent subset of variables are considered binary. Others are either fixed or relaxed.	108
Figure 19 – Hardware stores, suppliers and local distribution areas in Melbourne (Australia).	114

LIST OF TABLES

Table 1	– Symptoms of the unsustainability of current logistic systems and whether they are of economical, environmental or societal nature. The table was adapted from (Montreuil, 2011).	22
Table 2	– Summary of features and solution approach (methodology) for selected papers related to cross-docking scheduling problems. In column Docks , n represent more than one input and/or output dock. The time windows column represent any kind of time constraint regarding the serving of trucks. NA means that feature is not applicable for that paper.	39
Table 3	– Papers with high performing results on average for each of the benchmarks considered. The dash (“-”) sign indicates we have not found a paper reporting results on that benchmark. Next to each reference, we report the methodology used in the paper with the following coding. TS : Tree search, H : Constructive heuristic, MIP : MIP-based approach, MH : Meta-heuristic and CG : Column generation.	51
Table 4	– Summary of features and solution approach (methodology) for selected papers related to container loading literature. In the table we use the following coding. WL : Weight limit, WD : Weight distribution, Or : Orientation, St : Stacking, CS : Complete shipment, Al : Allocation, Po : Positioning, Sb : Stability and Co : Complexity.	53
Table 5	– Summary of parameters for CLPTAC-Om.	64
Table 6	– Summary of notation used in the formulation related to the dynamic programming.	65
Table 7	– Definition of variables for the mixed integer linear programming model.	71
Table 8	– Comparison between model (3.9)–(3.18) and DDH with $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$. TL indicates that the time limit (3600 seconds) was reached for that instance. For the DDH heuristic, the computational time was less than one second for each instance. For each objective, for each instance, best values are marked in bold.	77
Table 9	– Comparison between model (3.9)–(3.18) and Algorithm DDH with $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.4)$. TL indicates that the time limit (3600 seconds) was reached for that instance. For the DDH heuristic, the computational time was less than one second for each instance. For each objective, for each instance, best values are marked in bold.	78

Table 10 – Comparison between case 1, 2 and 3 (used as superscripts) for each instance for $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$. Results were averaged across 1000 runs of the simulation. The computational time of each run, for every case, was less than a second.	81
Table 11 – P-values for the Wilcoxon test for each pair of cases for each probability of arrival (β) considering the dispatching objective.	82
Table 12 – P-values for the Wilcoxon test for each pair of cases for each probability of arrival (β) considering the number of trucks objective.	82
Table 13 – Summary of parameters for the network scheduling problem.	88
Table 14 – Summary of the variables for the network scheduling problem.	89
Table 15 – Summary of the results comparing primal bound, gap and time to solve (s) for the decompositions NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III for <i>narrow</i> and <i>wide</i> instances <i>with</i> and <i>without</i> time window pre-processing.	95
Table 16 – Comparison of best primal and dual bounds for the model (NSCD) and the three variations of the decomposition strategy NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III. NA represents that no solution was found within the time limit (3600s). Bold values indicate best primal and dual bounds for each instance.	96
Table 17 – P-values for the different pairs of solution methods (NSCD, NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III), regarding the results for primal and dual bounds.	96
Table 18 – Comparison of gap and time to solve for the model (NSCD) and the three variations of the decomposition strategy NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III. NA represents that no solution was found within the time limit (3600s). Gap is presented as $(100 \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}})$. Bold values indicate best primal and dual bounds for each instance. TL indicates the time limit was reached for that case.	97
Table 19 – P-values for the different pairs of solution methods (NSCD, NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III), regarding the results for gap and time to solve.	97
Table 20 – Evaluation of primal bound comparing the same model with valid inequalities (cuts) for decompositions NSCD-B-MP-II and NSCD-B-MP-III. Bold values represent which of the alternatives was better for both decompositions.	98
Table 21 – Evaluation of gap $(100 \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}})$ comparing the decompositions NSCD-B-MP-II and NSCD-B-MP-III with valid inequalities (cuts) decompositions. Bold values represent which of the alternatives was better for both decompositions.	99

Table 22 – Evaluation of time to solve (s) the original model compared to the same model with valid inequalities (cuts) decompositions NSCD-B-MP-II and NSCD-B-MP-III. Bold values represent which of the alternatives was better for both decompositions. TL indicates that the time limit was reached for that case.	100
Table 23 – P-values for the Wilcoxon signed-rank test comparing the case without (<i>original</i>) and (<i>with</i>) cuts for NSCD-B-MP-II and NSCD-B-MP-III for primal bound, gap and time to solve.	100
Table 24 – Summary of the parameters for the hub selection problem.	105
Table 25 – Summary of variables used in modelling the hub selection problem. . . .	106
Table 26 – Results comparing non-collaborative, collaborative with branch-and-cut and collaborative with relax-and-fix approaches (part 1). For each approach, we report the solution value and time to reach that solution (TL if the time limit was reached). For the branch-and-cut we also report the best lower bound. Bold values represent the best solution between collaborative model for branch-and-cut with and without relax-and-fix.	111
Table 27 – Results comparing non-collaborative, collaborative with branch-and-cut and collaborative with relax-and-fix approaches (part 2). For each approach, we report the solution value and time to reach that solution (TL if the time limit was reached). For the branch-and-cut we also report the best lower bound. Bold values represent the best solution between collaborative model for branch-and-cut with and without relax-and-fix.	112
Table 28 – Results comparing non-collaborative, collaborative with branch-and-cut and collaborative with relax-and-fix approaches (part 3). For each approach, we report the solution value and time to reach that solution (TL if the time limit was reached). For the branch-and-cut we also report the best lower bound. Bold values represent the best solution between collaborative model for branch-and-cut with and without relax-and-fix.	113
Table 29 – Areas allocated to suppliers.	113
Table 30 – Average weekly requirements from suppliers to stores in each of the areas (kg).	113
Table 31 – Comparison between non-collaborative approach and collaborative approach (branch-and-cut and relax-and-fix). Savings related to the non-collaborative approach are also reported. Bold values indicate best result. The sign “-” indicate that the cell is not applicable to that particular case.	114
Table 32 – Comparison between <i>narrow</i> and <i>wide</i> instances solved by the model NSCD. We present primal bound (UB) and dual bound (LB), gap and time to solve (s). The gap is presented as $100\frac{UB-LB}{UB}$. <i>NA</i> indicates that no feasible solution was found within the time limit.	138

Table 33 – Comparison of primal bound of <i>narrow</i> instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III <i>with</i> time window preprocessing and <i>without</i> time window preprocessing. . . .	139
Table 34 – Comparison of primal bound of <i>wide</i> instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III <i>with</i> time window pre-processing and <i>without</i> time window pre-processing.	140
Table 35 – Comparison of gap, $(100 \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}})$, of <i>narrow</i> instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III <i>with</i> time window preprocessing and <i>without</i> time window preprocessing.	141
Table 36 – Comparison of gap, $(100 \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}})$, of <i>wide</i> instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III <i>with</i> time window pre-processing and <i>without</i> time window pre-processing.	142
Table 37 – Comparison of time to solve (s) <i>narrow</i> instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III <i>with</i> time window preprocessing and <i>without</i> time window preprocessing. . . .	143
Table 38 – Comparison of time to solve (s) <i>wide</i> instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III <i>with</i> time window pre-processing and <i>without</i> time window pre-processing.	144

1	INTRODUCTION	21
1.1	City logistics	22
1.1.1	<i>Container loading</i>	23
1.1.2	<i>Cross-docking facilities</i>	23
1.2	Research questions and outline of the document	24
2	LITERATURE REVIEW	27
2.1	Cross-docking	27
2.1.1	<i>Cross-docking operational problems</i>	31
2.2	Container loading problems	40
2.2.1	<i>Output maximisation problems</i>	42
2.2.2	<i>Input minimisation problems</i>	46
2.2.3	<i>Other approaches and problems</i>	48
2.2.4	<i>Benchmarks</i>	50
2.2.5	<i>Summary</i>	51
2.3	Network scheduling and collaborative freight distribution	54
2.3.1	<i>Network scheduling with consolidation centres</i>	54
2.3.2	<i>Collaborative freight distribution</i>	57
2.3.2.1	<i>Collaborative vehicle routing</i>	59
3	CONTAINER LOADING WITH TIME AVAILABILITY CONSTRAINTS	63
3.1	An output maximisation problem with time availability constraints	64
3.1.1	<i>A dynamic programming algorithm for CLPTAC-Om</i>	65
3.1.2	<i>Computational experiments for CLPTAC-Om</i>	67
3.1.2.1	<i>Results on the dynamic programming approach</i>	69
3.1.3	<i>Remarks on CLPTAC-Om</i>	70
3.2	An input minimisation problem with time availability constraints	70
3.2.1	<i>A heuristic procedure</i>	73
3.2.2	<i>Computational experiments for CLPTAC-Im</i>	74
3.2.2.1	<i>Parameter settings for the heuristics</i>	75
3.2.2.2	<i>Evaluating the model</i>	75
3.2.3	<i>Remarks on CLPTAC-Im</i>	82
3.3	General remarks on container loading with time availability constraints	83
4	NETWORK SCHEDULING WITH CROSS-DOCKING	85

4.1	Problem definition and model	87
4.1.1	<i>Valid inequalities</i>	89
4.2	Decomposition method	90
4.2.1	<i>Improving the cuts</i>	92
4.3	Computational experiments for the network scheduling problem	92
4.3.1	<i>Generating instances</i>	93
4.3.2	<i>Comparing narrow and wide instances</i>	94
4.3.3	<i>Evaluating decomposition approach</i>	95
4.3.4	<i>Evaluating the valid inequalities</i>	95
4.4	Remarks for the network scheduling problem	101
5	DESIGN OF HUB EXCHANGE NETWORKS	103
5.1	Collaborative hub exchange network	104
5.2	A heuristic method: Relax-and-fix and proximity search	107
5.2.1	<i>Repairing feasibility</i>	108
5.3	Computational experiments for the design of hub exchange networks	109
5.3.1	<i>Evaluating model and heuristic</i>	110
5.3.2	<i>Insights on different scenarios</i>	110
5.3.3	<i>Case study: Melbourne hardware retailer</i>	111
5.4	Remarks on design of hub exchange networks	114
6	CONCLUSIONS AND FUTURE WORK	117
	BIBLIOGRAPHY	121
APPENDIX A	DETAILS ON THE EXPERIMENTS FOR THE NET- WORK SCHEDULING PROBLEM	137

INTRODUCTION

We depend on logistics. Logistics is the backbone of our lifestyle ([Montreuil, Meller and Ballot, 2012](#)), enabling a global trading system with different demands ranging from import and export agreements between nations to single items purchased on the internet. Even fresh picked flowers can be moved from South America to Hong Kong in a matter of days ([Mervis, 2014](#)). According to an American report ([Nguyen *et al.*, 2015](#)), purchases and investments in transports accounted for 8.6% of the country Gross Domestic Product (GDP) in 2013. Also, the sector moved, daily, 55 million tons of freight with an aggregated value of more than U\$49.3 billion in the United States of America (USA). In Brazil, in 2010, more than 1.4 trillion tons per kilometre were transported, roughly 65% using roads ([Fleury, 2012](#)). Furthermore, the logistic industry is an important source of jobs, from 1990 to 2013, nearly 5% of the employed people worked on transportation in the USA. In the United Kingdom (UK), there were 255 thousand employed drivers of heavy goods vehicles (HGVs) ([Freight Transport Association, 2014](#)) in 2013. These numbers support the economic and social importance of the logistic sector.

Despite its importance, overall, the logistic system is unsustainable and highly inefficient. Indeed, [Montreuil \(2011\)](#) enumerated thirteen symptoms (see Table 1) of societal, environmental and economic unsustainability of current logistic systems. We highlight three of these symptoms closely related to our work: low rate of truck volume utilisation (items 1 and 2) and the difficulty of moving products in urban environments (item 10). Statistics from the UK report less than 60% of vehicle utilisation when travelling loaded and nearly 30% of vehicles travelling empty from more than 370 thousand licensed HGVs ([Freight Transport Association, 2014](#)). This inefficiency in vehicle utilisation leads to higher transportation costs, an increase in the number of trips, possibly increasing the number of vehicles and drivers needed to maintain the operations. As oil-based vehicles are the most common, with every trip there is an associate emission of CO_2 , NO_x and other pollutants. In fact, the transport sector was responsible for 23% of the total CO_2 emissions in 2010, more than 70% of that being from road transport ([Sims *et al.*, 2014](#)).

This situation is worse in urban environments, since it directly affects the population quality of life due to air and noise pollution, increase in traffic and possibility of incidents.

The planning and operation of transportation activities concern different stakeholders – e.g., suppliers, retailers, consumers, local transportation companies and population – each one of which present different challenges in managing efficiently the logistics of moving goods (and people) to, out and across the cities. These challenges motivated the concept of *city logistics*.

Table 1 – Symptoms of the unsustainability of current logistic systems and whether they are of economical, environmental or societal nature. The table was adapted from (Montreuil, 2011).

	Unsustainability symptoms	Economy	Environment	Social
1	We are shipping air and packaging.	✓	✓	
2	Empty travel is the norm rather than the exception.	✓	✓	
3	Truckers are nearly always on the road.	✓		✓
4	Products mostly sit idle, yet unavailable where needed.	✓		✓
5	Production and storage facilities are poorly used.	✓	✓	
6	So many products are never sold, never used.	✓	✓	✓
7	Products do not reach those who need them the most.	✓		✓
8	Products unnecessarily move, crisscrossing the world.	✓	✓	
9	Fast and inter-modal transport is still a dream.	✓	✓	✓
10	Getting goods in and out of cities is a nightmare.	✓	✓	✓
11	Networks are neither secure nor robust.	✓		✓
12	Smart automation and technology are hard to justify.	✓		✓
13	Innovation is strangled.	✓	✓	✓

1.1 City logistics

There are many different definitions of city logistics (Savelsbergh and Woensel, 2016b) but the core idea is to find a way of efficiently moving goods in urban environments without overlooking the negative effects on people lives. As reported by Savelsbergh and Woensel (2016b) and references therein, in 1950, about 30% of the world’s population lived in cities, by 2010, this number was more than 50% and is expected to be around 66% by 2050. These figures not only show the increasing importance but suggest that challenges in city logistics are dynamic. This dynamic nature can be illustrated by the growth of e-commerce. In the USA, according to the US Census Bureau (Hayashi, Nemoto and Visser, 2014), the business-to-consumer sales (B2C) share via e-commerce rose from less than 1% in 2010 to 5.2% in 2012, exceeding sales at department stores. Hayashi, Nemoto and Visser (2014) stressed that innovative systems and sophisticated delivery services are essential for solving last-mile transport issues arising from B2C e-commerce.

A major challenge of efficient distribution in this context is how to achieve high levels of consolidation of freights of different suppliers and shippers. A good consolidation strategy depends on managing processes in different scales in a freight distribution network: from local processes happening at a particular point inside a distribution centre to network related events. We explore some local processes which are related to loading the vehicles

(or containers) in a distribution centre (Subsection 1.1.1) and some broader aspects related to the cross-docking distribution strategy (Subsection 1.1.2).

1.1.1 Container loading

From an optimisation point of view, container loading problems fall into the broader family of cutting-stock and packing problems (Wäscher, Haußner and Schumann, 2007). These problems may be in one, two, three or more dimensions (Figure 1). In a generic description, the goal is to position geometrical pieces (e.g. boxes) inside a container (e.g. vehicle) minimising the cost of the used containers or maximising the value of the loaded (packed) pieces. Moreover, there must not be overlaps between any pair of pieces inside a container.

Figure 1 – Representation of one, two and three-dimensional packing problems.

Container loading problems, in particular, are three-dimensional versions of packing problems in which the goal is to position three-dimensional objects inside three-dimensional containers optimising one of the aforementioned metrics. Due to the NP-hardness of the problem (Pisinger, 2002) and the size of practical instances, the literature is dominated by heuristic solution approaches. In addition, many studies consider other aspects of the container loading like the stability of the cargo, the distribution of weight inside the container and the orientation of particular objects. A detailed review in container loading problems is presented in Chapter 2. In this thesis, we are interested in the particularities of container loading processes happening at outbound docks of a cross-dock.

1.1.2 Cross-docking facilities

Cross-docking facilities, or simply cross-docks (see example in Figure 2), are logistic facilities which operate to reduce storage and order picking in a distribution system. To achieve this, the whole system must be designed to synchronise the inbound and the outbound flow of trucks in the cross-docks. Ideally, this synchronisation would allow for the inbound goods to be unloaded and directly loaded into the outbound trucks. Sometimes, loads can temporarily be placed in a staging area near the dock which will serve the

Figure 2 – Schematic representation of a cross-dock.

outbound truck. Cross-docks can increase consolidation opportunities, reducing the size of the outbound fleet and, consequently, reducing traffic and noise and air pollution.

In the literature the most frequent efficiency metric used is the process makespan, i.e., the period between the first unloading of goods and the last loading of goods. This pressure for a high throughput facility might force other activities in the logistic operation to sub-optimal solutions. Particularly, we are interested in container loading activities. In the container loading process, the more boxes are available the better, potentially, is the filling rate of the containers (or trucks). But having more boxes to load imply waiting longer for the arrival of inbound trucks, damaging the makespan of the process. We explore this trade-off between a faster loading and waiting for more boxes specially in Chapter 3. The consolidation is also explored in a broader view, considering a network with cross-docks (Chapter 4) and in a collaborative distribution system (Section 5).

1.2 Research questions and outline of the document

We highlight the main gaps we explored here, at the beginning of the document, so the reader may be aware of them throughout the text. The basic underlying concept of this work is *consolidation* or, to present it as a question, *how to increase consolidation rates in freight transportation?* for three different situations: (i) at the outbound doors of cross-docks, considering the trade-off between quick dispatching and consolidation of load, (ii) when planning larger distribution networks, considering the location of cross-docks and distribution centres and the flow of commodities in the network and (iii) when integrating multiple independent distribution networks into a centrally managed one (collaborative freight distribution). We explore (i), (ii) and (iii) in chapters 3, 4 and 5, respectively. However, before that, in Chapter 2, we present a literature review in cross-docking operational aspects, container loading and network scheduling problems closely related to our research. Then, chapters 3, 4 and 5 present our contributions to container loading problems, networks with cross-docking and collaborative freight distribution, respectively.

Chapter 3 explores container loading problems. In container loading problems, we have a set of three-dimensional boxes that must be positioned inside containers with two basic requirements: (i) every box to be loaded must be fully inside its container and (ii) boxes in the same container must not overlap. Although decades of research in container loading have contemplated many practical cases, we could not find one that was particularly suitable for cross-docking environments. This is due to the fact that, in cross-docks, there is a lack of storage space leading to a fast dispatching of boxes. In this case, boxes might not be promptly available for loading. *How can we extend or adapt container loading problems' solution methods for cross-docking environments?*

Chapter 4 explores distribution networks which have multiple cross-docks. Most of the optimisation literature regarding cross-docking has focused on optimising the scheduling of inbound and/or outbound trucks or cases with networks with a single cross-dock facility. Cases with multiple cross-dock are scarce and are usually an extension of vehicle routing problems. To the best of our knowledge, optimising the distribution of goods in open networks with cross-docking in an operational context remains rather unexplored. *How can we model and solve the operation of distributing goods in networks with multiple cross-docks?*

Chapter 5 focuses on a more tactical case. In city logistics, urban consolidation centres might be used to increase the rates of consolidation, reducing the distribution costs (traffic, noise and air pollution). However, installing and maintaining such facilities might be expensive. Indeed, from our literature review, we found that many of the initiatives for using urban consolidation centres begin with public support but mostly fail once the support ends. One alternative would be for suppliers to collaborative use/share the current available logistic infrastructure. *How can we define, model and solve a problem of effectively and collaboratively use current available facilities for distributing goods?*

In chapters 3, 4 and 5, we formally define the respective problems, present models and methods for their solution and report some experiments to support our conclusions. Finally, in Chapter 6, we draw some concluding remarks and elaborate on some future work that may emerge from this PhD research.

LITERATURE REVIEW

This chapter elaborates on the literature review closely related to our research. It begins, in Section 2.1, describing the operational characteristics of cross-docking facilities (cross-docks). It also presents the most frequent optimisation problems in the cross-docking literature. Some of these operational characteristics and optimisation problems motivated the container loading problems and the network scheduling problems explored in this thesis. Section 2.2 presents a literature review on container loading problems, identifying practical constraints and various solution approaches. Section 2.3 focus on freight distribution from a network point of view. It reviews the optimisation of freight distribution networks, particularly networks which have consolidation centres/cross-docks. Moreover, it explores collaborative freight distribution systems. Sections 2.2 and 2.3 conclude identifying some research gaps which we explore in Chapters 3, 4 and 5.

Note that this is an extensive literature review. The experience reader may prefer to skip this chapter for now and refer back when reading chapters 3, 4 and 5. These chapters are related to the literature review as follows. Chapter 3 explores container loading problems (Chapter 3) in cross-docking facilities (Section 2.1). Chapter 4 explores a network scheduling problem in networks with consolidation centres (Subsection 2.3.1). Finally, Chapter 2.3 design a hub exchange network for a collaborative distribution system (Subsection 2.3.2).

2.1 Cross-docking

The definition of the logistic distribution strategy of a company depends on the market characteristics. If consumers and suppliers are geographically near to each other, two possibilities appear. The first one is the direct shipment, which consists of sending goods directly from supplier to consumer. This strategy is suitable if the demands are enough to be consolidated in (almost) full truckloads, ensuring a high volume utilisation of the vehicles. The second one is milk-runs, strategy that groups the demands, then

consumers are visited sequentially (Hosseini, Shirazi and Karimi, 2014). One advantage of milk-runs is the increase in volume utilisation. However, finding an optimal route is a NP-hard problem (Laporte, 2007). Also, both strategies may lead to a poor vehicle utilisation and long delivery times if the demands are small or suppliers and consumers are not close to each other (Buijs, Vis and Carlo, 2014). Consequently, two alternatives are used: warehousing and cross-docking.

Traditional warehousing reduces the problems due to supplier-consumer distance, transportation costs and freight consolidation, decreases the impact of uncertainty in the supply and demand and provides a location for temporarily storing material to be discarded or recycled, among others (Koster, Le-Duc and Roodbergen, 2007). In a typical warehouse, the main activities are: receiving, storing, order picking and shipping. Storing and order picking tend to be the most costly activities due to inventory holding costs and the labour intensity required (Belle, Valckenaers and Cattrysse, 2012). The cross-docking strategy aims at eliminating these two disadvantages of warehousing.

Inside a cross-docking terminal, named cross-dock, inbound vehicles with goods from different suppliers arrive and are unloaded. Then, the inbound goods are sorted according to their destination and transported across the terminal to the corresponding outbound vehicles. Once loaded, these vehicles depart immediately to their respective destinations (Stephan and Boysen, 2011). Although one of the objectives of cross-docking is to eliminate storing, it may not be possible in practical scenarios, therefore one tries to minimise it (many studies in the literature define 24 hours as a threshold for keeping goods in storage, however an exact time limit is hard to specify (Buijs, Vis and Carlo, 2014)). This strategy allows the consolidation of different small freights from different suppliers into fewer vehicles, reducing transportation costs.

Some practical examples of cross-docking usage can be found in: Wal Mart (Stalk, Evans and Shulman, 1992), Eastman Kodak Co. (Cook, Gibson and MacCurdy, 2005), Goodyear GB Ltd. (Kinneer, 2006), Dots, LLC. (Napolitano, 2011), UPS (Forger, 1995) and Toyota (Witt, 1998). Despite extensive utilisation, cross-docking is not always the best logistic strategy. In markets with highly variable demands or with high unit stock-out cost, traditional warehousing is recommended (Belle, Valckenaers and Cattrysse, 2012). According to Belle, Valckenaers and Cattrysse (2012), other factors that should be addressed while choosing a logistic strategy are: the distance between suppliers and consumers, value and life cycle of the products and demand volume. Furthermore, for an efficient implementation of cross-docking, a partnership among the supply chain members is advisable. Also, there should be trust in the quality and availability of the goods, communication among the members of the supply chain and control of the cross-dock operations. Therefore, cross-docking efficiency depends on the market characteristics, available infra-structure, engagement of the other members of the supply chain, coordination and control of the operational processes. For a quantitative evaluation of the impact of using cross-docking in inventory levels, the reader is referred to (Waller, Cassady and Ozment, 2006).

Figure 3 – Scheme of a typical cross-dock. In the facility, there are inbound docks to receive freights from the inbound trucks (suppliers S1, S2 and S3). These freights are moved from the input docks to the output docks. In the output docks, there is an outbound vehicle for each consumer or group of consumers (C1, C2, C3 and C4). This figure was adapted from Belle, Valckenaers and Cattrysse (2012).

Figure 3 shows a simple example of a cross-dock facility. The freights of consumers C1, C2, C3 and C4 are sent by suppliers S1, S2 and S3 to the cross-dock, where they are unloaded. Then, the goods are consolidated according to each consumer's request and are delivered by the outbound trucks. Some facilities have a staging area for temporarily storing the goods before the beginning of the loading process. Even in this simple example, it is possible to define some strategic, tactical and operational decision problems. Strategically, one needs to decide on location, size and layout for the cross-dock. Tactically, a suitable mix of input and output docks need to be established as well as the fleet to transport the goods. From the operational point of view, there is the need to schedule loading and unloading processes, allocate resources (machinery and staff) and to synchronise such operations in order to minimise downtimes, such as vehicles waiting to be loaded (or unloaded) or staff waiting for vehicles to arrive.

There are, also, characteristics of the logistic network which depend on the type of market. Figure 4 shows some typical configurations of cross-docking networks. According to Buijs, Vis and Carlo (2014), the configuration *few to many* is common in the retail industry, the configuration *many to few* is usual in the manufacturing industry (e.g. automobile), and the case *many to many* often appears in less-than-truckload delivery companies. Besides this categorisation, there are also *single layer* and *hub and spoke* networks. In single layer scenarios, suppliers and consumers are separated by just one cross-dock facility whilst in hub and spoke systems commodities may visit several cross-docks before reaching their destinations.

Figure 4 – Scheme with typical cross-dock networks connecting suppliers and consumers. In a *few to many* scenario, few suppliers feed a larger number of consumers. The opposite happens in *many to few* network. In a *many to many* case there is an equivalent number of suppliers and consumers. It is also possible to classify the networks according to the number of cross-docks that may be visited by freight to go from a supplier to a consumer. In *single layer* networks only one cross-dock is visited whilst more than one may be visited in *hub and spoke* systems. This figure was adapted from Buijs, Vis and Carlo (2014).

Here, due to the nature of our work, we focus on operational problems, scheduling of trucks in particular. For extensive reviews the reader is referred to Agustina, Lee and Piplani (2010), Stephan and Boysen (2011), Belle, Valckenaers and Cattrysse (2012), Buijs, Vis and Carlo (2014) and Guastaroba, Speranza and Vigo (2016). Before presenting our review, we define some terminology commonly used in the cross-docking literature. *Dock doors* (or *doors*) are specific places in the facility where inbound (also called input or receiving) and outbound (also called output or shipping) vehicles are processed (loaded or unloaded). *Preemption* of loading and unloading operations may or may not be allowed. If it is allowed, the preempted vehicle must return later to finish its (un)loading process. Also, the docks may have different *dock holding patterns*, which can be *exclusive* or *mixed*. In the exclusive pattern, a dock serves only inbound or outbound trucks, but not both, and in the mixed pattern, a dock can serve both. Boysen and Fliedner (2010) proposed a classification for the truck scheduling problem considering the different operational characteristics, contemplating dock holding patterns and preemption among others.

2.1.1 Cross-docking operational problems

In this section we focus on reviewing the literature on cross-docking operational problems, most notably optimising the scheduling of inbound and/or outbound trucks. In a sense, we present an extensive review, for a summary (and as a quick reference) of features and solution methods of selected papers the reader is referred to Table 2.

Since the 1990s, the literature on cross-docking has been growing and different strategies have been used to analyse and optimise cross-docking operations. Some papers focus on the analysis of operations, as those of Rohrer (1995), Macal and North (2005) and Magableh, Rossetti and Mason (2005) which used simulation for evaluating different scenarios. McWilliams (2005) also used simulation but within a simulation-optimisation framework. Rohrer (1995) discussed the importance of simulation for cross-docking strategies. The implementation of cross-docking may be expensive, demand a synchronisation between input and output flow of goods and specific hardware and software. Therefore, a simulation model may help evaluate strategies before their implementation. Rohrer (1995) also argued that a good cross-docking simulation model should account for delivering shipping schedules, manoeuvring of trucks in the yard, capacity of dock doors, material handling effort, failures and downtimes. Also, the output of the simulation should enable a comparison of scenarios and track the time spent on loading and unloading of goods, inventory moving and throughput of the system for individual areas (e.g. delivering, shipping and sorting). Macal and North (2005) proposed a generic discrete event simulation model for the operations within a cross-docking facility. The model contemplated resource competition at the dock doors, flexible assignments of a load to inbound and outbound doors, workforce demands and competition for material handling resources. Among the collected data were: cross-docking facility utilisation, the ratio of orders that exceeded the deadline, outbound truck waiting times and the ratio of trucks leaving with available space. The simulation model was validated against a real world cross-docking facility. McWilliams (2005) proposed a simulation-optimisation strategy to minimise the total operation time in the scheduling of inbound trucks. The simulation proposed considers loading, unloading and sorting operations. The authors compared an iterative improvement strategy against a simulated annealing method and remarked that the results showed no statistical difference in the total operation times given by these two approaches.

Vis and Roodbergen (2008) focused on the problem of positioning the goods in the staging area to minimise the total travel distance between input and output docks. In a typical I-shaped facility, with input and output docks positioned across from each other, after being unloaded, the goods are positioned in rows, waiting to be loaded. In general, a package is unloaded and staged at some row and then loaded at an output dock. If the referred row belongs to the shortest path between the respective input and output docks, then the package was optimally placed, otherwise there is additional travel distance for the process. The problem was modelled as a minimum cost flow problem and solved

using a polynomial time algorithm. [Vis and Roodbergen \(2008\)](#) compared their approach against a heuristic that placed goods in the nearest row available after unloading. The experiments contemplated different number of dock doors (input and output), rows and workloads in the input doors. The proposed method outperformed the heuristic in all the scenarios. The authors also showed how the proposal could be used in a design phase to decide the number of storage rows.

[Yu \(2002\)](#), in his thesis, identified 32 configurations of cross-docking based on the number of docks available, dock holding pattern and existence of temporary storage. The dock holding pattern specifies if docks may be used for both loading and unloading (*mixed*) or if they must be used for just one of the operations (*exclusive mode*). The author chose to investigate three of these configurations which vary in their dock holding patterns and their temporary storage capacities with a single shipping and a single receiving dock. [Yu \(2002\)](#) evaluated solution approaches and concluded that heuristics and search algorithms were able to obtain competitive results relative to those of exact approaches based on mixed integer programming, branch and bound and complete enumeration. The measure of performance was the total operation time, i.e., the difference between the time the last truck finishes its loading and the first truck starts its unloading procedure (makespan). Later, [Vahdani and Zandieh \(2010\)](#) compared genetic algorithm, tabu search, simulated annealing, electromagnetism-like algorithm and a variable neighbourhood search heuristic to schedule inbound and outbound trucks aiming to minimise the total time of the operation. The results outperformed the heuristics proposed by [Yu \(2002\)](#) regarding solution quality but with higher computational effort. The authors performed a statistical analysis and concluded that neighbourhood search algorithm and electromagnetism-like algorithm had the best solution qualities but with a higher computational cost than the alternatives.

One of the first papers dealing with truck scheduling in cross-docking environment was ([Yu and Egbelu, 2008](#)). It focuses on a scenario with one inbound and one outbound dock, with no preemption and with a staging area. The authors developed a mixed integer programming model for minimising the total operation time of the cross-dock. It decided the sequence of inbound and outbound trucks to be served at the respective doors. They also proposed nine heuristic strategies which were evaluated using a complete enumeration technique as a baseline. Using the mathematical model, only instances with small sizes could be solved efficiently and the heuristics could obtain results with 1.8% deviation from the optimal solution on average. Later, [Keshtzari, Naderi and Mehdizadeh \(2016\)](#) proposed a more effective model and a hybridised meta-heuristic combining particle swarm optimisation and simulated annealing. The model was more efficient – smaller and faster to solve – than the one proposed by [Yu and Egbelu \(2008\)](#), the heuristic outperformed the electromagnetism-like algorithm from [Vahdani and Zandieh \(2010\)](#) and the genetic algorithm from [Arabani, Ghomi and Zandieh \(2011\)](#) on randomly generated instances. [Keshtzari, Naderi and Mehdizadeh \(2016\)](#) argued that the hybridised meta-heuristic

outperformed the other solutions because it was tailor-made for the problem while the solutions by [Vahdani and Zandieh \(2010\)](#) and [Arabani, Ghomi and Zandieh \(2011\)](#) used general operators.

[Belle et al. \(2013\)](#) proposed a model for multiple inbound and outbound doors and time windows for processing the trucks. The outbound truck to load each product is assumed to be known and the goal was to minimise total operation time and tardiness. Using a tabu search, also proposed by the authors, was faster than solving the model. Their model was combined with the one from [Yu and Egbelu \(2008\)](#) by [Wisittipanich and Hengmeechai \(2017\)](#) for scheduling inbound and outbound trucks in a cross-dock with multiple inbound and outbound docks. Since [Wisittipanich and Hengmeechai's](#) model could only handle instances with moderated size, the authors experimented with two variations of the particle swarm optimisation strategy (see [Eberhart and Kennedy \(1995\)](#) and [Pongchairerks and Kachitvichyanukul \(2009\)](#)). Results showed that the variation proposed by [Pongchairerks and Kachitvichyanukul \(2009\)](#) achieved a faster convergence.

[Amini and Tavakkoli-Moghaddam \(2016\)](#) addressed truck scheduling with a probability of breakdowns. A model is developed inspired by [Yu and Egbelu \(2008\)](#) and a second objective function of maximising the reliability of the process due to truck breakdown is added. Assuming a Poisson distribution for the number of breakdowns in one unit of time revealed that maximising the reliability is related to minimising total completion time. Three meta-heuristics for multi-objective problems were evaluated, namely, Non-dominated Sorting Genetic Algorithm II (NSGA-II), Multi-Objective Simulated Annealing (MOSA) and Multi-Objective Differential Evolutionary (MODE). The results showed that MODE outperformed the other regarding five different performance metrics.

[Arabani, Ghomi and Zandieh \(2010\)](#) addressed the scheduling of inbound and outbound trucks in a cross-docking environment with a just-in-time philosophy. The key difference from similar studies is that, in just-in-time philosophy, both earliness and tardiness are harmful for the performance of the system. The authors proposed genetic, particle swarm optimisation and differential evolution algorithms to minimise a weighted sum of earliness and tardiness in the scheduling of a cross-dock with one output and one input dock. The genetic algorithm outperformed the other methods. [Assadi and Bagheri \(2016\)](#) also contemplated the scheduling in a just-in-time philosophy. The products are assumed to be interchangeable, some trucks might not be available at the beginning of the process and the transshipment from receiving to shipping doors are different for each pair of docks. The objective function was to minimise the sum of earliness and tardiness of the solution. A differential evolution and a population-based simulated annealing heuristics were proposed to solve large instances. Regarding computational time, both heuristics were similar whereas the differential evolution was able to achieve better solution qualities.

The truck scheduling problem within cross-docks can also be viewed as a flow shop problem ([Garey, Johnson and Sethi, 1976](#)). [Chen and Lee \(2009\)](#) were among the

first authors to address it from this point-of-view, formulating the problem as a two-machine flow shop problem. The inbound trucks (jobs) are processed in one inbound door, while the outbound trucks (also jobs) are processed in one outbound door and there are precedence relations among the inbound and outbound trucks. Reducing from another flow shop problem (Monma, 1980), the authors showed that the problem is NP-hard. Furthermore, they proposed a polynomial approximation algorithm and a branch and bound procedure. The branch and bound procedure was able to solve instances with up to 60 jobs in reasonable computational time. Also from a flow shop perspective, Chen and Song (2009) extended the mixed integer linear programming model proposed in Chen and Lee (2009) to consider multiple inbound and outbound doors. For larger instances, the authors proposed four heuristics based on Johnson's rule (Johnson, 1954). Later, Cota *et al.* (2016) proposed a time-indexed model that dominates the one proposed by Chen and Song (2009). The dominance is shown analytically and experimentally. Furthermore, Cota *et al.* (2016) proposed a polynomial time heuristic and demonstrated some properties regarding its solution quality. The experiments showed that the heuristics consistently outperformed the best ones from Chen and Song (2009) as the size of instances increase. The instances were generated with the same methodology of (Chen and Song, 2009), but are not the same instances.

Bellanger, Hanafi and Wilbaut (2013) extended the two-stage (receiving and shipping) flow shop problems aforementioned to a three-stage case: receiving, sorting and shipping. Aiming at the minimisation of the total operation time, the authors proposed lower bounds, used in a branch and bound framework. As the exact method was only able to solve instances with limited size, different heuristic methods were also proposed. The heuristics differ relative to the stage they focus on (receiving, sorting or shipping) and tests on different sets of instances did not reveal any clear dominance of a method over the other.

Ladier and Alpan (2018) proposed a model to minimise storage use and deviation from a previously defined schedule. They proposed an integer programming model and three heuristics. Although the model could only solve small instances, it was used in two of the heuristic methods for providing partial solutions. Evaluating the heuristics, the experiments showed that fixing the number outbound allocation usually lead to a better storage utilisation.

Ladier and Alpan (2016) considered the same scenario as Ladier and Alpan (2018)¹, in which inbound and outbound trucks have preferable time windows to be scheduled and there are penalties if the windows are violated. Ladier and Alpan (2016) proposed nine robust models assuming that the arrival times of the trucks are stochastic. The goal was not to compare the models against themselves but with the non-robust model from Ladier and Alpan (2018) in a stochastic scenario. Indeed none of the robust models dominated

¹ The paper was online in 2014, however, the current available version is dated as 2018. DOI:<<http://dx.doi.org/10.1007/s10845-014-1014-4>>

the other in all test cases, however the models based on resource redundancy gave the best results overall. The idea motivating these models is to ensure a resource (e.g. cross-dock door) will be available to deal with possible disruptions in the process.

Also exploring uncertainty in arrival times, [Konur and Golias \(2013\)](#) considered truck arrival times to be uncertain, but within time windows. The authors proposed strategies based on a risk-aversion and risk-taking decision maker, which were formulated under a bi-level programming paradigm. Using a genetic algorithm for solving the problem, they concluded that a strategy combining information of risk-aversion and risk-taking decision maker was a good alternative to the deterministic assumption.

[Arabani, Zandieh and Ghomi \(2011\)](#) analysed a multi-objective version of truck scheduling in which the goals were to minimise makespan and total lateness time. The minimisation of the makespan was based on [Yu \(2002\)](#) and the total lateness on [Yu and Egbelu \(2008\)](#). Three different genetic algorithms were proposed and they were based on: *Non-dominating sorting genetic algorithm II* (NSGA-II), *Strength Pareto evolutionary algorithm II* (SPEA-II) and *Sub-population genetic algorithm II* (SPGA-II). The authors used four measures of performances and concluded that SPEA-II can outperform NSGA-II and SPGA-II.

[Zhang et al. \(2010\)](#) also considered a multi-objective problem. The authors formulated a problem to minimise total starting and handling time of inbound trucks, total weighted travel distance by freight and total departure time. They proposed a multi-objective mixed integer linear programming. Results indicated that departure time for outbound trucks was influenced more by inbound service time than travel distance.

A combination of techniques from multi-objective evolutionary algorithms was used by [Chmielewski et al. \(2009\)](#) to minimise travel distance inside the facility and waiting times of inbound trucks. However, the evolutionary approach was outperformed by a column generation procedure also implemented by the authors. The experiments showed that the evolutionary algorithms was faster but had poor solution quality for instances with practical interest (from German logistic companies).

[Arabani, Ghomi and Zandieh \(2011\)](#) proposed five meta-heuristic algorithms for scheduling trucks in a cross-docking: genetic algorithm, tabu search, particle swarm optimisation, ant colony optimisation and differential evolution. The experiments were designed and analysed statistically and the results showed that the meta-heuristics outperformed the heuristics from [Yu and Egbelu \(2008\)](#). The differential evolution presented the best results. [Liao, Egbelu and Chang \(2012\)](#) compared the differential evolution algorithm from [Arabani, Ghomi and Zandieh \(2011\)](#) against two hybrid differential evolution algorithms. They incorporated a local search and a tabu list, respectively. Both outperformed the differential evolution algorithm with the one with the tabu list being generally better. [Liao, Egbelu and Chang \(2012\)](#) also proposed a more realistic and effective operational policy than the one in [Yu and Egbelu \(2008\)](#). The policy allowed, for example, loading

and unloading activities during the truck change operation.

Related to the scheduling problem is the truck to door assignment, namely, choosing in which door the trucks will be served (loaded or unloaded). [Miao, Lim and Ma \(2009\)](#) focused on the truck to door assignment with resource constraints and is among the few studies we could find that limited the capacity of the staging area. The authors considered a situation in which there are more trucks to be unloaded than docks to process them. They also included time windows for the trucks to be processed the objective was to minimise the total operational cost and the number of unfulfilled shipments. The objectives were combined into a single objective function for an integer programming model and two meta-heuristics were evaluated, since a commercial solver (CPLEX) could not deal with practical size instances. A genetic algorithm and a tabu search strategy outperformed CPLEX especially for medium and large sized instances. The authors concluded that the tabu search was more suitable for the problem although the genetic algorithm could also produce good solutions compared to CPLEX.

[Gelareh *et al.* \(2016\)](#) explored a similar scenario and used the same definition of variables as [Miao, Lim and Ma \(2009\)](#). The authors proposed a series of facet-defining inequalities and used them in a branch and cut for a problem with mixed docks (a dock can be used for serving inbound and outbound trucks). The algorithm proposed could solve 74% of the proposed instances at the root node and 90% of them within a 5% gap within the time limit of 1500 seconds.

[Tootkaleh, Ghomi and Sajadieh \(2016\)](#) and [Bazgosha, Ranjbar and Jamili \(2017\)](#) also limited the size of the staging area and scheduled loading and unloading operations for a multi-door scenario. [Tootkaleh, Ghomi and Sajadieh \(2016\)](#) proposed a mixed integer linear programming model and a constructive heuristic for minimising total inventory holding costs. [Bazgosha, Ranjbar and Jamili \(2017\)](#) proposed a model under the same paradigm, two constructive heuristics and meta-heuristics based on genetic algorithm, particle swarm optimisation and cuckoo optimisation. Combining the cuckoo optimisation and one of the constructive heuristics yielded the best results for minimising total operation time.

[Madani-Isfahani, Tavakkoli-Moghaddam and Naderi \(2014\)](#) explored a scenario with limited cross-dock capacity and multiple cross-docks. The authors proposed a mixed integer linear programming model to decide the scheduling in each facility and also to which facility a truck should go to minimise total operation time. To handle bigger instances, they proposed heuristics based on simulated annealing and firefly algorithms. The latter consistently provided better results.

[Shakeri *et al.* \(2012\)](#) also dealt with a resource constrained operational scenario. The authors limited the number of dock doors and resources available for transporting goods from inbound to outbound docks. Furthermore, it was considered the possibility of having more trucks to be unloaded than available dock doors. The authors proposed

an integer programming model with the goal of minimising the total operation time. The model solves three sub-problems, namely, assignment of trailers to dock doors, sequencing the trailers assigned to the same dock door and scheduling the movement of trailer inside the cross-dock. Due to the size of the instances, the authors proposed a two-phase heuristic. In the first phase, there is a search for a feasible trailer sequence, then the second phase assigns trailers to doors according to the previously defined sequence. The proposed heuristic outperformed a commercial solver in solution quality and runtime.

As [Shakeri *et al.* \(2012\)](#), [Shiguemoto, Netto and Bauab \(2014\)](#) considered a case with preemption. The authors proposed two models for the problem: an integer programming model and a constraint programming model. Both could only handle instances of moderate size, although the constraint programming model achieved better solutions, in general. A genetic algorithm was able to outperform both approaches.

[Hermel *et al.* \(2016\)](#) proposed a framework for integrating dock door assignment and scheduling problems at a cross-docking terminal contemplating resource requirements and constraints. The aim was to minimise the total operation time while handling constraints related to material flow inside a terminal. The authors proposed a hierarchical approach which grouped similar containers, assigned containers to dock doors, scheduled the workflow for each cluster and then performed the scheduling of containers to minimise the total operation time. Many clustering techniques are used to reduce problem size and the solution showed an improvement of roughly 40% on operation time and weighted travel distance of material compared to the solution of an industrial partner.

[Tavana *et al.* \(2017\)](#) compared truck and drone delivery in a scenario allowing for direct shipment using drones. The authors proposed a model for truck scheduling which contemplated two objective functions (total operation time and total costs), multiple products, multiple doors and multiple fleet. Efficient solutions showed that direct shipment using drones could be an alternative depending on the cost of their operations.

Another important measure of efficiency is the number of double handlings inside a cross-dock, namely, the number of times a box need to be touched twice. [Maknoon, Soumis and Baptiste \(2017\)](#) proposed a branch and bound to minimise the number of products which were placed in the staging area before being loaded. A linear integer programming model was proposed with strategies for pre-processing the model, heuristics for finding initial solutions and branching rules. The contribution of each phase of the solution procedure was analysed on randomly generated instances.

More recently, [Molavi, Shahmardan and Sajadieh \(2018\)](#) proposed a model and meta-heuristics to explore a case with hard time constraints. The objective of the problem is to minimise the cost of delayed and non-delivered products. They highlight that strict time constraints are a practical application (i.e. postal services) and suggest a combination of a genetic algorithm and variable neighbourhood search for obtaining good solutions.

In another variation, [Fard and Vahdani \(2019\)](#) proposed a bi-objective model to

minimise total operation time and energy consumption of forklifts. The authors consider a cross-dock with multiple doors in which a waiting queue might appear. Also, they accounted for the energy consumption of the forklifts when travelling with and without loads. Using different multi-objective algorithms they analysed the sensibility of the system to multiple parameters and concluded, for example, that the distances between inbound and outbound doors were one of the most critical factors in the energy consumption of the forklifts, arguing in favour of a good layout design of the facility.

In general, there are different scenarios contemplated in the literature of cross-docking operational problems. Most of them consider the exclusive dock holding pattern, the exception being [Magableh, Rossetti and Mason \(2005\)](#). The number of inbound and outbound docks also differ from papers assuming only one inbound and one outbound door (e. g. [Yu \(2002\)](#), [Yu and Egbelu \(2008\)](#), [Chen and Lee \(2009\)](#), [Vahdani and Zandieh \(2010\)](#), [Arabani, Ghomi and Zandieh \(2011\)](#) and [\(Amini and Tavakkoli-Moghaddam, 2016\)](#)) to multiple inbound and outbound doors (e. g. [McWilliams \(2005\)](#), [Vis and Roodbergen \(2008\)](#), [Chen and Song \(2009\)](#), [Shakeri *et al.* \(2012\)](#), [Cota *et al.* \(2016\)](#) and [Assadi and Bagheri \(2016\)](#)). In practice, the one door scenarios have limitations, but they are important for understanding structural properties of the models. Most of the literature does not consider the possibility of preemption, two exceptions found were [Yu \(2002\)](#) and [Shakeri *et al.* \(2012\)](#). Furthermore, the staging area utilisation may be allowed (e. g. [Magableh, Rossetti and Mason \(2005\)](#), [Vis and Roodbergen \(2008\)](#), [Yu and Egbelu \(2008\)](#), [Vahdani and Zandieh \(2010\)](#), [Assadi and Bagheri \(2016\)](#) and [Amini and Tavakkoli-Moghaddam \(2016\)](#)) or prohibited (e. g. [McWilliams \(2005\)](#), [Chen and Song \(2009\)](#), [Chen and Lee \(2009\)](#), [Arabani, Zandieh and Ghomi \(2011\)](#), [Hermel *et al.* \(2016\)](#) and [Cota *et al.* \(2016\)](#)). Additional literature on cross-docks as part of a larger distribution network will be presented in Section 2.3.1. Table 2 presents a summary of features for selected papers related to the cross-dock scheduling literature.

Table 2 – Summary of features and solution approach (methodology) for selected papers related to cross-docking scheduling problems. In column **Docks**, n represent more than one input and/or output dock. The **time windows** column represent any kind of time constraint regarding the serving of trucks. NA means that feature is not applicable for that paper.

Paper	Docks	Holding pattern	Preemption	Staging area	Time windows	Available from beginning?	Model	Methodology
(Yu, 2002)	1	Exclusive	Both	Both	No	Yes	Yes	Meta-heuristics
(McWilliams, 2005)	n	Exclusive	No	No	No	Yes	Simulation	Meta-heuristics
(Yu and Egbelu, 2008)	1	Exclusive	No	Yes	No	Yes	Yes	Constructive
(Chmielewski <i>et al.</i> , 2009)	n	Exclusive	No	Yes	No	Yes	Yes	Exact method and meta-heuristics
(Chen and Song, 2009)	1	Exclusive	No	No	No	Yes	Yes	Constructive
(Miao, Lim and Ma, 2009)	n	Shared	No	Yes	Yes	No	Yes	Meta-heuristics
(Chen and Lee, 2009)	1	Exclusive	No	No	No	Yes	No	Exact method
(Zhang <i>et al.</i> , 2010)	1	Exclusive	No	Yes	Yes	No	Yes	Exact method
(Arabani, Ghomi and Zandieh, 2010)	1	Exclusive	No	Yes	No	Yes	No	Meta-heuristics
(Vahdani and Zandieh, 2010)	1	Exclusive	No	Yes	No	Yes	Yes	Meta-heuristics
(Arabani, Zandieh and Ghomi, 2011)	1	Exclusive	No	Yes	No	Yes	No	Meta-heuristics
(Arabani, Ghomi and Zandieh, 2011)	1	Exclusive	No	Yes	No	Yes	Yes	Meta-heuristics
(Liao, Egbelu and Chang, 2012)	1	Exclusive	No	Yes	No	Yes	No	Meta-heuristics
(Shakeri <i>et al.</i> , 2012)	n	Shared	No	Yes	No	Yes	Yes	Constructive
(Bellanger, Hanafi and Wilbaut, 2013)	n	Exclusive	No	Yes	No	Yes	No	Various
(Konur and Golias, 2013)	n	Exclusive	No	NA	Yes	No	Yes	Meta-heuristics
(Belle <i>et al.</i> , 2013)	n	Exclusive	No	Yes	Yes	No	Yes	Meta-heuristics
(Madani-Isfahani, Tavakkoli-Moghaddam and Naderi, 2014)	1	Exclusive	No	Yes	No	Yes	Yes	Meta-heuristics
(Tootkaleh, Ghomi and Sajadieh, 2016)	n	Exclusive	No	Yes	Yes	Yes	Yes	MIP model and constructive heuristic
(Amini and Tavakkoli-Moghaddam, 2016)	1	Exclusive	No	Yes	Yes	Yes	Yes	Meta-heuristics
(Assadi and Bagheri, 2016)	n	Exclusive	No	Yes	Yes	No	Yes	Meta-heuristics
(Cota <i>et al.</i> , 2016)	n	Exclusive	No	Yes	No	Yes	Yes	Constructive
(Hermel <i>et al.</i> , 2016)	n	Mixed	No	Yes	No	Yes	Yes	Hierarchical
(Keshtzari, Naderi and Mehdizadeh, 2016)	1	Exclusive	No	Yes	No	Yes	Yes	Meta-heuristics
(Ladier and Alpan, 2016)	n	Exclusive	No	Yes	Yes	No	Yes	Robust models
(Gelareh <i>et al.</i> , 2016)	n	Mixed	No	Yes	No	No	Yes	Exact method
(Bazgosha, Ranjbar and Jamili, 2017)	n	Exclusive	No	Yes	Yes	No	Yes	Constructive and Meta-heuristics
(Wisittipanich and Hengmeechai, 2017)	n	Exclusive	No	Yes	No	Yes	Yes	Meta-heuristics
(Tavana <i>et al.</i> , 2017)	n	Exclusive	No	Yes	No	Yes	Yes	MIP model
(Molavi, Shahmardan and Sajadieh, 2018)	n	Exclusive	No	No	Yes	No	Yes	Meta-heuristics
(Ladier and Alpan, 2018)	n	Exclusive	No	Yes	Yes	No	Yes	Math-heuristics and meta-heuristics

2.2 Container loading problems

Container loading problems can be classified within the family of cutting-stock and packing problems. They appear in many contexts in the optimisation literature and real-world problems. They can be applied to divide a resource into smaller pieces. This resource can be a roll of paper, a sheet of glass, an empty space in a pallet or container (Sweeney and Paternoster, 1992). Wäscher, Haußner and Schumann (2007) proposed a typology for classifying cutting-stock and packing problems, which is summarised in Figure 5.

Figure 5 – Summary of the typology for cutting-stock and packing problems (Wäscher, Haußner and Schumann, 2007).

In this typology, a problem can be classified according to five criteria. Criteria *kind of assignment*, *assortment of items* and *assortment of large objects* define a basic type. In case *dimensionality* is also specified then we have an intermediate problem type. Finally, by adding *shape of small items*, we have a refined problem type. In our context, large objects are the containers, which will accommodate small items (boxes).

We focus the review on container loading problems (CLPs), a packing problem in which one wants to load boxes into one or more larger boxes (containers). We describe the five criteria with CLPs in mind as follows:

1. *kind of assignment*: Output maximisation or input minimisation. In output maximisation problems, there are not sufficient containers to load all boxes and the goal is to select a subset of boxes that maximises the value (or volume) loaded. On the other hand, in input minimisation problems, there is a set of boxes to be loaded and the goal is to load all boxes with the least expensive set of containers.
2. *assortment of items*: Identical, weakly heterogeneous or strongly heterogeneous. In problems with identical assortment of items, all the boxes (items to be loaded) have the same dimensions. The difference between weakly and strongly heterogeneous is imprecise. An informal notion of the heterogeneity concept is: If there are many boxes of a few different types then the set of boxes is weakly heterogeneous, if there are a few boxes of many different types then the set is strongly heterogeneous.

3. *assortment of large objects*: One or several large objects. If there is one container to accommodate the boxes then the problem is classified as having one large object, otherwise it has several large objects. In CLPs, if there is more than one container to load the objects, the problem is also called multiple container loading problems.
4. *dimensionality*: 1D, 2D, 3D or multi-dimensional. Container loading problems are usually three-dimensional. However, existing decomposition approaches solve classes of 3D problems by solving multiple 1D or 2D problems.
5. *shape of small items*: Regular or irregular. The items to be packed can be regular (e.g. rectangles, circles, spheres) or irregular (e.g. star-shaped). It is also common in the literature to consider that rectangular items, like boxes, are to be positioned with faces orthogonal to wall of the large object.

In the literature, it is common to find variations regarding *kind of assignment*, *assortment of items* and *assortment of large objects*. Figure 6 summarises output maximisation and input minimisation problems classified according to the heterogeneity of boxes and cargo for container loading problems. For input minimisation problems, there are also Open Dimension Problems (ODPs), not shown in the figure. In such problems, the goal is to load cargo into a container which has one or more dimensions as variable(s).

Figure 6 – Classification scheme for container loading problems based on heterogeneity of cargo and boxes. This figure was inspired on (Wäscher, Haußner and Schumann, 2007).

The vast majority of research in container loading problems deals with regular items – some exceptions are Egeblad, Nielsen and Odgaard (2007), Egeblad, Nielsen and Brazil (2009) and Egeblad *et al.* (2010). Also, container loading is usually a 3D packing problem. Regarding criteria *kind of assignment*, *assortment of items* and *assortment of large objects*, there is a considerable amount of research for each combination of their characteristics. Furthermore, numerous approaches can be applied – or adapted – to cope with more than one class of problem. Bortfeldt and Wäscher (2013) presented a review on container loading problems classifying the papers according to Wäscher, Haußner and Schumann’s typology (2007). The review is complemented by Zhao *et al.* (2016) which reviewed the solutions methodologies applied to container loading problems. Moreover,

many papers dealing with container loading incorporate *practical constraints*. These practical constraints aim at enhancing the practical feasibility of the solution, considering, for example, the stability of the cargo, the orientation of the boxes, weight limit and weight distribution in the containers.

One of the firsts analytical models for a container loading problem was proposed by [Chen, Lee and Shen \(1995\)](#). It can handle orientation of boxes, multiple containers and can be modified to cope with issues like weight balance and variable container length. There are other exact approaches, but, since the container loading problem is NP-hard ([Pisinger, 2002](#)) and the fact that practical instances are usually too large to be tackled by exact methods, researchers have focused on heuristic approaches. [Zhu et al. \(2012\)](#) grouped these heuristic approaches in three classes, not necessarily disjoint, according to how the loading plans are generated: *constructive*, *divide-and-conquer* and *local search*.

Here, we choose to group the articles reviewed by the kind of assignment. If only a subset of small items can be loaded, we have an *output maximisation problem* (Subsection 2.2.1), if there are enough containers to accommodate all items we have an *input minimisation problem* (Subsection 2.2.2). There are few papers which dealt with both or integrated other problems like routing. They will be discussed in Subsection 2.2.3. We also describe the benchmark instances usually used in computational experiments in Subsection 2.2.4. Subsection 2.2.5 presents a summary of the literature review.

2.2.1 Output maximisation problems

In output maximisation problems, there are not enough containers for accommodating all the boxes, thus the goal is to output a loading plan (how to position each box chosen for loading) that maximises the total volume (or value) of boxes inside the available containers (or minimises unused spaces).

[George and Robinson \(1980\)](#) were probably the first to consider this version of the problem. The authors proposed a heuristic approach to load the boxes layer by layer into the container. In order to create a loading plan, they devised strategies to create and handle layers, empty spaces and to choose the sequence of boxes to be loaded. The same general principles were used in many other papers (e.g. ([Davies and Bischoff, 1999](#)), ([Moura and Oliveira, 2005](#)), ([Zhu et al., 2012](#)), ([Wang, Lim and Zhu, 2013](#)) and ([Domingo, Ponnambalam and Kanagaraj, 2013](#))).

An improvement in [George and Robinson's \(1980\)](#) heuristic was suggested by [Moura and Oliveira \(2005\)](#). They also filled the container layer by layer, however the new heuristic modifies the loading strategy in the final layers and the way it deals with the length of the container. The initial solution found is improved with a randomised local search procedure. The solution outperformed other meta-heuristics for most of the benchmarks used in testing for volume utilisation and stability of the cargo.

The first modelling approach is attributed to [Tsai \(1987\)](#) (according to [Bortfeldt and Wäscher \(2013\)](#)). The author was motivated by pallet (not container) loading and proposed a mixed-integer programming model for placing boxes into a pallet while preventing overlaps. Solving the model was time consuming, specially considering the computational power of the late 1980s. Since the goal was to embed the solution procedure on a robotic system, the author also proposed a heuristic dynamic programming approach to obtain “good” solutions faster. Later, [Chen, Lee and Shen \(1995\)](#) proposed a mixed integer model able to deal with different variants of container loading problem. [Chen, Lee and Shen](#)’s model generalises the ideas contained in ([Tsai, 1987](#)). Improving [Chen, Lee and Shen](#)’s model, [Tsai and Li \(2006\)](#) reduced the number of variables and used the model to solve an open dimension problem.

The use of integer programming models was also explored by [Fasano \(1999\)](#) and [Padberg \(2000\)](#). In fact, the container loading model presented by [Fasano \(1999\)](#) may be considered a particular case of the model by [Chen, Lee and Shen \(1995\)](#). It is inspired in a space engineering scenario arising at the International Space Station ([Jr, 1999](#)) and is basically a feasibility problem of checking if the required boxes fit into a container. [Padberg \(2000\)](#) proposed an improvement on [Fasano](#)’s model. This improvement was demonstrated using a polyhedral analysis and some valid inequalities were also proposed, although the author did not report any experiments.

[Morabito and Arenales \(1994\)](#) generalised previous approaches for lower dimension problems from [Gilmore and Gomory \(1965\)](#) and [Morabito and Arenales \(1992\)](#) to the three-dimensional case. The approach implicitly enumerated guillotine loading patterns using heuristics and search strategies to reduce the search space.

Not many papers dissociate value and volume of the boxes to be loaded. The earliest one contemplated in this review is ([Mohanty, Mathur and Ivancic, 1994](#)). The authors proposed an integer programming based heuristic procedure which used the fractional knapsack problem to generate packing patterns. The authors obtained better results for some combinations of container and boxes dimensions. [Zheng, Chien and Gen \(2015\)](#) also explored the dissociation between value and volume of boxes. The authors proposed a multi-population genetic algorithm that maximises both: the value and the volume loaded. It is one of the few papers approaching a container loading problem in a multi-criteria paradigm. [Takahara \(2008\)](#) also explored the volume and the value of boxes in a multiple container loading problems. The authors proposed a multi-start local search combined with a greedy heuristic. The goal of the strategy was to maximise the volume and the value of loaded boxes. Results showed that it was a competitive method.

Highlighting the importance of practical considerations for container loading problems, [Davies and Bischoff \(1999\)](#) proposed an heuristic of swapping and rotating layers to achieve an even distribution of weight inside the container. The heuristic was applied as a post-processing technique once a solution for the loading problem was obtained. The

authors argued that an even distribution of weight could be reached without much decrease in volume utilisation.

Also following the concept of generating patterns to be aggregated in a loading plan, Brunetta and Grégoire (2005) combined/enhanced the ideas of Morabito and Morales (1998) and Pisinger (1998) (as cited by Morabito and Morales (1998)). In the approach, the patterns from Morabito and Morales (1998) were used to load multiple containers and the heuristics from Pisinger (1998) ensured the demands are satisfied. According to the authors, the method was able to produce adequate solutions for their industrial partner.

Other techniques that generate guillotine patterns were proposed by Pisinger (2002), Fanslau and Bortfeldt (2010) and Sheng *et al.* (2013). Pisinger (2002) subdivided each layer in strips and each strip is filled optimally as a knapsack problem. The approach to fill the layers included a backtracking step to improve the solution. One of the decisions in the approach regards the depth of the layers and strips, which were given considering 27 different ranking functions. Sheng *et al.* (2013) used a tree search approach and considered the full support constraint. The main conceptual difference from Pisinger (2002) was in the definition of layers orientation inside the container: while Pisinger (2002) only allowed layers along the length of the container, Sheng *et al.* (2013) allowed them along length and width which increased the search space. The algorithm performed better on strongly heterogeneous instances – since Pisinger (2002) did not consider support constraints, the results of Sheng *et al.* (2013) have a smaller mean volume utilisation. Fanslau and Bortfeldt (2010), in turn, used the concept of generalised blocks to assemble blocks with different types of boxes and also used a search tree algorithm. The results showed the best volume utilisation among four other techniques.

Chien *et al.* (2009) proposed a two-phase algorithm to approach a single container loading problem. In the first phase, the loading patterns of the layers are created and, in the second phase, the layers are combined into a loading pattern. Then, the best loading pattern among the generated solutions is selected. The study was based on a real case of an automotive industry and the solution developed outperformed the previously manual solution used by the company.

Zhu *et al.* (2012) used an iterative block-building approach. In each iteration, boxes were combined into a block which was placed inside the container. This process creates available spaces to place the next block, the selection of the next space favoured spaces that are near the corner of the container. The authors also used a tree search to generate several complete packings and selected the best one. Furthermore, they adapted the approach to deal with full support constraints. The solution showed improvement over previous approaches, even with fewer computational resources.

Also combining block building and tree search, Ren, Tian and Sawaragi (2011) and Wang, Lim and Zhu (2013) proposed approaches for a container loading problem with shipment priorities. In this problem, there is a subset of boxes which must be loaded

beforehand. [Ren, Tian and Sawaragi \(2011\)](#) used a tree search procedure to improve the solution of a greedy heuristic and the authors concluded that high volume utilisation can be obtained despite the addition of shipment priorities. Later, [Wang, Lim and Zhu \(2013\)](#) used a beam search based approach to explore the tree of solutions, which outperformed the strategy of [Ren, Tian and Sawaragi \(2011\)](#). The container loading problem with shipment priorities was called less-than-container loading problem by [Jamrus and Chien \(2016\)](#) to refer to companies transporting a small number of goods for its client but do not want to mix goods from different costumers. The authors devised a hybrid genetic algorithm with a fuzzy logic controller for auto-tuning its parameters. According to the authors, the results showed practical viability.

In a layer building approach, [Gonçalves and Resende \(2012\)](#) proposed a biased random-key genetic algorithm to determine the type of layer to be loaded next. The solution can handle supported and unsupported cases and achieved the best average results among 13 other solution techniques. Using a similar approach with a differential evolution mechanism ([Storn and Price, 1997](#)), [Domingo, Ponnambalam and Kanagaraj \(2013\)](#) achieved comparable results.

[Junqueira, Morabito and Yamashita \(2012b\)](#) proposed mixed integer programming models that explicit cope with loading bearing, vertical and horizontal stability constraints. The models were able to handle instances of moderated size. The same limitation emerged in [Junqueira, Morabito and Yamashita \(2012a\)](#) in which the authors considered vertical stability and multi-drop constraints. Multi-drop constraints model the situation in which cargo is loaded in order to simplify the unloading process according to its destination. For example, if the loading was done carelessly, one might need to unload several boxes to unload the one that satisfies a particular client demand. Besides considering stability and multi-drop constraints, a difference of these two papers from the previous studies focusing on models (such as ([Fasano, 1999](#)) and ([Padberg, 2000](#))) is that they used a discrete paradigm (inspired on the 2D case ([Beasley, 1985](#))), namely, the boxes are allowed to be positioned on predefined points inside the container. Previous works considered the boxes could be positioned continuously along containers axes.

Another practical constraint which the literature contemplates is the so called axle weight constraint. It emerges when considering the legislation regarding weight per axle of a vehicle. [Lim *et al.* \(2013\)](#) addressed this issue. The authors argued that a heuristic approach was more suitable than the alternative of trying to solve a large non-linear program. Therefore, the authors integrated a randomised search wall-building approach with procedures that solve few linear mixed integer programming models. The search was a modification of the proposal of [Moura and Oliveira \(2005\)](#) to deal with the weight on the axles. More recently, [Ramos, Silva and Oliveira \(2018\)](#) included the axle weight distribution as a feature and used a biased random-key genetic algorithm to solve the problem. The experiments showed that it is possible to consider the distribution of weight without compromising volume utilisation.

Using elements from [Moura and Oliveira \(2005\)](#), [Fanslau and Bortfeldt \(2010\)](#) and [Zhu, Huang and Lim \(2012\)](#), [Araya and Riff \(2014\)](#) proposed a beam search algorithm in which a state is represented by a subset of boxes already loaded into the container and there is a mechanism to remove similar states. The approach is also able to handle full support constraints. The results showed that the algorithm could achieve the best mean performance in the groups of benchmarks tested. Later, [Araya, Guerrero and Nuñez \(2017\)](#) refined the strategy for ranking and selecting blocks to be loaded, further enhancing the results.

A particular situation was explored by [Sheng *et al.* \(2016\)](#). The authors explored the problem of loading pallets into containers. By the definition of the problem, a subset of pallets which maximise utilisation should be loaded. Then, some of the remaining pallets may be disassembled and individual goods may be placed into the container as long as all it is possible to fit all the goods – it is assumed that a re-palletisation process can be performed at the unloading site. The authors proposed a tree search for the loading the pallets in the first step and a greedy heuristic for the second. We could not find any other paper exploring this specific scenario.

Another scenario that is not much explored is the dynamic packing. [Hemminki, Leipala and Nevalainen \(1998\)](#) compared first-fit and best-fit algorithms for arranging boxes into pallets assuming that the next box to be placed only becomes known when the previous one is already accommodated. This simulates boxes being transported to the loading place by a conveyor belt. The best-fit algorithm usually performed better than the first-fit. [Abdou and Elmasry \(1999\)](#) also explored an online scenario, but with temporary holding area in which boxes could be placed before loading. According to the authors, near optimal solutions could be obtained mixing layered and stacking procedures.

2.2.2 *Input minimisation problems*

In these problems, all boxes need to be loaded and each type of container has an associated cost. The goal is to devise a loading plan (where to load each box) that minimises the total cost. If all containers' costs are equal, the problem is equivalent to minimising the number of containers to load all the boxes. As cited by [George and Robinson \(1980\)](#), probably the first paper to deal with this problem explicitly was [Tinarelli and Addonizio \(1978\)](#).

The first exact algorithm for the three-dimensional bin packing problem (a basic version of CLPs) was proposed by [Martello, Pisinger and Vigo \(2000\)](#). The authors developed a lower bounding procedure for the problem, proposed an approximation and a branch and bound algorithm.

Extending the model proposed by [Eley \(2003\)](#) for multiple container loading, [Che *et al.* \(2011\)](#) proposed a parameter which controls the percentage of boxes to be loaded. They also proposed three strategies for generating loading patterns (solutions for a single

container loading problem) for a mixed linear integer programming model, which selects the best patterns to fulfil the demand for boxes. The first tries to efficiently fill the container layer by layer. The second and third are based on a Greedy Randomised Adaptive Search Procedure (GRASP) to select the boxes to be loaded and generate a pattern. The results showed that the average number of containers used by this approach was smaller than previously reported results. Also based on Eley's model (2003), Zhu, Huang and Lim (2012) proposed a column generation technique for a multiple container loading problem. The problem was formulated as a set cover problem and pricing problems were relaxed to be solved. The relaxed pricing problem gave an approximated solution (column). Then, a hill climbing algorithm converted the column to a feasible solution. The results had a better solution quality the reported by Che *et al.* (2011) in comparable computational times. This approach was further improved by Wei, Zhu and Lim (2015), in which a search among the available containers was performed to find cheaper solutions.

Tian *et al.* (2016) introduced the multiple container loading problems with preference. In this variation, the value (preference) of products loaded is indicated by a combination of delivery deadline, shelf life and monetary value. A combinatorial formulation was proposed with a two phase heuristic to solve it. The first phase chooses a promising region of the solution space to be explored and, in the second phase, the algorithm searches the best solution in the promising region.

Alonso *et al.* (2016) were one of the few references we could find that consider axle weight constraints in the input minimisation problem. They developed a Greedy Randomised Adaptive Search Procedure (GRASP) that was tested on real instances. Alonso *et al.* (2017) also consider axle weight constraints and devise a solution procedure that starts with basic features of the problem (e.g. packing with a layered strategy) and successively includes new features (e.g. axle weight constraints and centre of gravity).

Besides the research in solution methods for the problem, some decision support systems can also be found in the literature. Two examples are: Fraser and George (1994) and Chan *et al.* (2006). Fraser and George (1994) proposed a software whose inputs were the orders for paper products in an industry and the output was a loading plan for sheets and reels into multiple containers. The loading algorithms were a combination of two-dimensional loading strategies for circles and an enumeration heuristic for packing the reels. In their approach, loading patterns were manually generated. Chan *et al.* (2006) developed a system which goal was to assist an air-cargo loading procedure. The main difference from usual container loading problems is that the cross-section of a plane changes along its height. The developed system contemplated constraints related to pallet loading, to the airline company and costumers preference. The proposed solution approach used a linear programming model, as a lower bounding procedure, and a heuristic method which tries to obtain a solution as close as possible to the lower bound defined by the model. The authors claim that the system improved the loading plan and reduced the manual time-consuming task of creating it.

Correcher *et al.* (2017) proposed a solution to a real-world problem in the car manufacturing industry. One of the challenges of the problem was the number of practical constraints which had to be considered simultaneously. They were related to weight limit, orientation, stacking, stability and complexity. The authors proposed a GRASP strategy that improved some solutions used by the company. Paquay *et al.* (2018a) also explored a practical situation. The authors are among the few that considered different container sizes, proposing three MIP-based constructive heuristic that accounted for weight limit, orientation, stacking and orientation constraints. The three heuristics outperformed CPLEX and two of them could compute a feasible solution quickly.

2.2.3 Other approaches and problems

In this section, we summarise papers which do not fit clearly in subsections 2.2.1 nor 2.2.2. We have identified some papers that dealt with input minimisation and output maximisation problems (Eley; Eley; Lim and Zhang; Ceschia and Schaerf, 2002, 2003, 2005, 2013) and a variant of container loading problem called *capacitated vehicle routing problem with container loading constraints* (3L-CVRP) which integrates the routing and loading decisions, giving container loading problems a logistic flavour.

Eley (2002) dealt with a single container using a greedy heuristic within a tree search procedure and evaluated the volume utilisation. The approach was then adapted to deal with a multiple container case and the number of containers used was evaluated. The proposed approaches were competitive with others from the literature regarding volume utilisation, load stability and weight distribution. They used the same greedy heuristic to generate patterns for integer programming models for input minimisation and output maximisation problems. Eley (2003) also adapted the method to cope with separation and complete shipment constraints. Lim and Zhang (2005) proposed a greedy heuristic, combined with a tree search, in which the priority of the boxes to be loaded changed dynamically to accommodate “strange” boxes (box that are relatively large in relation to the dimensions of the container). The authors analysed the output maximisation case with one container and then, adapted the approach to cope with multiple containers in an input minimisation case. Ceschia and Schaerf (2013) contemplated box rotations, loading bearing strength, full support constraints, container weight limit and multi-drop constraints. Their approach was based on a local search in which they found that a simulated annealing strategy usually outperformed tabu search. Also, they were able to obtain an average increase of more than 10% over literature approaches for a real world application. They also evaluated volume utilisation using one container and number of container used to load all boxes.

Gendreau *et al.* (2006) integrated a capacitated vehicle routing problem (CVRP) with a container loading problem (3L-CVRP). In the vehicle routing problem, each client has a demand that must be satisfied by a fleet that should be routed in an efficient manner.

It is a practical requirement to account for the ease in the unloading procedure. [Gendreau et al. \(2006\)](#) proposed tabu search algorithms for the container loading problem and for the integrated problem. The proposed heuristics were robust to parameter settings and the authors state they yielded interesting results for the generated instances and for the problem of an industrial partner. [Tarantilis, Zachariadis and Kiranoudis \(2009\)](#) also dealt with the 3L-CVRP. In their paper, the routing problem was approached using tabu search and guided local search while the loading problem was solved using a collection of six packing heuristics. This strategy was inspired by [Zachariadis, Tarantilis and Kiranoudis \(2009\)](#), which also proposed a combination of tabu search and guided local search for the two-dimensional version of the same problem. [Tarantilis, Zachariadis and Kiranoudis \(2009\)](#) improved the results found in ([Gendreau et al., 2006](#)). Both approaches to the 3L-CVRP (([Gendreau et al., 2006](#)) and ([Tarantilis, Zachariadis and Kiranoudis, 2009](#))) were improved by an hybrid strategy ([Bortfeldt, 2012](#)), which consisted of a tabu search for the routing problem and a tree search ([Fanslau and Bortfeldt, 2010](#)) for the container loading problem. Results showed improvements in solution quality and computational effort.

The 3L-CVRP with practical loading constraints was also explored. [Junqueira et al. \(2013\)](#) proposed mixed integer programming models accounting for stability, multi-drop and load bearing strength of the boxes. The experiments highlighted the difficulty in solving the moderated size instances, specially as practical constraints are included.

Extending the 3L-CVRP, [Moura and Oliveira \(2009\)](#) considered time windows for each served client. The authors presented a mixed integer programming model for placing the boxes into vehicles minding the unloading procedure and with the objective of minimising the number of vehicles and total route time. Two solution approaches were proposed: a sequential method and a hierarchical method. In the sequential method, the container loading and vehicle routing problems are solved together and the authors evaluated a Monte Carlo procedure, a local search heuristic and a GRASP heuristic. In the hierarchical method, the container loading is considered as a sub-problem of the vehicle routing and a three-phase solution approach was proposed. In the first phase, a constructive GRASP from the literature is applied to build the routes. Then, the algorithm tries to reduce the number of routes and, in the third phase, a local search is performed to reduce total travel time. In the sequential procedure, the GRASP approach outperformed the others and there was no clear dominance between sequential and hierarchical approaches.

Besides the CVRP, the *pickup and delivery problem* (see the survey by [Parragh, Doerner and Hartl \(2008\)](#)) was also integrated with CLP. [Dirk and Bortfeldt \(2016\)](#) aimed at finding a set of routes with minimum cost for a homogeneous fleet satisfying a set of requests. These requests are 3D boxes that must be moved from a loading site to the respective unloading site. The authors discussed that some reloading effort (e.g. rotating some boxes after an unload operation) might be desirable and introduced some variants of the *vehicle routing problem with pickup and delivery and three-dimensional loading constraints* (3L-PDP), which allow different levels of effort in performing extra

handling of the cargo. The authors have adapted a large neighbourhood search (Ropke and Pisinger, 2006) for the routing problem and a tree search algorithm (Bortfeldt, 2012) for loading the boxes. They concluded that there was a clear trade off between travel distance and reloading effort.

In the integration of the routing and container loading problem, it is possible to consider every practical constraint for the cargo (e.g. stability and weight distribution). For a review on the integration routing problems and container loading problem with practical constraints the reader is referred to Pollaris *et al.* (2015).

2.2.4 Benchmarks

Some of the papers cited in subsections 2.2.1 and 2.2.2 tested their solution approaches in their own artificial generated data, others in data based on real-world cases, but a considerable number used literature benchmarks. There is not a unique standardised benchmark for container loading problems. Below, we compile the data sets which are usually considered for comparison of solution strategies. Also, when some practical constraints are introduced, it is common that authors modify one of the benchmarks presented here to suit their needs. The benchmarks are:

- (Ivancic, Mathur and Mohanty, 1989): Commonly referred to as IMM instances, this benchmark consists of 47 instances and is the most used test bed in the papers mentioned in this chapter. Some authors considered this group of instances to have a small number of box types and boxes per type, thus the number of loading patterns is also small. In the benchmark, there is only one type of container.
- (Loh and Nee, 1992): Commonly referred to as LN instances, this benchmark contains 15 instances. In each instance, the container has a different size and its volume is large enough to pack all items in 11 of the 15 cases. There is from 100 to 250 available items and from 6 to 10 different types of boxes.
- (Mohanty, Mathur and Ivancic, 1994): This benchmark consists of 16 randomly generated problems. All instances have at most three types of boxes and three types of containers, the number of boxes per type vary from 12 to 80 and the number of containers per type vary from one to five. Also, this benchmark specifies the value of boxes, besides their dimensions. We call this MMI benchmark.
- (Bischoff and Ratcliff, 1995): Commonly referred to as BR (BR1-7) instances, this benchmark consists of 700 instances for single container loading problems. The benchmark is grouped by the number of types of boxes (3, 5, 8, 10, 12, 15 and 20) and each group consists of 100 instances. The container has the same dimension in all instances.

- (Davies and Bischoff, 1999): This benchmark is sometimes considered complementary of (Bischoff and Ratcliff, 1995). It is consisted of 900 stronger heterogeneous instances, with up to 100 types of boxes per instances. It is usual to see the benchmarks from Bischoff and Ratcliff (1995) and Davies and Bischoff (1999) to be referred as BR (BR0 and BR8-15).
- (Ren, Tian and Sawaragi, 2011): This is a set of instances based on (Loh and Nee, 1992) and (Bischoff and Ratcliff, 1995) benchmarks and it includes information on shipment priorities. We call this benchmark RTS.
- (Che *et al.*, 2011): This benchmark consists of 350 instances for the multiple container cost minimisation problems. It is grouped by the number of types of boxes (from three to nine), each group having 50 instances and there are three different types of containers. We call this benchmark CHLZ.

More than 60% of the papers in this section used at least one of the reported benchmarks and most of the others used instances based on real-world cases. Artificially (or randomly) generated benchmarks were more common in early studies. Table 3 highlights some high performance solution procedures found in the literature for each of the benchmark datasets in both supported and unsupported cases. We made an effort to point just one algorithm per case and this indication is based on the average value of the objective functions for each group of instances considered. Therefore, the approach indicated in the table might not completely dominate others in the literature. Also, we focused only on supported and unsupported cases. The authors may have considered other practical constraints.

Table 3 – Papers with high performing results on average for each of the benchmarks considered. The dash (“-”) sign indicates we have not found a paper reporting results on that benchmark. Next to each reference, we report the methodology used in the paper with the following coding. **TS**: Tree search, **H**: Constructive heuristic, **MIP**: MIP-based approach, **MH**: Meta-heuristic and **CG**: Column generation.

Benchmark	Unsupported	Supported
IMM	(Zhu and Lim, 2012)/TS	(Zhu and Lim, 2012)/TS
LN	(Loh and Nee, 1992) ²	(Ren, Tian and Sawaragi, 2011)/TS
MMI	(Eley, 2003)/H+MIP	(Takahara, 2008)/MH
BR1-7	(Araya, Guerrero and Nuñez, 2017)/TS	(Araya, Guerrero and Nuñez, 2017)/TS
BR0, BR18-15	(Araya, Guerrero and Nuñez, 2017)/TS	(Ren, Tian and Sawaragi, 2011)/TS
RTS	-	(Wang, Lim and Zhu, 2013)/TS
CHLZ	(Che <i>et al.</i> , 2011)/CG	-

2.2.5 Summary

This subsection summarises the information gathered with the literature review. Particularly, we present an overview of solution approaches and the frequency of practical

² We could not find the original paper, the results reported here are available at <<https://www.ibr.cs.tu-bs.de/alg/packlib/article/ln-pahb-92.shtml>>

constraints in the literature of container loading problem. Our goal is to provide a broad view of the focus of research effort along the past decades. Table 4 presents features and solution approaches for selected papers.

In general, in our review, the most common solution approaches (around half of the references) for container loading problems are constructive heuristics and meta-heuristics. Also, roughly 20% of papers proposed solutions based on tree search. Less often, column generation techniques and papers dealing exclusively with mixed integer programming models could be found. It was common to see different combinations of practical constraints being considered (vertical and horizontal stability, loading bearing, shipment priority and others). Orientation and stability (vertical or horizontal) were the most frequent in our review. The same was noted by [Bortfeldt and Wäscher \(2013\)](#).

Roughly 80% of the papers in our review considered at least one practical constraint. Around one-third of the papers consider one practical constraint (usually the orientation constraint), but there are also papers dealing with more than four constraints. Noteworthy is [Chan *et al.* \(2006\)](#) which dealt with weight limit, weight distribution, orientation, allocation, positioning and stability constraints to develop a decision support system for loading cargo on planes. According to the authors, the system was able to provide solutions for complex industrial datasets with more than 90% volume utilisation.

Container loading problems have been studied by decades and it is possible to find sophisticated methods to tackle different variations of the problem. Particularly, there are variations related to increasing its applicability related to cargo stability, road transportation regulations, restriction of box positioning and quality of service. Nevertheless, we could not find references that handled container loading in environments with limited storage nor managing the demand for a fast loading process. These cases arise, for example, when the cross-docking distribution strategy is applied. In cross-docks, there is a very limited staging to serve as a buffer for the goods and there is a pressure to minimise the operation makespan. This has motivated us to explore container loading in cross-docking scenarios.

Table 4 – Summary of features and solution approach (methodology) for selected papers related to container loading literature. In the table we use the following coding. **WL**: Weight limit, **WD**: Weight distribution, **Or**: Orientation, **St**: Stacking, **CS**: Complete shipment, **Al**: Allocation, **Po**: Positioning, **Sb**: Stability and **Co**: Complexity.

Paper	WL	WD	LP	Or	St	CS	Al	Po	Sb	Co	Type (Wäscher2007)	Methodology
(George and Robinson, 1980)	0	0	0	0	0	0	0	0	0	0	SLOOP	Constructive heuristic
(Morabito and Arenales, 1994)	0	0	0	1	0	0	0	0	0	0	SLOOP	AND/OR graph
(Mohanty, Mathur and Ivancic, 1994)	0	0	0	1	0	0	0	0	1	1	MHLOPP	Math-heuristic
(Chen, Lee and Shen, 1995)	0	1	0	1	0	0	0	0	0	0	All IPT	MIP model
(Davies and Bischoff, 1999)	0	1	0	1	0	0	0	0	1	0	SLOPP	Improving heuristic
(Martello, Pisinger and Vigo, 2000)	0	0	0	0	0	0	0	0	0	0	SBSBPP	Tree search
(Pisinger, 2002)	0	1	0	1	0	0	0	0	1	0	SKP	Tree search
(Eley, 2002)	0	0	0	0	0	0	0	0	0	0	SLOPP, SSSCSP	Constructive heuristic and tree search
(Eley, 2003)	0	0	0	0	0	1	1	0	0	0	MSSCSP, MHLOPP	Constructive heuristic and MIP model
(Brunetta and Grégoire, 2005)	0	0	0	0	0	0	0	0	0	0	MSSCSP, MBSBPP	Tree search
(Moura and Oliveira, 2005)	0	0	0	1	0	0	0	0	0	0	SLOPP	Meta-heuristic
(Lim and Zhang, 2005)	0	0	0	0	0	0	0	0	0	0	SKP, SBSBPP	Constructive heuristic and tree search
(Takahara and Miyamoto, 2005)	0	0	0	1	0	0	0	0	1	0	MHLOPP	Meta-heuristic
(Takahara, 2008)	0	0	0	0	0	0	0	0	1	0	MILOPP, MHLOPP	Meta-heuristic
(Fanslau and Bortfeldt, 2010)	0	0	0	1	0	0	0	0	1	1	SKP, SLOPP	Tree search
(Ren, Tian and Sawaragi, 2011)	0	0	0	1	0	0	0	0	0	0	SKP, SLOPP	Tree search
(Che <i>et al.</i> , 2011)	0	0	1	1	0	0	0	0	1	0	SSSCSP, MSSCSP	Column generation
(Zhu and Lim, 2012)	0	0	0	0	0	0	0	0	1	0	SKP, SLOPP	Constructive heuristic and tree search
(Zhu, Huang and Lim, 2012)	0	0	0	1	1	0	0	0	1	0	SSSCSP, MSSCSP	Column generation
(Gonçalves and Resende, 2012)	0	0	0	1	0	0	0	0	1	0	SKP, SLOPP	Meta-heuristic
(Junqueira <i>et al.</i> , 2013)	0	0	0	1	1	0	0	1	1	1	SKP, SLOPP	MIP models
(Ceschia and Schaefer, 2013)	1	0	0	1	1	0	0	1	0	0	MHKP, MBSBPP	Meta-heuristic
(Wang, Lim and Zhu, 2013)	0	0	1	1	0	0	0	0	1	0	SKP, SLOPP	Tree search
(Lim <i>et al.</i> , 2013)	1	1	1	1	0	0	0	0	1	0	SKP, SLOPP	Meta-heuristic
(Domingo, Ponnambalam and Kanagaraj, 2013)	0	0	0	1	0	0	0	0	0	0	SKP, SLOPP	Meta-heuristic
(Sheng <i>et al.</i> , 2013)	0	0	0	1	0	0	0	0	1	1	SKP, SLOPP	Tree search
(Araya and Riff, 2014)	0	0	0	1	0	0	0	0	1	0	SKP, SLOPP	Tree search
(Zheng, Chien and Gen, 2015)	0	0	0	1	0	0	0	0	0	0	SKP, SLOPP	Meta-heuristic
(Wei, Zhu and Lim, 2015)	0	0	0	1	0	0	0	0	1	0	MSSCSP, MBSBPP	Column generation
(Tian <i>et al.</i> , 2016)	0	0	1	1	0	0	0	0	0	0	MSSCSP, MBSBPP	Meta-heuristic
(Jamrus and Chien, 2016)	0	0	1	1	1	0	0	0	1	0	SKP, SLOPP	Meta-heuristic
(Sheng <i>et al.</i> , 2016)	0	0	1	0	0	0	0	1	1	0	SKP, SLOPP	Tree search
(Paquay <i>et al.</i> , 2018a)	1	0	0	1	1	0	0	0	1	0	MBSBPP	Constructive heuristic
(Correcher <i>et al.</i> , 2017)	1	0	0	1	1	0	0	0	1	1	MSSCSP	Meta-heuristic
(Araya, Guerrero and Nuñez, 2017)	0	0	0	1	0	0	0	1	0	0	SKP, SLOPP	Tree search

2.3 Network scheduling and collaborative freight distribution

Container loading problems may be considered the first step in consolidation for transportation. But then, the goods must be distributed by a freight distribution system. In this section, we review the literature in network scheduling problems, focusing on networks which have some kind of consolidation centre (or cross-dock), see Subsection 2.3.1. Also, we review the literature regarding collaborative freight distribution (Subsection 2.3.2).

2.3.1 Network scheduling with consolidation centres

Logistic distribution problems, in the optimisation context, often have an associated network. Some characteristics of this network may be used to classify the problems. In particular, we are interested in closed and open networks. In closed networks, the distribution routes have the same starting and end point (e.g. the traditional *vehicle routing problem*). Also, we can have open networks in the sense that routes have different starting and endpoint. Besides, in both cases, the goal is related to the efficient scheduling the trucks, namely, the order in which the nodes in the network are visited. To the best of our knowledge, the open network case remains rather unexplored. Therefore, in this section, we focus on the literature regarding to closed and network scheduling in the context of cross-docking.

Lee, Jung and Lee (2006) were one of the firsts to consider a network scheduling problem contemplating cross-docks. The authors proposed a mixed integer programming model to perform pickup and delivery operation for a cross-docking in which the vehicles must arrive together at the facility. Using instances solved by complete enumeration, the experiments showed that a proposed tabu search could obtain solutions with an average deviation of 4%. The same scenario was approached with a hybrid meta-heuristics by Vahdani *et al.* (2012). Combining elements from particle swarm optimisation, simulated annealing and variable neighbourhood, yielded a better than a tabu search algorithm on randomly generated instances. A pickup and delivery operation with cross-docking was also integrated with the pollution-routing problem (Abad *et al.*, 2018). In this integration, besides routing and consolidation decisions the fuel consumption is considered as well. Abad *et al.* (2018) proposed a model and three multi-objective meta-heuristics to solve it. The experiments investigated whether using cross-dock was a good alternative to a traditional distribution system. The results supported that although, for some cases, using cross-docks lead to a later delivery, the system could provide more accurate delivery times.

Wen *et al.* (2009) defined the *vehicle routing problem with cross-docking* (VRPCD). The main difference from the problem studied by Lee, Jung and Lee (2006) and Vahdani *et al.* (2012) is that the synchronisation was achieved by consolidation decisions rather than time constraints. The authors proposed a mixed integer linear model and were

able to obtain high-quality solution for instances based on real data using a tabu search within and adaptative memory procedure, which was outperformed by iterated local search procedures by [Morais, Mateus and Noronha \(2014\)](#). Later, better solutions were achieved by a math-heuristic based on large neighbourhood search proposed by [Grangier et al. \(2017\)](#).

Extensions on the VRPCD were proposed by [Moghadam, Ghomi and Karimi \(2014\)](#) and [Maknoon and Laporte \(2017\)](#). [Moghadam, Ghomi and Karimi \(2014\)](#) considered a pickup and delivery operation allowing for split deliveries. The authors proposed a mixed integer non-linear programming model and meta-heuristics based on simulated annealing and ant colony optimisation. Combining simulated annealing and ant colony optimisation, in a hybrid approach, presented the best results. [Maknoon and Laporte \(2017\)](#) proposed an extension of the VRPCD contemplating networks with multiple cross-docks. The authors presented a mixed integer linear formulation and an adaptative large neighbourhood search which outperformed CPLEX solver for larger instances.

[Dondo, Méndez and Cerdá \(2011\)](#) proposed the Multi-echelon Vehicle Routing Problem with Cross-docking in Supply Chain Management (VRPCD-SCM). The authors considered a multi-echelon network with suppliers, cross-docks and destinations in one of the few papers considering a heterogeneous fleet. In each echelon, only specific types of vehicle might transit and each vehicle may have a different volume and weight capacity. A mixed integer linear programming model is proposed and is evaluated in five study cases that were solved to optimality in reasonable computational time.

A scenario with multiple cross-docks was also explored by [Chen et al. \(2016\)](#). The authors presented a model for minimising total costs in pickup and delivery operations originating at multiple cross-docks. Also, results showed that a solution approach based on particle swarm optimisation outperformed a genetic algorithm in different test cases. [Ahkamiraad and Wang \(2018\)](#) considered multiple cross-docks, with homogeneous fleets and limited capacity and pickup and delivery nodes with time windows. Their solution approach, a hybrid of genetic algorithm and particle swarm optimisation, outperformed CPLEX for medium and large problem sizes.

[Santos, Mateus and Cunha \(2013\)](#) explored a variant of the vehicle routing problem with cross-docking in which vehicles do not have to stop at the consolidation centre if it does not yield cost reductions. The authors proposed a branch and price algorithms and experiments showed that routes not using the cross-dock appeared in the solution even if (un)loading costs are high relative to routing costs. [Hu, Zhao and Choi \(2013\)](#) explored another variant in which the minimisation of waiting times of trucks in the cross-docks is one of the objectives, besides minimising the total cost of routing inbound and outbound trucks and the size of the fleet. The authors proposed a mixed integer linear programming model which was able to handle instances with limited size.

[Ahmadizar, Zeynivand and Arkat \(2015\)](#) formulated a multi-echelon routing prob-

lem with cross-docking. In the first layer, suppliers for specific demands are assigned to cross-docks and, in the second layer, the goods from these suppliers are distributed. A pickup and delivery problem must be solved at each layer, minding the synchronisation of trucks in the cross-docks to allow for consolidation. The authors proposed a mixed integer non-linear model for minimising total distribution costs. The model was challenging to solve even for small instances, thus a genetic algorithm was proposed which performed better.

Grangier *et al.* (2016) also explored a multi-echelon scenario with pickup and delivery operations. All vehicles start at a cross-dock and must perform pickups, return to the cross-dock for moving the cargo to more appropriate vehicles and then perform the deliveries. Delivery vehicles are allowed to multi-trip and must share some time in the cross-dock with the corresponding pickup vehicle. The authors reported that, although similar scenarios were already described, they were not aware of any implementation. They presented a mixed integer linear formulation and an adaptative large neighbourhood heuristic. Results showed that the heuristics could find a good solution in a reasonable time and concluded that the algorithm could be applied to other routing problems with synchronisation. A similar problem in a multi-objective context was explored by Mohtashami *et al.* (2015). The authors proposed a model for the minimisation of the makespan, transportation cost and truck trips. Two algorithms were proposed to solve the problem (based on the non-dominated sorting genetic algorithm (NSGA-II) and the multi-objective particle swarm optimisation (MOPSO), respectively). Evaluating the algorithms using different metrics, the authors showed that there was not a clear dominance of one over the other across all criteria. More recently, Baniamerian, Bashiri and Tavakkoli-Moghaddam (2019) defined a similar problem, with a different objective function. Arguing that, due to resource limitation, all demands may not be satisfied, managers could plan to maximise profit. The authors proposed a mixed integer linear programming model and a modified variable search algorithm combined with a genetic algorithm for the problem. The experiments showed that the hybrid algorithm could achieve optimality for small test cases and needed less computational time for bigger instances when compared to an artificial bee colony and simulated annealing algorithms.

Enderer, Contardo and Contreras (2017) proposed the Dock-Door Assignment and Vehicle Routing Problem, which integrates decisions from inbound and outbound door assignments and routing of the outbound fleet. Focusing on a single facility, the authors considered the routing costs between each pair of inbound-outbound doors and to distribute the goods for consumers. Each consumer had a demand for multiple commodities that must be routed through a cross-dock. Two mixed integer linear programming models were formulated and one was used in a column generation framework for providing lower bound. Furthermore, a heuristic which used information from the column generation was able to obtain good solutions. The results showed that integrating dock assignment and routing decision could lead to significant savings.

Geoffrion and Graves (1974) were one of the firsts to study an open network setting with distribution centres. Based on a multi-commodity flow problem, the authors formulated a mixed integer linear programming model for the selection of distribution centres to be used as intermediate facilities between suppliers and consumers. Each consumer was only served by one distribution centre which had a limited capacity. For the solution of the model, a Benders decomposition was used. The results showed an unexpected small number of Benders' cuts were needed to achieve an optimal solution.

An early model for the open network scheduling problem was proposed by Chen *et al.* (2006). Although similar to the minimum-cost multi-commodity flow problem (MCMCFP), in the version explored by Chen *et al.* (2006) there was no explicit source and sink pairs, split deliveries were not allowed. The authors considered time windows for consumers and multiple cross-docks. A model and a heuristic combining local search, simulated annealing and tabu search are proposed. The hybrid approach outperformed CPLEX solver in solution quality and computational time. Yu, Jewpanya and Redi (2016) studied the *open vehicle routing problem with cross-docking*, which is related to the problem studied by Chen *et al.* (2006) but without time windows and with only one cross-dock. A simulated annealing procedure could obtain solutions similar to CPLEX but faster, for small and medium size instances.

Hernández, Peeta and Kalafatas (2011) also study the open network case. In their scenario, there were multiple carries which needed to move goods from origin to destination. Contemplating the possibility of collaboration among the carries, the authors proposed a model for identifying a set of routes which minimise the total transportation cost.

In general, to the best of our knowledge, the open network scheduling remains a rather unexplored case in the vehicle routing literature and particularly in the networks with cross-docking. In Chapter 4, we explore an open network scheduling problem with cross-docking and, in Chapter 5 a closed network scheduling problem with collaborative facilities is proposed.

2.3.2 Collaborative freight distribution

Due to the highly competitive nature of the transportation industry (Gansterer and Hartl, 2018), some advocate that the only way to further increase efficiency is through collaboration among different business. Specially in urban environments, besides reducing costs, collaboration can reduce congestion, pollution and noise (Cleophas *et al.*, 2019). This section gives an overview on different types of collaboration before focusing on collaborative vehicle routing problems (Subsection 2.3.2.1).

Here, we consider collaboration as sharing of information, infrastructure or resources among business, which may lead to benefits to the overall system or to individual participants of the collaborative coalition. Some authors differentiate between *collaboration* and *cooperation*. In the literature, there is an agreement that collaboration is a strong type of

cooperation although the boundaries may be fuzzy (Cruijssen, Dullaert and Fleuren, 2007). Therefore, we use *collaboration* and *cooperation* interchangeably.

There are different ways in which participants might collaborate. They can be classified as *centralised* or *decentralised* collaboration (Gansterer and Hartl, 2018). In a centralised setting, there is a central authority that has all the information about the orders (requests by consumers), infrastructure and capabilities of each member of the coalition. Thereby, the central authority can optimise the overall system performance and distribute the orders for the participants accordingly. In a decentralised system, there is no central authority and the members of the coalition are not required to share all information. Then, all the members of the coalition must agree with a collaboration policy and the orders are usually exchanged between members when they identify win-win situations. Moreover, each member should decide on which members to collaborate with, requests that will be offered to others and mechanisms to exchange requests. The exchange of requests can also be done using auctions, where collaborators submit their orders to a common pool, allowing others to bid for it. Figure 7 visually represents some differences between centralised and decentralised collaborative systems.

Figure 7 – Schematic representation of centralised (a) and decentralised collaboration (b) systems.

Besides the *centralised-decentralised* classification, we can also identify *vertical* and *horizontal* collaboration. In vertical collaborative transport, partners are often operating different services or modes of transport (Cleophas *et al.*, 2019) or there is among a hierarchical relationship among them (e.g. subcontracting). In a city logistics context, for example, delivering the goods to consolidation centres in the outskirts of the city might be done by long-haul trucks, whereas the city distribution might be done by small, environmentally friendly vehicles. In horizontal collaboration, on the other hand, partners work in the same section of transport, potentially sharing fleet, facilities and other infrastructure as well as orders. Particularly in the horizontal collaboration, the sharing of benefits and costs also need to be considered. For a review on this allocation of cost and benefit, the reader is referred to Guajardo and Rönnqvist (2016).

Due to the nature of our work, we focused our review on the collaborative routing distribution systems found within the optimisation literature. For broader reviews, the reader is referred to Cruijssen, Dullaert and Fleuren (2007), Gansterer and Hartl (2018)

and Cleophas *et al.* (2019). Cruijssen, Dullaert and Fleuren (2007) elaborated on the type of horizontal relationships among competitors and partners in a general market and then explored the general concepts in transport and logistics. Gansterer and Hartl (2018) focused on collaborative vehicle routing and review the literature considering both the *centralised-decentralised* and the *vertical–horizontal* classification. Cleophas *et al.* (2019) explored recent advances in theory and practice related to urban collaborative freight distribution.

2.3.2.1 Collaborative vehicle routing

Chu (2005) introduced what became known as *vehicle routing problem with private fleet and common carrier* (VRPPC). The VRPPC consists of routing a fixed number of trucks with limited capacity and select less-than-truckload carriers to minimise the total cost of satisfying the demands. The authors proposed an integer linear programming model and modified the Clarke and Wright heuristic for the vehicle routing problem (Clarke and Wright, 1964) to the VRPPC. The authors argued that the heuristic could obtain optimal or near-optimal solutions in reasonable computational time. These results were consistently and successively improved by the heuristics from Bolduc, Renaud and Boctor (2007), Bolduc *et al.* (2008) and Côté and Potvin (2009) respectively, although some with higher computational time. In general, the VRPPC was mostly explored using heuristics (e.g. Potvin and Naud (2011), Stenger *et al.* (2013) and Vidal *et al.* (2015)). The most recent work that we are aware of being the biased random-key genetic algorithm by Higino and Chaves (2018), which could achieve average results with less than 1% deviation from best known solutions.

Similar to the VRPPC is the *task selection and routing problem in collaborative transportation* proposed by Liu *et al.* (2010). In this problem, a carrier, operating in full-truckload, receives orders and need to decide whether to realise them with a private vehicle or subcontract another carrier. The authors proposed a mixed integer linear programming model for minimising the variable and fixed costs to operate the private fleet and the total costs for the subcontracts. Also, a memetic algorithm was used to tackle large-scale instances. The results indicated that the memetic algorithm outperformed a genetic algorithm regarding solution quality.

Özener, Ergun and Savelsbergh (2011) consider exchange of orders in full-truckload shipments. They proposed an integer programming model to identify the highest potential savings in collaboration and then devise different mechanisms for the exchange of orders. The mechanisms differ, for example, in the amount of information shared among the collaborators. The experiments showed that the cost savings are potentially higher when sharing more information. Also, some mechanisms could achieve around 95% of the potential savings.

Wang, Kopfer and Gendreau (2014) combined the *integrated operation transporta-*

tion planning (IOTP) (Krajewska and Kopfer, 2009) and the *collaborative transportation planning* (CTP) (Wang and Kopfer, 2014). The IOTP is related to carriers that carefully assign the most profitable orders to their fleet and subcontract other carriers for the remaining orders. Note that the IOTP is a form of vertical collaboration. The CTP is defined as a joint decision making to satisfy transportation orders among partners in a horizontal collaboration scheme. Wang, Kopfer and Gendreau (2014) extend the pickup and delivery problem with time windows to accommodate the decisions related to IOTP and CTP. The results showed that adding the exchange of requests to the subcontracting strategy can reduce costs by more than 10% on average.

Chinh *et al.* (2016) compared two strategies of collaboration for two suppliers to distribute their goods. In the first one, a freely collaborative distribution of orders is allowed. However, the authors argued that companies might resist sharing complete information with others and the difference in business processes could make the collaboration infeasible. Therefore, in the second strategy, each supplier prioritises its orders before collaborating. Compared to a non-collaborative setting, the experiments showed a cost reduction of 41% in the first strategy and of 17% in the second strategy.

Zibaei, Hafezalkotob and Ghashami (2016) use cooperative game theoretic approaches to form a coalition in multiple depot vehicle routing problems. The authors solved to the model considering no coalitions are present, then, they solved the model again considering coalitions of two players, then coalitions of three players and so on. Results showed that the cost savings in distribution could reach almost 50%. Kaewpuang *et al.* (2017) also explored coalition formation. The authors considered shippers that operate with full-truckload (FTL) and less-than-truckload (LTL). They proposed a framework that manages vehicle allocation and cost sharing. The authors formulated optimisation models to allocate vehicles shared by shippers in the coalition to minimise delivery cost. A mixed integer linear programming model is used in a deterministic case and, when customer demands follow a probability distribution, the authors use a stochastic programming model. The delivery costs are based on the contribution of each shipper to the vehicle pool. Moreover, they proposed a coalition formation game to decide whether and with how many vehicles each shipper should contribute to the the pool. The experiments showed a reduction in the number of vehicles and cost of delivery when shippers cooperate.

Defryn, Sörensen and Cornelissens (2016) considered a vehicle routing problem in which the number of vehicles and maximum distance of a vehicle are bounded, therefore, generally, it is not possible to visit all consumers. The consumers which are not visited receive compensation. In their collaborative environment, companies share their vehicles to increase the level of services and minimise the costs of distribution. They performed experiments with different cost allocation strategies and showed that, with the right incentive, companies might behave in a way that benefits the coalition.

Zissis, Aktas and Bourlakis (2018) used real data from retailers and consumers to

evaluate the potential economic, environmental and social costs of possible collaborations. The authors used the same vehicle routing model and solved it independently for each retailer and for all the requests of all retailers to estimate the potential benefits. They analysed the impact of these solutions according to 12 performance metrics. The results suggested that gains with collaboration depend on distances between consumers and the number of orders.

Most of the works we could find in the optimisation literature regarding collaborative freight distribution focus on how to share orders and/or costs. In some cases, shared facilities (e.g. consolidation centres) have some public subsidies. However, the collaborative initiative might fail if the subsidies are removed. This is a gap we explored in Chapter 5 by deciding how to use the currently available infrastructure to promote the collaboration among the suppliers.

CONTAINER LOADING WITH TIME AVAILABILITY CONSTRAINTS

From a city logistic perspective, moving goods in urban regions should consider the impact on traffic and noise and air pollution, mainly if large vehicles are used. One strategy to reduce the size of vehicles needed is to use Urban Consolidation Centres, which operate as cross-docks. These cross-docks are ideally able to receive, sort and ship goods using more appropriate vehicles in a matter of hours. In this chapter, we investigate, from an optimisation point of view, the loading process at an outbound dock of a cross-dock. The reader is referred to Figure 2 (Chapter 1) for a schematic representation of a cross-dock.

In the previous chapter, we identified that the literature on cross-docks usually simplifies the packing problems to a single dimension (Section 2.1) and focus on operational scheduling decisions. Three dimensions in packing are considered by authors working on Container Loading Problems (Section 2.2) but they, in turn, tend to ignore the operational decisions that take place when the packing occurs within a dynamic environment such as a cross-dock. In this chapter, we focus on the intersection between these two literatures and study the Container Loading Problem with time Availability Constraints (CLPTAC). The main idea here is to consider the Container Loading Problem in which the load to be packed becomes available dynamically, simulating the situation of a cross-dock. In sections 3.1 and 3.2, we present an output maximisation version (CLPTAC-Om, maximising the volume loaded into a limited number of containers) and an input minimisation version of the problem (CLPTAC-Im, minimising the cost of containers used to load all boxes), respectively. Section 3.3 presents some concluding remarks for the chapter.

In both problems, CLPTAC-Om and CLPTAC-Im, the idea is to solve a container loading problem without totally disregarding cross-docking efficiency (i.e., perform the load activities as fast as possible). This naturally leads to multi-objective problems. Since our goal is to gain insight in this rather unexplored scenario, we approached the multi-objective aspect simply by considering a weighted sum of the objectives, although other approaches

could be equally (or more) suitable for the problem, depending on the practical interests of the decision maker. Moreover, for both problems, we focused on designing approaches which could rely on the extensive knowledge of traditional container loading problems. For this, we proposed a dynamic programming framework and a constructive heuristic for CLPTAC-Om and CLPTAC-Im, respectively.

3.1 An output maximisation problem with time availability constraints

As seen in Section 2.1, cross-dock's efficiency is usually measured in terms of the process makespan, the total operation time. Container Loading Problems, in turn (see Section 2.2) are usually evaluated in terms of the volume utilisation of the containers. In an integrated problem such as the CLPTAC proposed here, both metrics are relevant. They are also conflicting, as the more we wait (worse makespan) the more boxes are available and potentially a greater volume can be loaded into a container/truck (better volume utilisation).

In this subsection, we define an output maximisation version of a container loading problem with time availability constraints (CLPTAC-Om). The objective is to position a subset of boxes inside of a container maximising the loaded volume and minimising the ready time of the container (i.e. the time the last box is loaded). The boxes must not overlap with each other, must be fully inside the container and cannot be loaded before they are available. Also, we assume orthogonal packing (boxes and containers faces must be parallel to each other). We define deterministic and stochastic versions of the problem, with respect to the available knowledge on the arrival times of the boxes.

Table 5 – Summary of parameters for CLPTAC-Om.

$\mathcal{B}$:	Set of available boxes.
(l_{i1}, l_{i2}, l_{i3}) :	Length of box $i \in \mathcal{B}$ along axis X , Y and Z , respectively.
(L_1, L_2, L_3) :	Length of the container along axis X , Y and Z , respectively.
t_i :	Time from which box $i \in \mathcal{B}$ becomes available for loading.
T :	Length of the time horizon.

More formally, assume a finite time horizon $[0, T]$. Let $\mathcal{B}$ be a set of boxes, each box $i \in \mathcal{B}$ is defined by its length along each axis (l_{i1}, l_{i2}, l_{i3}) and the time it becomes available for loading t_i . Simulating the arrival process of the boxes, we assume that, once a box becomes available, it never becomes unavailable. We have a single container, which is a three-dimensional rectangle and has dimensions (L_1, L_2, L_3) . Also, we consider that every $i \in \mathcal{B}$ fits inside the container, otherwise, we could remove the ones that do not fit with a pre-processing step. For the deterministic version, the arrival time of boxes $t_i \in [0, T]$, $i \in \mathcal{B}$ is defined *a priori*. For the stochastic version, only the probability distribution of t_i is known. This section focus on the stochastic version, although the same solution

method could be applied to the deterministic version. The parameters of the problem are summarised in Table 5.

3.1.1 A dynamic programming algorithm for CLPTAC-0m

We develop a stochastic dynamic programming algorithm which decomposes the temporal and geometric aspects of the problem. This enables us to use any approach for the traditional container loading problems to compute the costs of the states in the dynamic programming algorithm. For this, we uniformly discretised the time horizon $[0, T]$ into time periods $\mathcal{T}$ and defined the sets $\mathcal{B}_t$, $t \in \mathcal{T}$, to be the set of boxes (shipment) arriving at time period $t \in \mathcal{T}$.

The dynamic programming approach is formulated as an optimum stopping problem (Peskir and Shiryaev, 2006) in which one needs to decide when to stop a process and there is usually a trade-off between stopping earlier and having a better container occupation. In our context, we want to decide when to stop waiting for new shipments ($\mathcal{B}_t$, $t \in \mathcal{T}$) and start loading the boxes into the container. The trade-off is: the longer we wait, the more boxes we have and potentially a better volume utilisation; on the other hand, the sooner we stop waiting (and perform the loading) the smaller is the ready time.

Table 6 – Summary of notation used in the formulation related to the dynamic programming.

$\mathcal{T}$:	Set of time periods in the time horizon.
$\mathcal{T}^*$:	Set of all time periods but the last in the time horizon.
s_t :	Set of possible decisions at time period $t \in \mathcal{T}$: load (0) and wait (1).
$\mathcal{X}_t$:	State of the system (set of boxes available) at time period $t \in \mathcal{T}$.
$\mathcal{C}(\mathcal{X}_t, s_t)$:	Cost of making decision s_t in state $\mathcal{X}_t$, $t \in \mathcal{T}$.
$\mathcal{E}(\mathcal{X}_t)$:	Solution value of traditional container loading problem with boxes $\mathcal{X}_t$, $t \in \mathcal{T}$.

Let the state of the system be the set of boxes $\mathcal{X}_t$ available at period $t \in \mathcal{T}$ and $\mathcal{C}(\mathcal{X}_t, s_t)$ be the cost of decision $s_t \in \{0, 1\}$, $t \in \mathcal{T}$, in state $\mathcal{X}_t$. The possible decisions are to perform the loading or to wait for more boxes, henceforth coded as 0 and 1, respectively. Also, let $\mathcal{E}(\mathcal{X}_t)$ be the solution value of a traditional container loading problem with boxes in $\mathcal{X}_t$, namely, a solution that maximises the volume loaded with boxes represented by $\mathcal{X}_t$, without any temporal constraints. Moreover, let μ be a parameter indicating the cost of empty volume in the container. Thereby, with $\mathcal{T}^*$ being the set of all time periods but the last, the cost functions of the dynamic programming model are given by (3.1)–(3.4). The notation used is summarised in Table 6.

$$\mathcal{C}(\mathcal{X}_t, 0) = \frac{1}{|\mathcal{T}|} + \mu \left(1 - \frac{\mathcal{E}(\mathcal{X}_t)}{L_1 L_2 L_3} \right), \quad t \in \mathcal{T}^*, \mathcal{X}_t \neq \emptyset, \quad (3.1)$$

$$\mathcal{C}(\mathcal{X}_t, 1) = \frac{1}{|\mathcal{T}|}, \quad t \in \mathcal{T}^*. \quad (3.2)$$

$$\mathcal{C}(\mathcal{X}_{|\mathcal{T}|}) = \frac{1}{|\mathcal{T}|} + \mu \left(1 - \frac{\mathcal{E}(\mathcal{X}_{|\mathcal{T}|})}{L_1 L_2 L_3} \right), \quad (3.3)$$

$$\mathcal{C}(\emptyset, 0) = \mathcal{C}(\emptyset) = 0, \quad t \in \mathcal{T}^*. \quad (3.4)$$

Equations (3.1) define the cost of the decision to stop the waiting process and proceed with the loading with available boxes $\mathcal{X}_t$ at time $t \in \mathcal{T}^*$. If the decision is to wait for more boxes, the respective partial cost is given by (3.2). The terminal cost, i.e., the cost of reaching the last period in the planning horizon is equivalent to choosing to perform the loading at the last period (3.3). If there are no boxes available for loading, $\mathcal{X}_t = \emptyset$, the costs are given by (3.4). We want to optimise the objective (3.5), in which $\mathbb{E}$ denotes the expected value due to the uncertainty in the arrival times of the boxes.

$$\text{Min } \mathbb{E} \left(\sum_{t \in \mathcal{T}^*} (C(\mathcal{X}_t, s_t)) + C(\mathcal{T}) \right). \quad (3.5)$$

Which is equivalent to minimising the sum of the makespan (in time periods) and the cost of the empty volume that it is possible to achieve with the boxes available until we decide to stop waiting, $s_t = 0$. Note that, if the decision in a particular period is to wait, $s_t = 1$, we sum the cost (regarding makespan) of that period, $\frac{1}{|\mathcal{T}|}$. Otherwise, $s_t = 0$, we still sum the cost related to the makespan (plus the cost of empty volume). In other words, we assume that the loading process only ends in the end of a period. We use $(|\mathcal{T}|)$ and $(L_1 L_2 L_3)$ as normalising factor for makespan and volume, respectively.

We may find an optimal solution for the stochastic case by using the value functions (3.6) and (3.7);.

$$\mathcal{V}_{|\mathcal{T}|}(\mathcal{X}_t) = \frac{1}{|\mathcal{T}|} + \mu \left(1 - \frac{\mathcal{E}(\mathcal{X}_t)}{L_1 L_2 L_3} \right), \quad \mathcal{X}_t \subseteq \bar{\mathcal{B}}_{|\mathcal{T}|}, \quad (3.6)$$

$$\mathcal{V}_t(\mathcal{X}_t) = \text{Min} \left\{ \frac{1}{|\mathcal{T}|} + \sum_{\bar{b} \subseteq \mathcal{X}_{t+1}} \rho_{t+1}(\bar{b} \mid \mathcal{X}_t) \mathcal{V}_{t+1}(\bar{b}), \frac{1}{|\mathcal{T}|} + \mu \left(1 - \frac{\mathcal{E}(\mathcal{X}_t)}{L_1 L_2 L_3} \right) \right\}, t \in \mathcal{T}^*, \mathcal{X}_t \subseteq \bar{\mathcal{B}}_t, \quad (3.7)$$

in which $\rho_t(\mathcal{B}_t)$ is the probability of shipment $\mathcal{B}_t$ arriving in time period t and $\bar{\mathcal{B}}_t$ is the set of all shipments $\{\mathcal{B}_1, \dots, \mathcal{B}_t\}$ that could be available at time t . Thus, the expected value of $\mathcal{V}_1(\mathcal{X}_1 = \mathcal{B}_1)$ gives an optimum solution to the stochastic version of the problem.

To compute $\rho_{t+1}(\bar{b} \mid \mathcal{X}_t)$ in (3.7), we need a probabilistic model for the arrival times of subsets $\mathcal{B}_t$. To illustrate our approach, we assumed the arrival time of subset $\mathcal{B}_{t_1}$ is independent of the arrival time of $\mathcal{B}_{t_2}$, $t_1 \neq t_2$, and the probability of a subset $\mathcal{B}_t$ arriving

at time $i \in \mathcal{T}$ is given by:

$$\rho_i(\mathcal{X}_t) = \begin{cases} \alpha_{it}(1 - \alpha_{it})^{i-t}, & \text{if } i \geq t, \\ 0 & \text{if } i < t, \end{cases} \quad (3.8)$$

in which $\alpha_{it} \in (0, 1]$, $i, t \in \mathcal{T}$, is a probability index associated with subset $\mathcal{B}_t$ arriving at time $i \in \mathcal{T}$, $i \geq t$. Within this framework, α_{it} , can be interpreted as a reliability profile for the company in charge of delivering the subset (cargo) $\mathcal{B}_t$. If $\alpha_{it} = 1$, $i, t \in \mathcal{T}$, we have the deterministic case, in which we known the arrival time of each box *a priori*. For simplicity, we used $\alpha_{it} = \alpha$, $i \in \mathcal{B}_t$, $t \in \mathcal{T}$, which can be viewed as a reliability index for the corresponding logistic environment. Figure 8 shows the probabilities for an example with reliability index $\alpha = 0.7$ for a set of boxes scheduled to arrive in time slot $t = 1$.

Figure 8 – Value of the probability of $\mathcal{B}_1$ arriving at $i \in \{1, \dots, 9\}$ for a reliability index of $\alpha = 0.7$.

Note that equations (3.6) and (3.7) are independent of the probabilistic model for the arrival of the boxes. Figure 9 exemplifies a scenario with $\mathcal{T} = \{1, 2, 3\}$. Each cell is a bit array ($\mathcal{B}_3\mathcal{B}_2\mathcal{B}_1$) representing the current state, i.e., set of boxes that have actually arrived at a point in time. The bit array $\bar{b}_1 = 001$, for example, means that shipment 1 has arrived while shipments 2 and 3 have not. In each time period, for each bit array with non-zero probability of occurrence, we need to evaluate the decisions of waiting and finishing the process (equation (3.7)). The full blue lines represent the computation of the sum in equation (3.7) for all $\rho_i(\mathcal{X}_3) \neq 0$.

3.1.2 Computational experiments for CLPTAC-Om

We performed computational experiments to evaluate the sensibility of the solutions to the reliability of the logistic environment (α) and also to the parameter related to the cost of empty volume (μ). The experiments used a set of instances which were generated

Figure 9 – Example of a scenario with $\mathcal{T} = \{1, 2, 3\}$. It is a graphical example of equation (3.7). The dashed red lines show possible next states for $\mathcal{X}_1$. The full blue lines show the states which need to be evaluated at state 010 at time $t = 2$.

based on instances for the traditional container loading problem. Moreover, we evaluate different combinations of the parameters (α and μ). For each combination, 5000 repetitions of the experiment were performed when the resulting problem is stochastic. We begin this section by presenting the instances and the computational environment for the experiments. Then, in Subsection 3.1.2.1, we describe and analyse the results.

The benchmark problems for the CLPTAC-Om were generated based on the 47 instances proposed by Ivancic, Mathur and Mohanty (1989) for the traditional container loading problem. In this benchmark, instances have from two to five types of boxes, the number of boxes varies from 47 to 181 and we consider only one container per instance – containers have different dimensions across instances. For each box, we generated a time from which it becomes available (if no delays occur), simulating the arrival process. This time was generated randomly, following a uniform probability distribution, and it is an integer number in $[1, |\mathcal{T}|]$, $|\mathcal{T}| = 10$. Therefore, we are assuming a planning horizon with ten blocks of 40 minutes in a eight-hour working day.

We experimented with different cost of empty spaces $\mu \in \{1, 2, 4\}$. Also, to solve the sub-problems, we used the container loading model defined in Chen, Lee and Shen (1995) without allowing the rotation of the boxes. For solving the model, we used Gurobi 7.0 in Ubuntu 12.04. Regarding hardware settings, we used four cores of an Intel® Xeon CPU E5-2620 at a frequency of 2 GHz each. For convenience, we denoted CLPTAC μ instances in which the cost of empty spaces is $\mu \in \{1, 2, 4\}$.

3.1.2.1 Results on the dynamic programming approach

Solving the stochastic model for an instance of the CLPTAC-Om may take an impractical amount of time. Each evaluation of function $\mathcal{V}_t(\cdot)$ needs to solve a deterministic version of the container loading problem (via $\mathcal{E}(\cdot)$), which for real-world cases can be impractical itself, using an exact method. Therefore, to validate our framework, we solved the sub-problem in which all items are available (with a time limit of 3600 seconds) and then reduced the time limit to five seconds to find a solution for each of the remaining sub-problems (we noted that the solver is able to find “good” solutions quickly and then struggles to prove optimality). The goal of these experiments is to evaluate the effect of cost of empty volume (μ) and reliability index (α) on total ready time and volume occupation. Then, with the optimal policy from equation (3.7), we simulated the arrival of boxes 5000 times for each $\alpha \in \{0.60, \dots, 0.9\}$.

The experiments revealed that, on average, operation time and volume occupation tend to improve as the reliability index increases. An increasing trend can be observed in both ready time and volume occupation (Figure 10). Regarding ready time, this trend can be explained due to a higher risk of waiting in a environment with lower reliability index. The same reasoning applies to the volume occupation, the higher the risk of waiting the lower the occupation achieved. For the decision maker, it suggests that the more reliable the logistic environment the better the expected vehicles occupation and the ready times.

Regarding the different values of μ , we performed a Wilcoxon signed-rank test (Woolson, 2007) for each α to compare the different values for μ . The test computes a p-value of less than 0.1% for any pair of μ values. In other words, we tested for statistical differences between every pair of μ (i.e., (1, 2), (1, 4), (2, 4)) and with probability more than 99.9% we can discard the null hypothesis both samples come from the same population.

Figure 10 – Profile of empty volume (dashed line) and operation time (full line) related to the reliability index (α) for $\mu \in \{1, 2, 4\}$. The values were averaged across all instances.

Also, the experiments revealed a linear trend specially for the operation time in relation to the reliability index. It can help managers in devising business plans to account for costs of operation, delivery times and value of possible penalties for delayed deliveries in seasonal variations of the reliability index. This, besides allowing a more efficient management of the facility, can potentially increase the service level for final customers.

3.1.3 Remarks on CLPTAC-Om

In this section, we proposed the *Output Maximisation Container Loading Problems with Time Availability Constraints* (CLPTAC-Om). Besides relaxing the hypothesis that each box is promptly available for loading, the goal is to minimise a weighted sum of the empty volume and ready time of the container. The motivation arises from the fact that the loading process is usually part of a broader logistics operation. This may force loading decisions before all the boxes are available, which is the case in cross-docking facilities due to the limited staging area and the focus on the minimisation of the makespan of the operations. Also, we define the stochastic version of the problem, in which the availability of the boxes may follow a probabilistic model. For solving the stochastic version, we propose a decomposition strategy which separates the temporal and geometric characteristics of the problem. This enables the use of all available (current and future) literature from container loading problems to assist in solving the CLPTAC-Om. The proposed decomposition relegates to the sub-problems the actual placing decisions and, therefore, different geometric constraints (such as stability, orientation, weight limit and distribution) can be incorporated by changing the sub-problem solution algorithm without any modification to the overall framework. The experiments quantified of a more reliable logistic environment on operational efficiency for different trade-off costs between volume utilisation and ready times.

Maximising the volume loaded is important to improve the overall efficiency of distribution system. Particularly, in an output door of cross-docking there is a pressure to dispatch trucks earlier, which could reduce loading rates. In this section, we analyse the case with one truck/container, which could be used to load all the boxes by repeating the same process iteratively leading, although, to a sub-optimal solution. Therefore, we also investigate an input minimisation version (Section 3.2), namely, minimising the number of trucks to load all the available boxes.

3.2 An input minimisation problem with time availability constraints

In many cross-docks, it is assumed a destination exclusive door management policy. In this policy, each outbound door is assigned to a single consumer. Therefore, every box

available in the staging area of that particular door is dispatched to the same destination. According to an industrial partner that we informally interviewed, it is not uncommon that multiple trucks leave for the same destination daily. Since reducing the number of trucks is important to optimise distribution costs, we propose a version of the *Input Minimisation Container Loading Problem with Time Availability Constraints*, CLPTAC-Im.

The container loading problem with time availability constraints is the problem of finding a loading plan, i.e. a three-dimensional position for each box inside one of the available containers (trucks). The loading plan must dispatch as much volume (or value) as possible as soon as possible and minimise the number (cost) of containers used. Also, it must consider: (i) The time each box arrives (becomes available for loading), (ii) the three-dimensional aspect the loading problem. We still assume orthogonal packing.

More formally, let $\mathcal{B}_t$ be a shipment (i.e. a set of boxes) arriving at the outbound door at time period $t \in \{1, \dots, \mathcal{T}\}$. Each box i is defined by its dimensions along the three axis (l_{i1}, l_{i2}, l_{i3}) . Also, let t_i and v_i be the time each box $i \in \mathcal{B} = \mathcal{B}_1 \cup \dots \cup \mathcal{B}_T$ becomes available and its volume (or value), respectively. We assume there is enough workforce to load multiple containers per period. We must load each box i in one of the available containers (trucks) $w \in \mathcal{W}$. Every container is rectangular with length, width and height given by L_1 , L_2 and L_3 , respectively, and we assume the container is positioned with its bottom-left-rear at the origin. We want to find a loading plan of boxes into containers that dispatch the greatest volume as soon as possible, using as few containers as possible. An upper bound on the number of available containers $|\mathcal{W}|$ is given by a solution for the 3D input minimisation problem (without time constraints) with boxes $i \in \mathcal{B}$, without any time availability consideration.

Thereby, let (x_{i1}, x_{i2}, x_{i3}) be non-negative variables indicating the position of the bottom-left-rear corner of box $i \in \mathcal{B}$ in axis X , Y and Z , coded as 1, 2 and 3 respectively. Binary variables p_{ik} indicating whether box $i \in \mathcal{B}$ is loaded into container/truck $w \in \mathcal{W}$. Also, binary variables $r_{ija}(s_{ija})$ indicating whether box i is positioned before (after) box $j \in \mathcal{B}$ in axis $a \in \{1, 2, 3\}$. The volume (or value) loaded into container $w \in \mathcal{W}$ at time period $t \in \mathcal{T}$ is represented by non-negative variables q_{wt} . Finally, binary variables u_{wt} indicates whether container $w \in \mathcal{W}$ is loaded at time period $t \in \mathcal{T}$. Table 7 summarises the definition of variables.

Table 7 – Definition of variables for the mixed integer linear programming model.

x_{ia} :	Non-negative position of left-bottom-front of box i in axes $a \in \{1, 2, 3\}$, respectively;
p_{iw} :	Binary variables, whether box $i \in \mathcal{B}$ is placed into vehicle $w \in \mathcal{W}$;
$r_{ija}(s_{ija})$:	Binary variables, whether box $i \in \mathcal{B}$ is before (after) box $j \in \mathcal{B}$ on axis $a \in \{1, 2, 3\}$;
q_{wt} :	Non-negative variables, the volume loaded in container $w \in \mathcal{W}$ at time period $t \in \mathcal{T}$;
u_{wt} :	Binary variables, whether container $w \in \mathcal{W}$ is loaded at time period $t \in \mathcal{T}$.

With variables in Table 7, we propose the deterministic mixed integer linear

programming model (3.9)–(3.18) for the CLPTAC-Im.

$$\text{Min } \sum_{w \in \mathcal{W}} \sum_{t \in \mathcal{T}} (t q_{wt} + u_{wt}) \quad (3.9)$$

subject to:

$$x_{ia} + l_{ia} \leq x_{ja} + (1 - r_{ija})L_a, \quad i, j \in \mathcal{B}, i < j, a \in \{1, 2, 3\}, \quad (3.10)$$

$$x_{ja} + l_{ja} \leq x_{ia} + (1 - s_{ija})L, \quad i, j \in \mathcal{B}, i < j, a \in \{1, 2, 3\}, \quad (3.11)$$

$$\sum_{a \in \{1, 2, 3\}} s_{ija} + r_{ija} \geq p_{iw} + p_{jw} - 1, \quad i, j \in \mathcal{B}, i < j, w \in \mathcal{W}, \quad (3.12)$$

$$x_{ia} \leq L_{ia} - l_{ia}, \quad i \in \mathcal{B}, a \in \{1, 2, 3\} \quad (3.13)$$

$$\sum_{w \in \mathcal{W}} p_{iw} = 1, \quad i \in \mathcal{B}, \quad (3.14)$$

$$t_i p_{iw} \leq \sum_{\bar{t} \in \mathcal{T}} \bar{t} u_{w\bar{t}}, \quad i \in \mathcal{B}, w \in \mathcal{W}, \quad (3.15)$$

$$\sum_{t \in \mathcal{T}} u_{wt} \leq 1 \quad w \in \mathcal{W}, \quad (3.16)$$

$$\sum_{t \in \mathcal{T}} q_{wt} = \sum_{i \in \mathcal{B}} v_i p_{ik}, \quad w \in \mathcal{W}, \quad (3.17)$$

$$q_{wt} \leq L_1 L_2 L_3 u_{wt}, \quad w \in \mathcal{W}, t \in \mathcal{T}, \quad (3.18)$$

Domain of the variables.

The objective function (3.9) minimises the volume in each container weighted by the time each container is ready for departure (**dispatching objective**, $\eta = \sum_{w \in \mathcal{W}} \sum_{t \in \mathcal{T}} t q_{wt}$) and also minimises the containers/trucks used ($\gamma = \sum_{w \in \mathcal{W}} \sum_{t \in \mathcal{T}} u_{wt}$). Constraints (3.10)–(3.13) are classical for modelling packing problems (e.g. [Chen, Lee and Shen \(1995\)](#)). Constraints (3.10)–(3.11) ensure there is no overlap between any pair of boxes. These constraints are only active if two boxes, i and $j \in \mathcal{B}$, are loaded into the same container (see constraints (3.12)). The position of each box is bounded by the size of the container and the size of the box in each of the three dimensions (see constraints (3.13)). Ensuring all the boxes are loaded is done by constraints (3.14). The time of each container is ready for departure is defined by constraints (3.15) and (3.16). Constraints (3.17) and (3.18) define the volume loaded into each container.

Note that there is a numeric imbalance between the objectives because the total operation time can be orders of magnitude higher than the number of containers/trucks used. In other words, the model favours the objective related to the total operation time. The priorities can be adjusted by changing the weights of the terms in the objective function.

3.2.1 A heuristic procedure

Some logistic environments might not be predictable enough to allow for an advance detailed plan of the operations. In this section, we propose a dynamic dispatching heuristic (DDH) which can quickly devise loading plans for environments in which the arrival of the boxes might be uncertain. The general idea is to solve a sub-problem for each time period, deciding, at each time period, whether a particular set of boxes should be dispatched. The DDH algorithm is described in Algorithm 1.

Algorithm 1: DDH. Outline of the solution framework for creating a loading plan for CLPTAC-Im.

Data: Set of boxes $\mathcal{B}$ in shipments $\mathcal{B}_1 \cup \dots \cup \mathcal{B}_{|\mathcal{T}|}$.
Threshold multiplier for volume α^{vol} .
Threshold multiplier for trucks α^{tr} .

Result: Loading plan for boxes in $\mathcal{B}$.

- 1 Solve an input minimisation problem for boxes $i \in \mathcal{B}$. Let V^{avg} be the average filling ratio of the trucks and $|\mathcal{W}|$ the number of trucks for that solution.
- 2 $nTrucks = 0$; $\mathcal{B}^{now} = \emptyset$. /* Trucks in the loading plan and available boxes */
- 3 **foreach** $t \in \{1, \dots, |\mathcal{T}| - 1\}$ **do** /* For each time period */
- 4 $\mathcal{B}^{now} = \mathcal{B}^{now} \cup \mathcal{B}_t$. /* Boxes available at period t */
- 5 Solve an output maximisation problem with one container and boxes $\mathcal{B}^{now}$.
Let V^{now} be the volume loaded for that solution and k the truck used.
- 6 **while** V^{now} is greater than $\alpha^{vol} V^{avg}$ **do**
- 7 $\mathcal{B}^{unloaded} =$ Boxes available which were not loaded in truck k .
- 8 $w^{extra} =$ Number of trucks in a solution of an input minimisation
problem with boxes $\mathcal{B}^{unloaded} \cup \mathcal{B}_{t+1} \cup \dots \mathcal{B}_{|\mathcal{T}|}$.
- 9 **if** $w^{extra} + nTrucks + 1 \leq \alpha^{tr} |\mathcal{W}|$ **then**
- 10 Dispatch truck k .
- 11 $nTrucks = nTrucks + 1$.
- 12 Remove dispatched boxes from $\mathcal{B}^{now}$.
- 13 **else**
- 14 **break**.
- 15 Solve an output maximisation problem with one container and boxes $\mathcal{B}^{now}$. Let V^{now} be the volume loaded for that solution and w the truck used.
- 16 Solve an input minimisation problem with boxes remaining in $\mathcal{B}^{now} \cup \mathcal{B}_{|\mathcal{T}|}$.

The first step of DDH is to solve an input minimisation disregarding the availability of the boxes in time. As an output of this procedure, we have an estimate of the average filling rate of the containers, V^{avg} , and the number of containers to be used $|\mathcal{W}|$. Then, for each time period t and correspondent available boxes, we check whether it is possible to satisfy a volume threshold $\alpha^{vol} V^{avg}$ ($\alpha^{vol} \geq 0$) for a truck w – solving an output maximisation problem for truck w . For each truck that we can satisfy it, we verify whether the total number of used trucks required to load all boxes would be a fraction $\alpha^{tr} \geq 0$ of the estimate for the number of containers to be used ($|\mathcal{W}|$). If so, we dispatch the current

truck and see if it is possible to load another one satisfying the volume threshold. If not, we go to the next period. Finally, we solve an input minimisation problem for all the boxes which were not loaded during the main loop.

Note that DDH does not specify how to solve the associated input minimisation and output maximisation problems. Provided with an algorithm for the loading sub-problems, DDH can be seen as a generic framework for taking decisions in a rolling horizon environment in which boxes become available dynamically. As it was the case for the dynamic programming algorithm of Section 3.1.1, this enables some flexibility regarding the practical aspects that can be considered for the loading (e.g. weight limit, weight distribution and stability). Also, by design, DDH can be applied to a case in which the arrival of shipments $\mathcal{B}_1 \cup \dots \cup \mathcal{B}_{|\mathcal{T}|}$ is a random event. Despite its flexibility, it can quickly (in less than a second) provide solutions. To evaluate the algorithm, we report results when using the solution procedure described by George and Robinson (1980) for the loading sub-problems, iteratively. This procedure consists of identifying three-dimensional spaces inside the container, positioning a box on its bottom-left-rear corner, then updating the list of empty spaces. This procedure is simple to be executed within loop 3-15. Also it is flexible to accommodate different practical constraints. Moreover, we illustrate our approach considering a particular probabilistic model for the arrival of the boxes (see Section 3.2.2).

3.2.2 Computational experiments for CLPTAC-Im

We performed computational experiments to: (i) evaluate the sensibility of the the DDH heuristic to the parameters α^{vol} and α^{tr} , (ii) to compare the solutions from the model and the heuristic and (iii) evaluate different dispatching policies derived from the heuristic. All the experiments were performed in the same set of instances, which were generated based on a benchmark from the literature. In Subsection 3.2.2.1, we evaluate the sensibility of the DDH to its parameters. Subsection 3.2.2.2 presents a comparison between the performance of the model against the heuristic and then the DDH heuristic and different dispatching policies are evaluated.

The benchmark instances for the CLPTAC-Im is the same described in Section 3.1.2. That is, it was generated based on the 47 instances proposed by Ivancic, Mathur and Mohanty (1989) for the Container Loading Problem (CLP). In this benchmark, instances have from two to five types of boxes, the number of boxes varies from 47 to 181 and there is only one type of container per instance – containers have different dimensions across instances. For each box, we generated a time from which it becomes available, simulating the arrival process. All boxes arriving at the same time define a shipment. This time was generated randomly, following a discrete uniform probability distribution, and is an integer number in $[1, |\mathcal{T}|]$, $|\mathcal{T}| = 10$. Therefore, we are assuming a planning horizon with 10-time slots, for example, in an eight-hour working day, this could mean that each time slot has

40 minutes.

For solving the model, we used Gurobi 7.5 in Ubuntu 12.04 with a time limit of 3600 seconds. Regarding hardware settings, we used four cores of an Intel® Xeon CPU E5-2620 at a frequency of 2 GHz each. For the heuristics, we used the same hardware settings, the solution time for any instance was less than a second so we do not detail them in the followings. Moreover, for the stochastic cases, we repeat the solution procedure 1000 times per instance to get an average behaviour.

3.2.2.1 Parameter settings for the heuristics

To gain more insight on the behaviour of the heuristic, we evaluate several combinations of the parameters $(\alpha^{vol}, \alpha^{tr})$, considering deterministic arrival times

The heuristic is not sensible to the threshold of volume (α^{vol}) if it is not flexible in the number of trucks to be used (given by parameter α^{tr}). Figure 11 shows that for small values of α^{tr} (the least transparent curves) both the dispatching objective (η) and the number of trucks (γ) are not sensible to variations in α^{vol} . If it is possible to use more trucks than what was previously defined, then we can, potentially, improve η for lower values of α^{vol} . This is expected since we could dispatch boxes as soon as they become available if we had enough trucks. Although using low values of α^{vol} seems unrealistic for a limited fleet.

Being strict on the number of trucks to be used (lower values for α^{tr}) lead to dispatching in later time periods. In Figure 11, values $\alpha^{tr} \leq 1.0$ lead to a stable curves for the number of vehicles. This behaviour is typical when we decide to not dispatch any trucks until the last period. In other words, the procedure is giving the highest priority to the consolidation of goods (number of trucks, γ) over the dispatching objective (η).

3.2.2.2 Evaluating the model

To evaluate the model, we compared it against different settings of the heuristic procedure in DDH (Algorithm 1).

We report the time (in seconds) consumed to solve model (3.9)–(3.18), the optimality gap (the ratio between the difference of upper and lower bound and the upper bound), the solution regarding the dispatching (η) and number of trucks (γ), see tables 8 and 9. Also, we use superscript 1 for results related to the model and 2 for the heuristic. Moreover, in the last two columns, we provide the ratio between the two solution approaches: A value smaller than one means the model provided a better solution and larger than one means Algorithm 1 was better.

The solver struggles to prove optimality for the majority of the instances. It could only prove optimality for 2 out of 47 instances (see Table 8 or 9), the average optimality gap is 55%. We compared the solution of the model with the heuristic (Algorithm 1) with

Figure 11 – Average of dispatching objective (η , circles) and number of trucks (γ , triangles) related to values of α^{vol} and α^{tr} . Values of α^{vol} are shown in the horizontal axis and $\alpha^{tr} \in \{0.1, 0.2, \dots, 1.7\}$ are coded in the transparency of the curves, the higher the transparency the higher the α^{tr} value. Note that some combinations lead to the same curves. The corresponding arrows indicate the increasing direction α^{tr} .

two combination of parameters $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$ and $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.4)$. For the first combination, we can see (Table 8) that the model and heuristic use the same number of trucks in all but five of the instances. Regarding the dispatching objective (η), the model is able to find solutions 6% better on average. The largest difference in performance in favour of the model (0.52) was obtained for instance 23. In instance 22, we had the greatest difference in performance in favour of the heuristic (1.63). To evaluate statistical significance, a Wilcoxon test was performed separately for the two objectives. For the dispatching objective, there is weak evidence against the null hypothesis (p-value of 14%), therefore the performances of the model and DDH cannot be considered different. The same conclusion can be drawn when considering the objective related to the number of trucks (p-value of 18%).

For the second combination (Table 9), the number of trucks (γ) remains about the same but the heuristic is 6% better than the model on average with the model being better at four instances and the heuristic better at two. The greatest difference in favour of the model was 0.86 and in favour of the heuristic was 1.18. Regarding the dispatching objective (η), the model outperformed the heuristic in 21 instances and was outperformed in 24. The greatest difference in performance in favour of the model was 0.65 and in favour of the heuristic was 1.54. The Wilcoxon test was used to test for statistical significance. The test revealed only weak evidence (p-value of around 38%) of difference in performance when analysing the dispatching objective. The same happen when evaluating the difference in performance related to the number of trucks (p-value of around 41%).

Table 8 – Comparison between model (3.9)–(3.18) and DDH with $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$. TL indicates that the time limit (3600 seconds) was reached for that instance. For the DDH heuristic, the computational time was less than one second for each instance. For each objective, for each instance, best values are marked in bold.

Id	Model				DDH		Ratio	
	Time (s)	Gap	η^1	γ^1	η^2	γ^2	$\frac{\eta^1}{\eta^2}$	$\frac{\gamma^1}{\gamma^2}$
1	TL	0.07	104063	50	151360	50	0.69	1.00
2	TL	0.75	114111	17	132640	17	0.86	1.00
3	TL	0.88	1038144	33	1161600	33	0.89	1.00
4	TL	0.87	818304	29	667904	29	1.23	1.00
5	TL	0.63	272448	19	212864	18	1.28	1.06
6	TL	0.69	687500	10	447500	10	1.54	1.00
7	TL	0.87	586625	19	702500	19	0.84	1.00
8	TL	0.65	607500	6	702500	6	0.86	1.00
9	TL	0.83	20123903	37	19418752	37	1.04	1.00
10	TL	0.83	18803967	55	19776000	55	0.95	1.00
11	TL	0.85	21239039	45	20331904	45	1.04	1.00
12	TL	0.01	280500	3	450000	3	0.62	1.00
13	TL	0.69	5999499	40	6049500	40	0.99	1.00
14	TL	0.51	6765000	40	6238500	40	1.08	1.00
15	TL	0.77	1089600	15	1168800	15	0.93	1.00
16	TL	0.85	948239	33	957360	33	0.99	1.00
17	TL	0.76	706560	11	804480	11	0.88	1.00
18	4	0.00	62139	2	83400	2	0.75	1.00
19	3	0.00	58799	2	83400	2	0.71	1.00
20	TL	< 0.01	54757	3	83400	3	0.66	1.00
21	TL	0.78	584114	33	728247	33	0.80	1.00
22	TL	0.84	812452	14	498744	13	1.63	1.08
23	TL	0.07	54911	5	105600	5	0.52	1.00
24	TL	0.60	580019	10	563950	9	1.03	1.11
25	TL	0.53	878420	7	1060800	7	0.83	1.00
26	TL	0.36	843080	4	1060800	4	0.79	1.00
27	TL	0.58	470823	6	484740	6	0.97	1.00
28	TL	0.75	409300	16	484740	16	0.84	1.00
29	TL	0.86	1130500	31	899120	31	1.26	1.00
30	TL	0.86	1005568	35	814546	34	1.23	1.03
31	TL	0.87	1108814	28	834561	28	1.33	1.00
32	TL	0.30	160236	6	252000	6	0.64	1.00
33	TL	0.24	110088	3	122400	3	0.90	1.00
34	TL	0.41	116820	5	122400	5	0.95	1.00
35	TL	0.34	216432	3	241920	3	0.89	1.00
36	TL	0.82	194063	24	145248	24	1.34	1.00
37	TL	0.85	732703	26	906752	26	0.81	1.00
38	TL	0.77	570431	61	879072	61	0.65	1.00
39	TL	0.74	831264	19	589968	19	1.41	1.00
40	TL	0.65	414411	13	343856	14	1.21	0.93
41	TL	0.80	311231	25	480600	25	0.65	1.00
42	TL	0.28	1899185	6	3554700	6	0.53	1.00
43	TL	0.01	2075120	4	3554700	4	0.58	1.00
44	TL	0.05	2125428	5	3554700	5	0.60	1.00
45	TL	0.37	1638950	3	1638950	3	1.00	1.00
46	TL	0.08	1253867	3	1638950	3	0.77	1.00
47	TL	0.45	1638950	4	1638950	4	1.00	1.00
Average	3446	0.55	2181445	18.47	2272880	18.40	0.94	1.00

Table 9 – Comparison between model (3.9)–(3.18) and Algorithm DDH with $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.4)$. TL indicates that the time limit (3600 seconds) was reached for that instance. For the DDH heuristic, the computational time was less than one second for each instance. For each objective, for each instance, best values are marked in bold.

Id	Model				DDH		Ratio	
	Time (s)	Gap	η^1	γ^1	η^2	γ^2	$\frac{\eta^1}{\eta^2}$	$\frac{\gamma^1}{\gamma^2}$
1	TL	0.07	104063	50	151360	50	0.69	1.00
2	TL	0.75	114111	17	132640	17	0.86	1.00
3	TL	0.88	1038144	33	1161600	33	0.89	1.00
4	TL	0.87	818304	29	667904	29	1.23	1.00
5	TL	0.63	272448	19	212864	18	1.28	1.06
6	TL	0.69	687500	10	447500	10	1.54	1.00
7	TL	0.87	586625	19	702500	19	0.84	1.00
8	TL	0.65	607500	6	702500	6	0.86	1.00
9	TL	0.83	20123903	37	19418752	37	1.04	1.00
10	TL	0.83	18803967	55	19776000	55	0.95	1.00
11	TL	0.85	21239039	45	20331904	45	1.04	1.00
12	TL	0.01	280500	3	280500	3	1.00	1.00
13	TL	0.69	5999499	40	6049500	40	0.99	1.00
14	TL	0.51	6765000	40	6238500	40	1.08	1.00
15	TL	0.77	1089600	15	1168800	15	0.93	1.00
16	TL	0.85	948239	33	957360	33	0.99	1.00
17	TL	0.76	706560	11	804480	11	0.88	1.00
18	4	0.00	62139	2	83400	2	0.75	1.00
19	3	0.00	58799	2	83400	2	0.71	1.00
20	TL	< 0.01	54757	3	61278	3	0.89	1.00
21	TL	0.78	584114	33	728247	33	0.80	1.00
22	TL	0.84	812452	14	498744	13	1.63	1.08
23	TL	0.07	54911	5	54912	5	1.00	1.00
24	TL	0.60	580019	10	563950	9	1.03	1.11
25	TL	0.53	878420	7	770080	7	1.14	1.00
26	TL	0.36	843080	4	718720	4	1.17	1.00
27	TL	0.58	470823	6	357310	6	1.32	1.00
28	TL	0.75	409300	16	484740	16	0.84	1.00
29	TL	0.86	1130500	31	899120	31	1.26	1.00
30	TL	0.86	1005568	35	814546	34	1.23	1.03
31	TL	0.87	1108814	28	834561	28	1.33	1.00
32	TL	0.30	160236	6	151596	6	1.06	1.00
33	TL	0.24	110088	3	88992	3	1.24	1.00
34	TL	0.41	116820	5	84744	5	1.38	1.00
35	TL	0.34	216432	3	155856	3	1.39	1.00
36	TL	0.82	194063	24	145248	24	1.34	1.00
37	TL	0.85	732703	26	906752	26	0.81	1.00
38	TL	0.77	570431	61	879072	61	0.65	1.00
39	TL	0.74	831264	19	589968	19	1.41	1.00
40	TL	0.65	414411	13	343856	14	1.21	0.93
41	TL	0.80	311231	25	480600	25	0.65	1.00
42	TL	0.28	1899185	6	2380968	7	0.80	0.86
43	TL	0.01	2075120	4	2267428	4	0.92	1.00
44	TL	0.05	2125428	5	2520220	5	0.84	1.00
45	TL	0.37	1638950	3	1110520	3	1.48	1.00
46	TL	0.08	1253867	3	1151748	3	1.09	1.00
47	TL	0.45	1638950	4	1244468	4	1.32	1.00
Average	3446	0.55	2181445	18.47	2141696	18.43	1.06	1.00

The model (3.9)–(3.18) and the heuristics in Algorithm 1 have a competitive performance, specially if there are enough trucks available. Also, some operations might have a greater flexibility with the parameters.

Stochastic case

We used Algorithm 1 to elaborate different dispatching policies to be used in a stochastic environment, namely, the arrival of the shipments follow a probabilistic distribution. We used the same probability model defined in Section 3.1.1. That is, the probability of arrival for a shipment $\mathcal{B}_t$, that is schedule to arrive at t is $\beta \in (0, 1]$. While the shipment (set of boxes) does not arrive, the probability of it arriving in a following period is again β . To simulate the arrival process, we assume a discrete time horizon of $\mathcal{T}$ time steps and, at each time step, we sampled the shipments that have arrived.

We consider three variants of DDH (**case 1**, **case 2** and **case 3**) and compare it against each other and against a baseline case, which we called **case 0**. In **case 0**, we have a 50% chance of dispatching a truck as long as there are boxes to load. For **case 1**, before the simulation, we run DDH in the deterministic setting, assuming each shipment arrives on time. Then, we observe, for each time period, whether any truck was dispatched. Case 1 consists of implementing these observations in the stochastic setting, that is, if there was a dispatching in the deterministic run, we dispatch all boxes available at the time. For **case 2**, we proceed in a similar fashion, running the deterministic version of DDH before the simulation, but we observe the number of trucks dispatched in each time period. Then, in the simulation, for each time period, we dispatch the same number of trucks (as long as there are boxes to be loaded). Finally, **case 3** consist of running the DDH heuristic inside the simulation, in each time period, and implementing the decision of the heuristic for the current period. Figures 12 and 13 qualify the typical average behaviour of these approaches. We choose to present these results with parameters $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$ for the heuristic because other reasonable choices produce the same qualitative behaviour.

Although case 3 is the most adaptable, it is not always the best choice, on average. Figures 12 and 13 show the average results for $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$ for the dispatching (η) and the number of trucks (γ), respectively. As expected, the random approach (case 0) is dominated by cases 1, 2 and 3 regarding both objectives dispatching (η) and number of trucks (γ). When considering the dispatching objective, case 1 is the most effective (see Figure 12), however, it used more trucks than cases 2 and 3 (Figure 13). Moreover, cases 2 and 3 are more resilient, on average, regarding the variation of probabilities of arrival (β), which might be seen as an advantage. Finally, Table 10 provides some details on the results comparing cases 1, 2 and 3. The difference between cases 1 and 3 increased as the probability of arrival increased, particularly regarding the number of trucks. Regarding the comparison between cases 2 and 3, the difference is consistently less sensible for both objectives.

Figure 12 – Comparison cases 0, 1, 2 and 3 for $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$ for the dispatching objective (η) different probabilities of arrival (β).

Figure 13 – Comparison cases 0, 1, 2 and 3 for $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$ for the number of trucks objective (γ) different probabilities of arrival (β).

Table 10 – Comparison between case 1, 2 and 3 (used as superscripts) for each instance for $(\alpha^{vol}, \alpha^{tr}) = (1.0, 1.1)$. Results were averaged across 1000 runs of the simulation. The computational time of each run, for every case, was less than a second.

Id	0.5		0.6		0.7		0.8		0.9		1.0	
	$\frac{\eta^1}{\eta^3}$	$\frac{\eta^2}{\eta^3}$	$\frac{\eta^1}{\eta^3}$	$\frac{\eta^2}{\eta^3}$	$\frac{\eta^1}{\eta^3}$	$\frac{\eta^2}{\eta^3}$	$\frac{\eta^1}{\eta^3}$	$\frac{\eta^2}{\eta^3}$	$\frac{\eta^1}{\eta^3}$	$\frac{\eta^2}{\eta^3}$	$\frac{\eta^1}{\eta^3}$	$\frac{\eta^2}{\eta^3}$
1	0.78	1.00	0.76	1.00	0.74	1.00	0.73	1.00	0.72	1.00	0.71	1.00
2	0.85	1.01	0.83	1.01	0.82	1.01	0.81	1.01	0.80	1.01	0.79	1.00
3	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
4	0.85	1.01	0.83	1.01	0.82	1.01	0.81	1.01	0.80	1.01	0.79	1.00
5	0.95	1.04	0.95	1.05	0.94	1.05	0.94	1.04	0.94	1.03	0.93	1.00
6	0.92	1.03	0.91	1.03	0.90	1.03	0.89	1.02	0.89	1.01	0.89	1.00
7	1.01	1.01	1.01	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
9	0.78	1.01	0.76	1.02	0.74	1.01	0.73	1.01	0.72	1.01	0.71	1.00
10	0.78	1.02	0.76	1.02	0.75	1.02	0.74	1.01	0.73	1.01	0.72	1.00
11	0.82	1.01	0.80	1.01	0.79	1.01	0.78	1.01	0.78	1.01	0.77	1.00
12	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
13	0.95	1.05	0.94	1.05	0.94	1.05	0.94	1.04	0.93	1.02	0.93	1.00
14	0.89	1.02	0.88	1.02	0.87	1.02	0.87	1.02	0.86	1.01	0.86	1.00
15	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
16	0.76	1.00	0.74	1.00	0.73	1.00	0.71	1.01	0.70	1.00	0.69	1.00
17	0.88	1.02	0.87	1.02	0.85	1.02	0.85	1.02	0.84	1.01	0.83	1.00
18	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
19	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
20	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
21	0.77	0.99	0.75	0.99	0.73	1.00	0.72	1.00	0.71	1.00	0.70	1.00
22	0.88	0.97	0.88	0.97	0.90	1.00	0.91	1.00	0.94	1.01	0.96	1.00
23	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
24	0.94	1.02	0.94	1.02	0.94	1.02	0.95	1.02	0.95	1.01	0.97	1.00
25	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
26	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
27	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
28	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
29	0.89	0.98	0.88	0.99	0.87	1.00	0.87	1.00	0.86	1.00	0.86	1.00
30	0.89	0.99	0.89	0.99	0.88	0.99	0.88	1.00	0.88	1.00	0.88	1.00
31	0.87	1.02	0.86	1.03	0.85	1.03	0.84	1.03	0.83	1.02	0.83	1.00
32	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
33	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
34	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
35	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
36	0.95	1.03	0.94	1.03	0.94	1.03	0.94	1.03	0.94	1.02	0.94	1.00
37	0.76	0.99	0.74	0.99	0.72	0.99	0.71	1.00	0.70	1.00	0.69	1.00
38	0.69	0.98	0.66	0.99	0.64	0.99	0.63	0.99	0.62	1.00	0.61	1.00
39	0.93	1.05	0.93	1.05	0.91	1.05	0.91	1.04	0.90	1.02	0.90	1.00
40	0.82	1.00	0.80	1.00	0.79	1.00	0.78	1.01	0.77	1.00	0.76	1.00
41	1.01	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
42	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
43	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
44	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
45	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
46	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
47	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Average	0.93	1.01	0.92	1.01	0.92	1.01	0.91	1.01	0.91	1.00	0.91	1.00

Regarding statistical differences, we performed a Wilcoxon signed-rank test for each pair of case and for each value of β (see tables 11 and 12). The tests did not reveal a strong evidence against difference between the random case (case 0) and the others for the dispatching objective. Although there is evidence in favour of different performances for every pair of case. On the other hand, for the number of trucks, the associated p-values, in general, can guarantee a statistical difference between any pair of cases. The series used in the test were the average values of the 1000 runs of the algorithm.

Table 11 – P-values for the Wilcoxon test for each pair of cases for each probability of arrival (β) considering the dispatching objective.

Case x case	Probability of arrival					
	0.5	0.6	0.7	0.8	0.9	1.0
0 x 1	0.3855	0.3914	0.3684	0.3972	0.4091	0.4152
0 x 2	0.6644	0.6491	0.6415	0.6415	0.6721	0.6644
0 x 3	0.4213	0.4398	0.4398	0.4461	0.4274	0.4152
1 x 2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1 x 3	0.0014	0.0003	0.0001	0.0000	0.0000	0.0000
2 x 3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 12 – P-values for the Wilcoxon test for each pair of cases for each probability of arrival (β) considering the number of trucks objective.

Case x case	Probability of arrival					
	0.5	0.6	0.7	0.8	0.9	1.0
0 x 1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0 x 2	0.0005	0.0029	0.0092	0.0194	0.0309	0.1100
0 x 3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1 x 2	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1 x 3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2 x 3	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

3.2.3 Remarks on CLPTAC-Im

We explore the container loading problem which happens in outbound doors of cross-docking facilities. The main difference from traditional container loading problems is that not all boxes are promptly available for loading. To deal with this problem, we propose a mixed integer programming model and a heuristic framework for the container loading problem with time availability constraints. Although the model can provide reasonable solutions, it is not flexible enough to incorporate many of the aforementioned practical constraints. On the other hand, the heuristic naturally allows the use of any algorithm to perform the loading, enabling the use of all the research in solutions techniques for the more traditional container loading problems. Also, we use the heuristic to generate policies for dispatching trucks in the case of uncertainty in the arrival times of the boxes. Despite its flexibility, it is also simple and competitive, providing solution comparable to the mixed integer linear programming model. To the best of our knowledge, this is the first time container loading problems are explored in a more dynamic setting.

3.3 General remarks on container loading with time availability constraints

Decades of research in container loading problems have extended their applicability to a wide range of practical constraints (e.g. vertical and horizontal stability, axle weight and weight distribution). We propose to further extend its applicability to a case in which the boxes have a scheduled arrival as opposed to being promptly available for loading. This feature is aligned with the cross-docking philosophy of having a synchronisation of inbound and outbound flows, aiming at minimum storage.

Besides the extensions for the other classes of container loading problems, like multiple containers and open dimension problems, more temporal characteristics can be explored in the future. For example, the Container Loading Problem with Time Availability Constraints (CLPTAC) could be extended for just-in-time environments, in which it is essential to follow a predefined schedule. Also, workforce limitations and resource availability could be contemplated, limiting, for example, the number of boxes that can be loaded per shift or the time availability of the containers.

NETWORK SCHEDULING WITH CROSS-DOCKING

Besides the loading process happening at cross-docks, another challenge of distribution systems in city logistics is the operation of the distribution network itself. It can still be considered an operational problem but with a broader view of the network when compared to the container loading problem studied in Chapter 3. In this sense, the container loading problems explored in Chapter 3 are sub-problems of the network scheduling problem with cross-docking we explore in this chapter.

In this chapter, we briefly revisit the literature review in cross-docking (presented in detail in Chapter 2) to motivate a problem of network scheduling with cross-docking. Then, we formally introduce the problem to be studied and propose a mixed integer linear programming model (Section 4.1) for it. Furthermore, as a solution procedure, we developed a method based on Benders decomposition (Section 4.2). Computational experiments and some concluding remarks are reported in sections 4.3 and 4.4, respectively.

As presented in Chapter 2, the optimisation literature in cross-docking has mostly focused on optimising the operation of a single facility. Perhaps, the most popular optimisation problem in this context is the scheduling of trucks at input and output docks. Research has been done on the case with one input and one output dock (some examples are the works by Yu and Egbelu (2008), Chen and Song (2009), Arabani, Ghomi and Zandieh (2010), Vahdani and Zandieh (2010), Arabani, Ghomi and Zandieh (2011), Liao, Egbelu and Chang (2012), Keshtzari, Naderi and Mehdizadeh (2016), Amini and Tavakkoli-Moghaddam (2016)). Although the one-dock case is mostly unrealistic, authors agree that it might provide insights into promising solution methodologies and a better understanding of the behaviour of larger systems. Multi-dock cases have recently been studied (e.g. Bellanger, Hanafi and Wilbaut (2013), Assadi and Bagheri (2016), Cota *et al.* (2016), Ladier and Alpan (2018)). Different cross-dock characteristics were also modelled. Some consider time windows for trucks to be loaded or unloaded (Miao, Lim and

Ma, 2009), limited resources for moving goods from inbound to outbound docks (Shakeri *et al.*, 2012), uncertainty in arrival times of goods (e.g. Konur and Golias (2013) and Ladier and Alpan (2016)), goods' handling time/effort (e.g. Zhang *et al.* (2010) and Maknoon, Soumis and Baptiste (2017)), travelling distance inside the cross-dock (Chmielewski *et al.*, 2009) and limited internal storage (e.g. Miao, Lim and Ma (2009), Gelareh *et al.* (2016), Tootkaleh, Ghomi and Sajadieh (2016) and Bazgosha, Ranjbar and Jamili (2017)). For detailed reviews, the reader is referred to the papers by Agustina, Lee and Piplani (2010), Stephan and Boysen (2011), Belle, Valckenaers and Cattrysse (2012), Buijs, Vis and Carlo (2014) and Guastaroba, Speranza and Vigo (2016).

The literature approaching transportation networks with cross-docking is much more scarce. Most of the research exploring this kind of network focused on closed networks, i.e., the routes of the vehicles start and end at the same node (e.g. Lee, Jung and Lee (2006), Wen *et al.* (2009), Dondo, Méndez and Cerdá (2011), Vahdani *et al.* (2012), Moghadam, Ghomi and Karimi (2014), Chen *et al.* (2016), Maknoon and Laporte (2017), Enderer, Contardo and Contreras (2017), Ahkamiraad and Wang (2018)). The open network scheduling problem with cross-docking remains, to the best of our knowledge, a rather unexplored case. Geoffrion and Graves (1974) were one of the firsts to study an open network setting with distribution centres similar to cross-docks. In this seminal work, the authors formulated a mixed integer linear programming model for the selection of distribution centres to be used as intermediate facilities for suppliers and consumers. For the solution of the model, a Benders decomposition proved its efficacy in finding optimal solutions. This is one of the first successful applications of Benders decomposition to a practical/realistic problem (Costa, 2005). Chen *et al.* (2006) proposed a model and a neighbourhood search heuristic for the open case with multiple cross-docks. Yu, Jewpanya and Redi (2016) proposed the *Open Vehicle Routing Problem with Cross-docking*, which is related to the problem studied by Chen *et al.* (2006) but without time windows and with a single cross-dock. Moreover, Hernández, Peeta and Kalafatas (2011) contemplated the case with multiple carriers which needed to move goods from origin to destination allowing collaboration among the carriers. The authors proposed a model for identifying a set of routes which minimises the total transportation costs.

Even though there is some research on open network problems (Geoffrion and Graves, 1974), (Chen *et al.*, 2006), (Hernández, Peeta and Kalafatas, 2011) and (Yu, Jewpanya and Redi, 2016)), the case we explore here has some particularities. Geoffrion and Graves (1974) approach a more tactical situation, deciding the flow of goods through consolidation centres and which centres to open. We focus on an operational problem, deciding on where to send trucks (rather than flows) and how to recombine boxes from different suppliers into trucks. We also consider possible delays (not present in (Geoffrion and Graves, 1974)). Chen *et al.* (2006) consider a network with two layers, similar to ours, but uses a discrete time horizon to manage the cross-docks inventory levels, whilst we focus on the distribution aspects, limiting the capacity of trucks. Hernández, Peeta and

Figure 14 – Scheme of a logistic network with cross-docks (consolidation centres). Suppliers $o_1, \dots, o_{|\mathcal{O}|}$ must send their shipments (set of goods) to one of the consolidation centres ($f_1, \dots, f_{|\mathcal{F}|}$). Then, goods are loaded into trucks and dispatched to their final destination ($d_1, \dots, d_{|\mathcal{D}|}$).

Kalafatas (2011) focus on a collaborative network, in which carriers may decide when and from which node they want to enter and exit the network. In our case, all participants of the network are previously defined and all supply and demand is always part of the same network. Finally, Yu, Jewpanya and Redi (2016) consider a single cross-dock manager that outsourced pick-up and delivery activities. Multiple visits in pick-up and delivery open routes are allowed and the synchronisation at the cross-dock is achieved by forcing the fleet to arrive at the cross-dock at the same time. On the other hand, we consider multiple cross-docks, with single delivery visits and achieve synchronisation by the availability of the outbound goods. Moreover, we focus on the open network scheduling problem with multiple cross-docks (see Figure 14). Consumers demand goods from different suppliers but instead of sending the goods directly, a supplier must ship them to a cross-dock where its load is consolidated with that of other suppliers. This strategy aims to increase consolidation, justifying the direct shipment between cross-docks and consumers.

Our contribution here is twofold. We developed a mixed integer linear programming model for minimising the cost of transportation and delays in the delivery (Section 4.1) and an effective decomposition algorithm based on logic-based Benders (Hooker and Ottosson, 2003) to solve the problem (Section 4.2). Both model and algorithm are evaluated on a set of randomly generated instances (Section 4.3). Section 4.4 ends this chapter with an overall analysis of the results.

4.1 Problem definition and model

We focus on distribution networks in which a set of suppliers (origins), $\mathcal{O}$, satisfies demands from a set of consumers (destinations), $\mathcal{D}$, by shipping their goods through intermediate (cross-docking) facilities, $\mathcal{F}$ (Figure 14). We consider that the distribution operation happens within a time interval $[0, T]$. Leaving each origin $o \in \mathcal{O}$, there is a set of shipments $k \in \mathcal{K}_{\mathcal{O}}$ and each shipment contains a set of boxes, or goods, $\mathcal{B}_k$. Let h_i be

the volume of each box $i \in \mathcal{B}$, $\mathcal{B} = \bigcup_{k \in \mathcal{K}} \mathcal{B}_k$. Also, let UT_k and ST_k be the time required to unload and sort shipment $k \in \mathcal{K} = \bigcup_{o \in \mathcal{O}} \mathcal{K}_o$, respectively. Each cross-dock $f \in \mathcal{F}$ has a set of available vehicles $\mathcal{W}_f$ (possibly heterogeneous), each with capacity CAP_w and loading time LT_w , $w \in \mathcal{W} = \bigcup_{f \in \mathcal{F}} \mathcal{W}_f$. Each destination $d \in \mathcal{D}$ demands for the set of boxes $\mathcal{B}_d$. All boxes demanded are supplied (i.e. $\bigcup_{k \in \mathcal{K}} \mathcal{B}_k = \bigcup_{d \in \mathcal{D}} \mathcal{B}_d$). Furthermore, let the cost and time of transporting goods from supplier $i \in \mathcal{O}$ to cross-dock $j \in \mathcal{F}$ or from cross-dock $i \in \mathcal{F}$ to consumer $j \in \mathcal{D}$ be c_{ij} and t_{ij} , respectively. We do not consider exchanges between cross-docks. Finally, there is a deadline τ_d associated with each consumer $d \in \mathcal{D}$. All the trucks must reach their respective destinations before the deadline, otherwise there is a penalty of R per unit of delay, per late delivery work.

Note that, since there are no constraints on the processing capabilities of the cross-docks $f \in \mathcal{F}$, we can compute the time from which each inbound box would become available for loading in facility f as $\bar{u}_{if} = t_{of} + UT_k + ST_k$, $o \in \mathcal{O}$, $k \in \mathcal{K}$, $i \in \mathcal{B}_k$, $f \in \mathcal{F}$. Also, we assume that each truck visits at most a single demand point (consumer), but a consumer might be visited by multiple trucks. All parameters are summarised in Table 13.

Table 13 – Summary of parameters for the network scheduling problem.

$\mathcal{O}$:	Set of origins of the distribution (suppliers in the network).
$\mathcal{D}$:	Set of destinations (consumers in the network).
$\mathcal{F}$:	Set of intermediate facilities.
T :	Length of the planning horizon, used in the continuous model.
$\mathcal{K}_o$:	Set of all shipments leaving urban distribution centre $o \in \mathcal{O}$.
$\mathcal{K}$:	Set of all shipments ($\mathcal{K} = \bigcup_{o \in \mathcal{O}} \mathcal{K}_o$).
$\mathcal{B}_k$:	Set of boxes in shipment $k \in \mathcal{K}$.
$\mathcal{B}_d$:	Set of boxes for client $d \in \mathcal{D}$.
$\mathcal{B}$:	Set of all boxes ($\mathcal{B} = \bigcup_{k \in \mathcal{K}} \mathcal{B}_k = \bigcup_{d \in \mathcal{D}} \mathcal{B}_d$).
UT_k :	Unloading time of shipment $k \in \mathcal{K}$.
ST_k :	Sorting time of shipment $k \in \mathcal{K}$.
$\mathcal{W}_f$:	Set of vehicles available at facility $f \in \mathcal{F}$.
CAP_w :	Capacity, in number of boxes, of vehicle $w \in \mathcal{W}$.
LT_w :	Loading time for truck $w \in \mathcal{W}$.
c_{ij} :	Travel cost (distance) between points (nodes) i and j in the distribution network.
$\bar{u}_{if}$:	The time from which box $i \in \mathcal{B}$ is available for loading in a truck in facility $f \in \mathcal{F}$.
t_{ij} :	Travel time between points $i \in \mathcal{O} \cup \mathcal{F}$ and $j \in \mathcal{F} \cup \mathcal{D}$.
τ_d :	Deadline associated with client $d \in \mathcal{D}$.
h_i :	Volume of box $i \in \mathcal{B}$.
R :	Penalty per time unit of delay.

The goal is to find an assignment of the shipments leaving suppliers to cross-docks and of boxes to trucks which minimises total distribution and delay costs. Therefore, let Δ_{kf} be binary variables indicating whether shipment $k \in \mathcal{K}$ is assigned to cross-dock $f \in \mathcal{F}$ and p_{iw} be binary variables indicating whether box $i \in \mathcal{B}$ is assigned to truck $w \in \mathcal{W}$. Also, binary variables v_{wd} define whether truck $w \in \mathcal{W}$ is assigned to destination $d \in \mathcal{D}$. Moreover, there are continuous non-negative variables m_w defining the time a

truck $w \in \mathcal{W}$ is ready for departure and e_{wd} defining the lateness of delivery truck w for consumer d . In summary, we have the variables in Table 14. Thereby, we formulate the resulting network scheduling problem with cross-dock (NSCD), (4.1)–(4.8)

Table 14 – Summary of the variables for the network scheduling problem.

Δ_{kf} :	Binary variables indicating whether shipment $k \in \mathcal{K}_o$ is sent to facility $f \in \mathcal{F}$.
p_{iw}	Binary variables, whether box $i \in \mathcal{B}$ is placed into vehicle $w \in \mathcal{W}$.
v_{wd} :	Binary variables indicating whether vehicle $w \in \mathcal{W}_f$, goes to client $d \in \mathcal{D}$.
m_w :	Time from which truck $w \in \mathcal{W}$ is ready for departure.
e_{wd} :	Lateness associated with truck $w \in \mathcal{W}$ and destination $d \in \mathcal{D}$.

$$(\text{NSCD}) \text{Min} \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}_o} \sum_{f \in \mathcal{F}} c_{of} \Delta_{kf} + \sum_{f \in \mathcal{F}} \sum_{w \in \mathcal{W}_f} \sum_{d \in \mathcal{D}} c_{fd} v_{wd} + R \sum_{w \in \mathcal{W}} \sum_{d \in \mathcal{D}} e_{wd} \quad (4.1)$$

subject to:

$$\sum_{f \in \mathcal{F}} \Delta_{kf} = 1, \quad k \in \mathcal{K}, \quad (4.2)$$

$$\Delta_{kf} = \sum_{w \in \mathcal{W}_f} p_{iw}, \quad f \in \mathcal{F}, k \in \mathcal{K}, i \in \mathcal{B}_k, \quad (4.3)$$

$$\sum_{i \in \mathcal{B}} h_i p_{iw} \leq CAP_w, \quad w \in \mathcal{W}_f, f \in \mathcal{F}, \quad (4.4)$$

$$v_{wd} \geq p_{iw}, \quad d \in \mathcal{D}, i \in \mathcal{B}_d, w \in \mathcal{W}, \quad (4.5)$$

$$\sum_{d \in \mathcal{D}} v_{wd} \leq 1, \quad w \in \mathcal{W}, \quad (4.6)$$

$$m_w \geq (LT_w + \bar{u}_{if}) p_{iw}, \quad f \in \mathcal{F}, w \in \mathcal{W}_f, i \in \mathcal{B}, \quad (4.7)$$

$$m_w + t_{fd} - \tau_d - (1 - v_{wd})T \leq e_{wd}, \quad d \in \mathcal{D}, f \in \mathcal{F}, w \in \mathcal{W}_f. \quad (4.8)$$

Domain of the variables.

The objective function (4.1) minimises the cost of first and second layer distributions and the cost of delays. The first layer distribution cost is defined as the total cost of moving goods from origins $o \in \mathcal{O}$ to cross-docks $f \in \mathcal{F}$ and the second layer distribution costs is defined as the total costs of moving goods from cross-docks $f \in \mathcal{F}$ to respective destinations $d \in \mathcal{D}$. The third term is the cost of delays. Constraints (4.2) ensure that every shipment goes through exactly one of the cross-docks. In each cross-dock f , the inbound boxes are unloaded and loaded into one of the available trucks $\mathcal{W}_f$, according to constraints (4.3). Each truck has a limited capacity, see constraints (4.4). The outbound trucks deliver goods to no more than one consumer (constraints (4.5) and (4.6)). The time each truck is ready for departure is defined by constraints (4.7), whilst delivery delays for each destination are computed by constraints (4.8).

4.1.1 Valid inequalities

Consider the set of boxes, H_d^k , which are from a particular shipment $k \in \mathcal{K}$ demanded by $d \in \mathcal{D}$. If these boxes are sent to a cross-dock $f \in \mathcal{F}$, we can compute a lower bound on the cost of moving these boxes from f to their destinations. Let CAP^{max} be the

capacity of the largest truck in cross-dock $f \in \mathcal{F}$, then, we can define valid inequalities (4.9).

$$\sum_{w \in \mathcal{W}_f} c_{fd} v_{wd} + R \sum_{w \in \mathcal{W}_f} \sum_{d \in \mathcal{D}} e_{wd} \geq c_{fd} \left\lceil \frac{\sum_{i \in H_d^k} h_i}{CAP_{max}} \right\rceil \Delta_{kf}, \quad f \in \mathcal{F}, d \in \mathcal{D}, k \in \mathcal{K}. \quad (4.9)$$

Inequalities (4.9) represent the fact that if a shipment k is sent to facility f , $\Delta_{kf} = 1$, the second layer cost related to shipment k and facility f , $\sum_{w \in \mathcal{W}_f} c_{fd} v_{wd}$, is bounded by the number of trucks needed to move boxes H_d^k from f to their respective destination.

4.2 Decomposition method

The idea of a layered distribution can be explored to reformulate the NSCD model. In the described problem, we have two layers. For the first layer distribution, one needs to assign shipments to cross-docks. Then, the second layer distribution consists of the dispatching of trucks to destinations. Using this idea, we reformulate the problem by projecting into the space of Δ_{kf} variables with additional variables z_{fd} representing the second layer cost of sending the items from the cross-docks to the consumers. The reformulation is presented below:

$$(\text{NSCD-B-MP}) \text{ Min } \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}_o} \sum_{f \in \mathcal{F}} c_{of} \Delta_{kf} + \sum_{f \in \mathcal{F}} \sum_{d \in \mathcal{D}} z_{fd} \quad (4.10)$$

$$\sum_{f \in \mathcal{F}} \Delta_{kf} = 1, \quad k \in \mathcal{K}, \quad (4.2)$$

$$z_{fd} \geq q_{fd} - q_{fd} \sum_{k \in \mathcal{S}} (1 - \Delta_{kf}), \quad f \in \mathcal{F}, d \in \mathcal{D}, \mathcal{S} \subseteq \mathcal{K}. \quad (4.11)$$

Domain of the variables.

Objective function (4.10) minimises the total cost of first and second layer distribution and violating the deadlines. Constraints (4.2) assign the supplier shipments to the cross-docks. Given this assignment, it is possible to compute q_{fd} , the optimal second layer distribution cost from the cross-docks to the suppliers. Constraints (4.11) are exponential in number and ensure that the right second layer cost is imposed for each possible first layer assignment. Instead of explicitly enumerating constraints (4.11), we can use a logic-based Benders decomposition algorithm to their iterative generation.

The decomposition algorithm is summarised in Figure 15. At iteration $j = 1, \dots, J$, we solve the relaxed master problem (NSCD-B-MP with some of constraints (4.11)) which gives a distribution plan for the first layer of the system, namely, the set of shipments that are sent to a facility f ($\Delta_{kf} = 1$) with at least one box for destination d , $\mathcal{K}_{fd}^j$ (which also define sets $\mathcal{I}_{fd}^j$, the set of boxes sent to facility f for destination d). Then, for each

facility f , we have a sub-problem that we can further decompose by destination, since goods for different destination cannot share a vehicle. Therefore, we solve the sub-problem NSCD-B-Sub $_{fd}$ for each intermediate facility f and destination d .

Figure 15 – Scheme of the decomposition algorithm for the network scheduling problem. Solving a master problem determines the set of shipments $\mathcal{K}_{fd_1}, \dots, \mathcal{K}_{fd_{|\mathcal{D}|}}$ sent to facility $f \in \mathcal{F} = \{f_1, \dots, f_{|\mathcal{F}|}\}$ which is equivalent to defining the set of boxes $\mathcal{I}_{fd_1}, \dots, \mathcal{I}_{fd_{|\mathcal{D}|}}$. Set $\mathcal{I}_{fd}$ is the set of boxes sent to facility $f \in \mathcal{F}$ for destination $d \in \mathcal{D} = \{d_1, \dots, d_{|\mathcal{D}|}\}$.

$$(\text{NSCD-B-Sub}_{fd}) \quad q_{fd}^j = \text{Min} \quad \sum_{w \in \mathcal{W}_f} c_{fd} v_{wd} + R \sum_{w \in \mathcal{W}_f} \sum_{d \in \mathcal{D}} e_{wd}. \quad (4.12)$$

$$\sum_{i \in \mathcal{I}_{fd}} h_i p_{iw} \leq CAP_w v_{wd}, \quad w \in \mathcal{W}, \quad (4.13)$$

$$\sum_{w \in \mathcal{W}_f} p_{iw} = 1, \quad i \in \mathcal{I}_{fd}. \quad (4.14)$$

$$m_w \geq (LT_w + \bar{u}_{if}) p_{iw}, \quad w \in \mathcal{W}_f, i \in \mathcal{I}_{fd}, \quad (4.7)$$

$$m_w + t_{fd} - \tau_d - (1 - v_{wd})T \leq e_{wd}, \quad w \in \mathcal{W}_f, \quad (4.8)$$

Domain of the variables.

Objective function (4.12) accounts for the distribution cost from cross-docks to destination plus delays. Constraints (4.13) consider that loaded boxes do not violate the capacity of each truck. Moreover, we use the aforementioned constraints (4.7) and (4.8). After solving each of the sub-problems, we can generate cuts of the form (4.11).

As remarks note that (i) to ensure solution optimality of the decomposition method, we require an optimal solution of the sub-problem at least in the last step. However, the method can still provide feasible solutions when using a lower bound for the sub-problem; (ii) if we cannot find a feasible solution in reasonable time and the cut generated by the lower does not cut off the current solution, the method stops with a sub-optimal solution; (iii) finally, after solving the master problem, we can pre-process the time windows, required

for each sub-problem and eliminate the related variables and constraints *a priori*. The effectiveness of this strategy is discussed in Subsection 4.3.2.

4.2.1 Improving the cuts

The main drawback of constraints (4.11) is in the sense that they are only active for a single set of decisions Δ_{kf} . We can strengthen these constraints. Note that q_{fd} consists of the aggregated costs of each shipment used to load the truck going to a particular destination. Therefore, whenever the value of a variable Δ_{kf} is changed from one to zero, we want to discount from q_{fd} only the part of the cost associated with shipment k . For this, we can solve a sub-problem of distributing goods of shipment k from all cross-docks to all respective destinations, producing the cost (or an upper bound) M_{kfd} of distributing goods from k through f to d . Allowing us to replace (4.11) by (4.15) in NSCD-B-MP.

$$z_{fd} \geq q_{fd} - \sum_{k \in \mathcal{S}} M_{kfd}(1 - \Delta_{kf}), \quad f \in \mathcal{F}, d \in \mathcal{D}, \mathcal{S} \subseteq \mathcal{K}. \quad (4.15)$$

Note that bounds M_{kfd} can be pre-computed to be used when needed. We experimented with generating these bounds using the model for the sub-problems and a simple greedy heuristic which load boxes sorted in decreasing order of volume, disregarding the deadline for delivery (results are presented in Section 4.3). To use the model for the sub-problems, we define an instance with the boxes from shipment k that are demanded by d for each facility f . For the greedy procedure, we consider the same set of boxes but we disregard the deadlines and load the boxes into trucks in descending order of volume.

4.3 Computational experiments for the network scheduling problem

The goal of the experiments is to (i) gain insights on the behaviour of the NSCD model for instances with narrow and wide time windows, (ii) evaluate the effectiveness of the different variations of the decomposition approach against the NSCD model and (iii) the effect of valid inequalities (4.9). For this, we generated instances based on a benchmark for the classical Vehicle Routing Problem from Augerat *et al.*¹. We generated 27 instances according to the rules described in Subsection 4.3.1. For a comparison between the model NSCD and the decomposition approach with model NSCD-B-MP, we used Gurobi 7.5 with a time limit of 3600 seconds for each instance, considering 10 seconds for each sub-problem. The decomposition method was implemented using the *callback* feature (Optimization, 2018). Also, a time limit of 10 seconds is used in the pre-processing stage to compute the upper bounds M_{kfd} . When running the decomposition approach, we discount the pre-processing time from the time limit. We assume unitary penalty costs for

¹ Available on <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances>

violating a deadline $R = 1$. Regarding hardware settings, we used four cores of an Intel[®] Xeon CPU E5-2620 at a frequency of 2 GHz each. Finally, the optimality gap is reported as a difference between primal and dual bounds with respect to the primal bound.

We report results regarding the model (4.1)–(4.8), NSCD, and three variations of the decomposition method, which we call NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III. NSCD-B-MP-I corresponds the model with objective function (4.10) and constraints (4.2) and (4.11). Both NSCD-B-MP-II and III use the same model but with the improved cuts (4.15). The difference between them is in the method used to compute the factors M_{kfd} . NSCD-B-MP-II compute them using the model for the sub-problems, while NSCD-B-MP-III uses a greedy heuristic.

4.3.1 Generating instances

Since there are no test sets for the defined problem, we generated one as follows. The position of each node was taken from instances for the Capacitated Vehicle Routing Problem made available by Augerat *et al.*. The number of nodes varies from 32 to 76 across the instances. We randomly assign the status of supplier, cross-dock or destination for each node. We ensure there is at least one supplier, one destination and the number of cross-docks $f \in \mathcal{F}$ is set to 10% of the total number of nodes (rounding to the nearest integer), ensuring there are at least two cross-docks. The costs c_{ij} are defined as the euclidean distance between i and j and the time $t_{ij} = c_{ij}$ (i.e. we assume the speed of vehicles is unitary). To guarantee each supplier and consumer handle at least one box, we generate a random number of boxes in the interval $[\max\{|\mathcal{O}|, |\mathcal{D}|\}; 20 \max\{|\mathcal{O}|, |\mathcal{D}|\}]$. Therefore, we are considering that each consumer $d \in \mathcal{D}$ demand for 20 boxes, at most. The volume of each box, h_i , is an integer number also sampled randomly from the interval $[1, 5]$. We assume each truck with the same capacity $CAP_w = CAP = 20$, $w \in \mathcal{W}$. Also, we set all unloading, sorting and loading times, UT_k , ST_k , UT_w , to be the same and equal to the minimum distance between any pair of nodes in the network. For the client time deadlines τ_d , consider time windows defined as the interval $[\bar{\tau}_d^L, \bar{\tau}_d^H]$ in which $\bar{\tau}_d^L = \min_{i \in \mathcal{B}, f \in \mathcal{F}} \{\bar{u}_{if} + LT + c_{fd}\}$ and $\bar{\tau}_d^H = \max_{i \in \mathcal{B}, f \in \mathcal{F}} \{\bar{u}_{if} + LT + c_{fd}\}$. Then, we generate two sets of instances. The first one has *narrow* time windows. In this set, the centre of each consumer time window is randomly sampled from the interval $[0.1\bar{\tau}_d^L, 0.9\bar{\tau}_d^H]$, the width of each time window is also randomly sampled to have the width from 10 to 50% of the length of $[\bar{\tau}_d^L, \bar{\tau}_d^H]$. For the other set of instances (*wide*), all parameters remain the same with the exception of the width of each time window, which is sampled to be from 50 to 90% of the width of $[\bar{\tau}_d^L, \bar{\tau}_d^H]$. Finally, without loss of generality, each origin have a single shipment leaving with all the boxes assigned to that origin. The assignment of boxes to origins and destinations is random. All the random sampling aforementioned is from a uniform distribution in the respective interval. Note that, although we generate a time window for each client, we only use the upper limit of the window, as a deadline for delivery ($\tau_d = \bar{\tau}_d^H$). The intervals related to the time windows were defined empirically starting from the most

relaxed case, with time largest time relevant windows, then we tried to contemplate the two extremes of the interval. That is, time windows with the width of at least 50% of the maximum and with width of at most 50% of the maximum.

4.3.2 Comparing narrow and wide instances

We designed experiments to check whether the *wide* instance would be easier to solve for the NSCD model and for the decompositions. For the NSCD model, we simply run the set of 54 instances (27 *narrow* and 27 *wide*) with a time limit of 3600 seconds. The instances are paired in the sense that for each *narrow* instance, there is a *wide* instances with the only difference being the deadline for delivery. The solver could not find a feasible solution for one instance (*narrow* and *wide* cases). Regarding solution quality, the gap for the *wide* instance was never worse, being better in 21 cases. The average gap for the *narrow* instances was 39.39% and for the *wide* instances it was 35.44%. The number of optimal solutions found was five for the *narrow* case and six for *wide* cases. Moreover, the optimal was always found faster in the *wide* instances. For detailed results the reader is referred to Appendix A.

Regarding the decomposition approaches NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III, in each iteration, after solving the master problem, we can check if the sub-problem must consider the time windows or if it is possible to reach every destination without delay. In fact, this was our initial motivation to generate two sets of instances (*narrow* and *wide*), since we had a hypothesis that pre-processing this feature would improve the decomposition sensibly for the *wide* cases. However, this was not the case (see Table 15). Indeed, without the pre-processing of the time windows the methods, in some cases, was better for different variations of the decompositions (I, II or III) and for *narrow* and *wide* instance. In general, the difference was not expressive and, therefore, we decided to not include the time window pre-processing as part of the decomposition method. For detailed results the reader is referred to Appendix A.

For the detailed results, we performed a Wilcoxon signed-rank test for each of the decompositions variation to compare the case *with* pre-processing and *without* pre-processing. Regarding the primal bound for the *narrow* case, the highest p-value was 0.4%, for variation III of the decomposition. Therefore, we can reject the null hypotheses that both samples come from the same distribution. For the *wide* case, the highest p-value was 5.1%, for variation I of the decomposition. Regarding gap, the null hypothesis is harder too reject (for the *narrow* and *wide* case) since the lowest p-value considering narrow and wide instances was 13.6% (the highest was 34.9%). Finally, related to the time to solve the instances, we can reject all but one of the null hypothesis. The null hypothesis we cannot reject is related to variation II of the decomposition with a p-value of 61.4% for the *wide* instances, all the others had a p-value lower than 1.0%.

Table 15 – Summary of the results comparing primal bound, gap and time to solve (s) for the decompositions NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III for *narrow* and *wide* instances *with* and *without* time window pre-processing.

	NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III		
	With	Without	With	Without	With	Without	
Primal bound	5047.92	5032.35	4747.31	4687.59	4686.77	4561.36	Narrow
	4675.56	4640.53	4302.75	4371.99	4284.28	4287.07	Wide
Gap	54.66	54.61	21.12	20.65	22.86	23.45	Narrow
	53.10	53.09	18.27	18.36	20.37	21.23	Wide
Time to solve (s)	2913	2914	1876	1898	1931	1945	Narrow
	2864	2858	1821	1817	1842	1853	Wide

4.3.3 Evaluating decomposition approach

The decomposition approach provide better bounds, on average, than the default solver procedure (Table 16). Although the most *naïve* variation of the decomposition (NSCD-B-MP-I) is indeed less effective, on average, than the default solver with model NSCD related to primal and dual bounds. Regarding primal bounds, the best average was achieved by NSCD-B-MP-III. Regarding dual bounds, NSCD-B-MP-II provided the best values more frequently than the other approaches. Also the decompositions could find a feasible solution for instance **A-n63-k9** for which the solver could not, within the time limit of 3600 seconds.

To evaluate the statistical significance of the results, we performed a pair-wise Wilcoxon signed test to compare each combination of the methods for primal and dual bounds (see Table 17). The higher p-values show that there is only weak evidence (p-value greater than 10%) for different performance of algorithms NSCD-B-MP-II and III regarding primal bound when compared against NSCD. Although, there is strong evidence (p-value less than 5%) to support the improvement in dual bounds for the decomposition algorithm in all variations I, II and III.

The average difference between NSCD and the decompositions NSCD-B-MP-II and III is can bee seen when comparing gap and time to solve (Table 18). Moreover, we see that NSCD-B-MP-I actually produce higher gaps, on average, than all the alternatives even though the time to solve is competitive. Regarding gap and time to solve, the best option is NSCD-B-MP-II. The number of the proven optimal solutions by NSCD-B-MP-II is also higher (10) than NSCD (four), NSCD-B-MP-I (five) and NSCD-B-MP-III (nine). Statistical difference measured by the Wilcoxon signed-rank did also argue in favour of a difference in performance in general except for the NSCD and NSCD-B-MP-I regarding computational time (see Table 19).

4.3.4 Evaluating the valid inequalities

The results presented in subsections 4.3.2 and 4.3.3 do not incorporate the valid inequalities (4.9). We evaluate the effect of these valid inequalities by adding them in the

Table 16 – Comparison of best primal and dual bounds for the model (NSCD) and the three variations of the decomposition strategy NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III. NA represents that no solution was found within the time limit (3600s). Bold values indicate best primal and dual bounds for each instance.

Inst	NSCD		NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III	
	Primal	Dual	Primal	Dual	Primal	Dual	Primal	Dual
A-n32-k5	3794.20	2775.84	3947.23	1453.76	3965.68	1800.82	4024.87	1549.17
A-n33-k5	2800.54	1225.24	2723.88	1154.11	2973.65	1360.74	2973.65	1258.75
A-n33-k6	3547.39	2120.98	3547.39	3547.39	3547.39	3547.39	3547.39	3547.39
A-n34-k5	2940.39	2099.73	2940.39	1817.16	2940.39	2940.39	2940.39	2940.39
A-n36-k5	3273.17	1802.86	3412.40	678.04	3412.40	2567.49	3410.09	2325.34
A-n37-k5	2270.46	2270.46	2616.31	678.93	2270.46	2270.46	2270.46	2770.46
A-n37-k6	3077.35	2097.83	3156.42	697.88	3080.17	2681.90	3132.25	2368.29
A-n38-k5	1721.78	1721.78	1721.78	1183.90	1721.78	1721.78	1721.78	1721.78
A-n39-k5	4800.77	1583.29	4742.26	4742.26	4742.26	4702.90	4742.26	4722.82
A-n39-k6	3998.34	2660.90	4706.31	1025.98	3998.34	3998.34	3998.34	3998.34
A-n44-k6	4728.43	1458.72	4694.89	4491.13	4694.89	4491.13	4694.89	3848.87
A-n45-k6	6140.23	1671.98	6128.45	5846.54	6128.45	5827.54	6128.45	4896.63
A-n45-k7	2659.36	2659.36	2659.36	2659.36	2659.36	2659.36	2659.36	2659.36
A-n46-k7	2891.16	2099.27	2957.11	785.70	2860.40	2453.37	2812.97	2279.07
A-n48-k7	7400.74	2271.29	6674.33	778.23	6676.82	2238.74	6676.82	2080.50
A-n53-k7	7050.52	1976.97	8177.51	366.35	7590.15	4427.33	6978.83	4277.32
A-n54-k7	4043.69	3244.25	4043.69	4043.69	4043.69	4043.69	4043.69	4043.69
A-n55-k9	5784.16	2816.31	6990.96	1054.24	5777.38	5777.38	5777.38	5776.80
A-n60-k9	13547.31	2867.97	8123.59	1254.28	9461.29	2261.25	9225.94	1990.04
A-n61-k9	2404.64	2404.64	3256.68	801.79	2404.64	2404.64	2404.64	2404.64
A-n62-k8	3068.42	2739.79	3213.13	1084.43	3068.42	3068.11	3068.42	3068.11
A-n63-k10	2235.76	2235.76	2308.40	1558.17	2235.76	2235.54	2235.76	2235.54
A-n63-k9	NA	NA	8326.47	1219.00	8282.84	2276.12	7770.14	1932.43
A-n64-k9	7581.08	2316.02	8121.01	711.40	6951.84	3742.18	7113.77	3492.15
A-n65-k9	14679.93	2677.62	14340.45	435.95	12125.72	2837.42	9853.34	2394.36
A-n69-k9	5601.93	2461.49	8971.65	852.31	5579.52	5174.45	5579.52	4887.10
A-n80-k10	3371.30	2811.66	3371.30	3371.30	3371.30	3371.30	3371.30	3371.30
Average	4823.58	2272.00	5032.35	1788.64	4687.59	3217.84	4561.36	3049.65

Table 17 – P-values for the different pairs of solution methods (NSCD, NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III), regarding the results for primal and dual bounds.

Method x method	Primal	Dual
NSCD x I	0.0325	0.0388
NSCD x II	0.1301	0.0004
NSCD x III	0.1301	0.0031
I x II	0.0095	0.0000
I x III	0.0095	0.0001
II x III	0.0000	0.0255

Table 18 – Comparison of gap and time to solve for the model (NSCD) and the three variations of the decomposition strategy NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III. NA represents that no solution was found within the time limit (3600s). Gap is presented as $(100 \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}})$. Bold values indicate best primal and dual bounds for each instance. TL indicates the time limit was reached for that case.

Inst	NSCD		NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III	
	Gap	Time	Gap	Time	Gap	Time	Gap	Time
A-n32-k5	26.84	TL	63.17	TL	54.59	TL	61.51	TL
A-n33-k5	56.25	TL	57.63	TL	54.24	3462	57.67	TL
A-n33-k6	40.21	TL	0.00	2379	0.00	511	0.00	586
A-n34-k5	28.59	TL	38.20	TL	0.00	761	0.00	582
A-n36-k5	44.92	TL	80.13	TL	24.76	3596	31.81	3599
A-n37-k5	0.00	2287	74.05	TL	0.00	73	0.00	52
A-n37-k6	31.83	TL	77.89	TL	12.93	3598	24.39	TL
A-n38-k5	0.00	2	31.24	TL	0.00	4	0.00	5
A-n39-k5	67.02	TL	0.00	1229	0.83	646	0.41	518
A-n39-k6	33.45	TL	78.20	TL	0.00	2189	0.00	2454
A-n44-k6	69.15	TL	4.34	301	4.34	463	18.02	394
A-n45-k6	72.77	TL	4.60	1817	4.91	1142	20.10	1663
A-n45-k7	0.00	369	0.00	47	0.00	5	0.00	6
A-n46-k7	27.39	TL	73.43	TL	14.23	3597	18.98	3599
A-n48-k7	69.31	TL	88.34	TL	66.47	TL	68.84	TL
A-n53-k7	71.96	TL	95.52	TL	41.67	3562	38.71	TL
A-n54-k7	19.77	TL	0.00	316	0.00	20	0.00	19
A-n55-k9	51.31	TL	84.92	TL	0.00	1763	0.01	1873
A-n60-k9	78.83	TL	84.56	TL	76.10	TL	78.43	TL
A-n61-k9	0.00	76	75.38	TL	0.00	8	0.00	9
A-n62-k8	10.71	TL	66.25	TL	0.01	149	0.01	105
A-n63-k10	0.00	461	32.50	TL	0.01	520	0.01	550
A-n63-k9	NA	TL	85.36	TL	72.52	3596	75.13	TL
A-n64-k9	69.45	TL	91.24	TL	46.17	TL	50.91	TL
A-n65-k9	81.76	TL	96.96	TL	76.60	3588	75.70	TL
A-n69-k9	56.06	TL	90.50	TL	7.26	3577	12.41	3599
A-n80-k10	16.60	TL	0.00	590	0.00	24	0.00	28
Average	39.39	3052	54.61	2914	20.65	1898	23.45	1945

Table 19 – P-values for the different pairs of solution methods (NSCD, NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III), regarding the results for gap and time to solve.

Method x method	Gap	Time
NSCD x I	0.0626	0.4806
NSCD x III	0.0001	0.0001
NSCD x II	0.0003	0.0007
I x II	0.0000	0.0000
I x III	0.0003	0.0002
II x III	0.0056	0.0690

master problem of the two decompositions that performed better (NSCD-B-MP-II and III), since they provided the best results overall. Note that z_{fd} represent the cost of second layer distribution (distribution and delays, $z_{fd} = \sum_{w \in \mathcal{W}_f} c_{fd} v_{wd} + R \sum_{w \in \mathcal{W}_f} \sum_{d \in \mathcal{D}} e_{wd}$).

On average, there is an improvement in the primal bound and gap, with a small increase in computational time (see tables 20, 21 and 22). The addition of valid inequalities improved the primal bound, on average, 3.25% for NSCD-B-MP-II and 0.68% for NSCD-B-MP-III related to the respective solution without the inequalities. For NSCD-B-MP-II, in all but one of the cases (**A-n64-k9**), adding the inequalities provided solution at least as good as not using the inequalities. The same happened for NSCD-B-MP-III in all but two of the instances (**A-n63-k9**) and (**A-n64-k9**).

Table 20 – Evaluation of primal bound comparing the same model with valid inequalities (**cuts**) for decompositions NSCD-B-MP-II and NSCD-B-MP-III. Bold values represent which of the alternatives was better for both decompositions.

Primal bound Inst	NSCD-B-MP-II		NSCD-B-MP-III	
	Original	Cuts	Original	Cuts
A-n32-k5	3965.68	3958.99	4024.87	4016.16
A-n33-k5	2973.65	2926.34	2973.65	2696.88
A-n33-k6	3547.39	3547.39	3547.39	3547.39
A-n34-k5	2940.39	2940.39	2940.39	2940.39
A-n36-k5	3412.40	3273.17	3410.09	3273.17
A-n37-k5	2270.46	2270.46	2270.46	2270.46
A-n37-k6	3080.17	3080.17	3132.25	3080.17
A-n38-k5	1721.78	1721.78	1721.78	1721.78
A-n39-k5	4742.26	4742.26	4742.26	4742.26
A-n39-k6	3998.34	3998.34	3998.34	3998.34
A-n44-k6	4694.89	4694.89	4694.89	4694.89
A-n45-k6	6128.45	6128.45	6128.45	6128.45
A-n45-k7	2659.36	2659.36	2659.36	2659.36
A-n46-k7	2860.40	2812.97	2812.97	2812.97
A-n48-k7	6676.82	6495.12	6676.82	6583.78
A-n53-k7	7590.15	6978.83	6978.83	6978.83
A-n54-k7	4043.69	4043.69	4043.69	4043.69
A-n55-k9	5777.38	5777.38	5777.38	5777.38
A-n60-k9	9461.29	8845.70	9225.94	8612.85
A-n61-k9	2404.64	2404.64	2404.64	2404.64
A-n62-k8	3068.42	3068.42	3068.42	3068.42
A-n63-k10	2235.76	2235.76	2235.76	2235.76
A-n63-k9	8282.84	7601.68	7770.14	7780.64
A-n64-k9	6951.84	7453.21	7113.77	7458.66
A-n65-k9	12125.72	9845.28	9853.34	9845.28
A-n69-k9	5579.52	5579.52	5579.52	5579.52
A-n80-k10	3371.30	3371.30	3371.30	3371.30
Average	4687.59	4535.39	4561.36	4530.50

The optimality gap is smaller when using the valid inequalities for both NSCD-B-MP-II and III on average (see Table 21). Although there are some cases in which the gap indeed increased, the average gap reduced 2.35% and 3.12% respectively for NSCD-B-MP-II and III. The number of instances for which the optimality could be proven increased by one for NSCD-B-MP-II but reduced by one in NSCD-B-MP-III.

Table 21 – Evaluation of gap ($100 \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}}$) comparing the decompositions NSCD-B-MP-II and NSCD-B-MP-III with valid inequalities (**cuts**) decompositions. Bold values represent which of the alternatives was better for both decompositions.

Gap Inst	NSCD-B-MP-II		NSCD-B-MP-III	
	Original	Cuts	Original	Cuts
A-n32-k5	54.59	53.54	61.51	56.13
A-n33-k5	54.24	50.69	57.67	48.07
A-n33-k6	0.00	0.00	0.00	0.00
A-n34-k5	0.00	0.00	0.00	0.00
A-n36-k5	24.76	0.00	31.81	11.27
A-n37-k5	0.00	0.00	0.00	0.00
A-n37-k6	12.93	18.97	24.39	21.83
A-n38-k5	0.00	0.00	0.00	0.00
A-n39-k5	0.83	0.95	0.41	0.95
A-n39-k6	0.00	0.00	0.00	0.01
A-n44-k6	4.34	4.67	18.02	4.36
A-n45-k6	4.91	5.25	20.10	5.36
A-n45-k7	0.00	0.00	0.00	0.00
A-n46-k7	14.23	9.54	18.98	14.17
A-n48-k7	66.47	54.75	68.84	58.23
A-n53-k7	41.67	34.07	38.71	48.84
A-n54-k7	0.00	0.00	0.00	0.00
A-n55-k9	0.00	0.01	0.01	9.07
A-n60-k9	76.10	64.12	78.43	63.47
A-n61-k9	0.00	0.00	0.00	0.00
A-n62-k8	0.01	0.00	0.01	0.01
A-n63-k10	0.01	0.01	0.01	0.01
A-n63-k9	72.52	61.45	75.13	63.77
A-n64-k9	46.17	59.58	50.91	57.84
A-n65-k9	76.60	53.04	75.70	58.96
A-n69-k9	7.26	23.53	12.41	26.57
A-n80-k10	0.00	0.00	0.00	0.00
Average	20.65	18.30	23.45	20.33

Although the optimality gaps were smaller on average, the average computational time increased by two seconds for NSCD-B-MP-II and 38 seconds for NSCD-B-MP-III. In seven instances the addition of the valid inequalities lead to an increase in computational time for NSCD-B-MP-II. The same phenomenon was observed four times in NSCD-B-MP-III. Some reductions in computational time can also be observed, therefore, it is hard to state a dominance favouring either the addition or not of the inequalities regarding computational time.

Regarding the statistical significance of these results, a summary of the p-values for a Wilcoxon signed-rank test are presented in Table 23. The primal bound for NSCD-B-MP-II is the one that presented a stronger evidence (p-value of 1.3%) against the null hypothesis. For the other cases, the evidence against the null hypotheses that both series come from the same distribution is weaker.

Table 22 – Evaluation of time to solve (s) the **original** model compared to the same model with valid inequalities (**cuts**) decompositions NSCD-B-MP-II and NSCD-B-MP-III. Bold values represent which of the alternatives was better for both decompositions. TL indicates that the time limit was reached for that case.

Time	NSCD-B-MP-II		NSCD-B-MP-III	
	Original	Cuts	Original	Cuts
A-n32-k5	TL	TL	TL	TL
A-n33-k5	3462	3482	TL	TL
A-n33-k6	511	390	586	422
A-n34-k5	761	316	582	396
A-n36-k5	3596	2772	3599	TL
A-n37-k5	73	32	52	34
A-n37-k6	3598	TL	TL	TL
A-n38-k5	4	3	5	3
A-n39-k5	646	471	518	458
A-n39-k6	2189	2712	2454	3074
A-n44-k6	463	267	394	267
A-n45-k6	1142	874	1663	1213
A-n45-k7	5	5	6	5
A-n46-k7	3597	3597	3599	3599
A-n48-k7	TL	TL	TL	TL
A-n53-k7	3562	3591	TL	TL
A-n54-k7	20	12	19	12
A-n55-k9	1763	3277	1873	TL
A-n60-k9	TL	3583	TL	TL
A-n61-k9	8	6	9	7
A-n62-k8	149	79	105	104
A-n63-k10	520	646	550	744
A-n63-k9	3596	3597	TL	TL
A-n64-k9	TL	3592	TL	TL
A-n65-k9	3588	TL	TL	TL
A-n69-k9	3577	3580	TL	3599
A-n80-k10	24	12	28	15
Average	1898	1900	1945	1983

Table 23 – P-values for the Wilcoxon signed-rank test comparing the case without (*original*) and (*with*) cuts for NSCD-B-MP-II and NSCD-B-MP-III for primal bound, gap and time to solve.

	NSCD-B-MP-II	NSCD-B-MP-III
Primal bound	0.0126	0.1036
Gap	0.3227	0.1165
Time	0.2628	0.0750

4.4 Remarks for the network scheduling problem

Load consolidation is essential not only for reducing costs in transportation networks but also to reduce some issues in large urban centres such as pollution and congestion. Cross-docking strategies are one of the main tools for increasing load consolidation. Nevertheless, they are rarely studied within the broader distribution networks to which they usually belong.

We propose a mixed integer linear programming model for a layered distribution network with multiple cross-docking facilities. We also propose a logic-based Benders decomposition strategy that explores the layered topology of the network. The decomposition strategy outperforms the solutions obtained by applying a commercial solver to the proposed MIP model, both in terms of solution quality and computational effort. Both model and decomposition strategy can be extended and adapted to handle other practical scenarios. For example, the cross-docking facilities might have a limited processing capability, either due to physical space or workforce available. The decomposition approach, in particular, can be easily adapted to contemplate routing decisions for trucks leaving the cross-docks or any other second layer constraints, as these are relegated to the sub-problems. This would imply in harder-to-solve sub-problems which is a drawback that can be tackled with the use of heuristic approaches. Indeed, our proposed framework, when tested with the proposed standard case, was not heavily affected by the choice of algorithm (exact or heuristic) to solve the sub-problems.

DESIGN OF HUB EXCHANGE NETWORKS

Some city logistics initiatives around the world have tried to implement multi-layered distribution systems, in which goods coming from outside the city are delivered to Urban Distribution Centres. We studied one possible architecture for these layered systems in Chapter 4. In this chapter, we propose an alternative approach. Instead of having dedicated facilities used as cross-docks, we design networks which use the currently available infrastructure to allow suppliers to exchange goods between themselves before distributing them. We begin the chapter by giving an overview of how distribution systems networks can be designed to provide the context of our proposal.

Physical distribution systems can be structured using different types of networks for transporting goods from a distribution centre to intermediate stops (terminals), between terminals, and between terminals and demand points (Daganzo, 2005). Some networks can be developed where goods are transferred from depots to terminals and separate routes developed from terminals (Daganzo and Newell, 1986). In such networks, levels of routes are defined, the higher level used to transport goods between the depot and terminals and the lower level routes serving customers from the terminals. Recently, there has been increased interest in applications of the sharing economy in city logistics that involves sharing logistics facilities and freight vehicles with other suppliers (or shippers) (Savelsbergh and Woensel, 2016a). Swapping orders between shippers (horizontal and enhanced cooperation) in regional distribution can substantially reduce distances travelled by vehicles (Fischer *et al.*, 1993). In urban delivery systems, for example, two common types of distribution are hub and spoke networks (Button and Stough, 2000) and pickup and delivery problems (Savelsbergh and Sol; Sigurd, Pisinger and Sig; Ropke and Pisinger, 1995, 2000, 2006).

In city logistics, urban consolidation centres can assist in rationalising the transport and storage of goods, providing considerable social, economic and environmental cost savings (Taniguchi and Thompson, 2015). Goods are typically stored and sorted for distribution at a common location. They can reduce distribution costs and improve the reliability of deliveries. However, urban consolidation centres are expensive to establish,

difficult to manage and have been difficult to maintain after public subsidies are withdrawn (Duin and Muñuzuri, 2015).

Here, in this chapter, we explore how an existing metropolitan distribution network operating with independent delivery systems from suppliers can be transformed into a collaborative network to reduce the aggregated distance travelled by vehicles. In this type of network, suppliers share existing storage space at their warehouses with other suppliers as well as delivery vehicles. This sharing strategy allow for an opportunistic consolidation. That is, besides consolidating goods for its own distribution, each consumer can take advantage of the collaborative space to further increase consolidation rates. We propose a model the design problem of a collaborative distribution network to transform independent retail distribution chains into collaborative networks which can reduce the financial and environmental cost of distribution in metropolitan regions (Section 5.1). We also proposed an heuristic based on the integer programming model (Section 5.2). Computational results are presented in Section 5.3 and we draw some conclusions in Section 5.4

5.1 Collaborative hub exchange network

We are interested in networks consisting in a set of suppliers, each with its own depot, and a set of consumers, each demanding goods from different suppliers. A non-collaborative approach to satisfy the demands would solve a capacitated vehicle routing problem for each supplier (see Figure 16). This may lead to trucks performing similar routes and consumers receiving multiple visits.

Figure 16 – Example of distribution without collaboration. Each large node (red square, blue star, green circle) represents a supplier, providing the associated type of good. Each cell represents a consumer which has demand from the type of items, indicated by the drawing inside the cell. In this non-collaborative case, each supplier is responsible for routing its own deliveries.

In this setting, we propose the following collaborative system. One of the suppliers is elected as a hub. This hub can act as a centre for exchanging goods between suppliers. After some exchanges happen, a single delivery route satisfy consumer demands from products of different suppliers (see Figure 17). Comparing to the non-collaborative case,

this potentially reduces the number of visits to each consumer and minimises total travel distance, which has effects on distribution costs, traffic and pollution.

Figure 17 – Example with collaboration with supplier hub being the square node. Each large node (red square, blue star, green circle) represents a supplier. Each cell represents a consumer which has demand from the type of items indicated by the drawing inside the cell. In this case, after some exchanges happen with the hub (indicated by the dotted black two-way arrows), a supplier can satisfy demands for different sets of consumers.

More formally, we present the problem of hub selection and routing as a collaborative distribution problem in a two-level network. In the first level, the model selects the hub, the transfers between suppliers and hub and respective inventory levels. The inventory variables act as a link between the first level and the distribution network, which decides on the delivery routes reaching the customers.

Table 24 – Summary of the parameters for the hub selection problem.

$\mathcal{O}$:	Set of origins of the distribution (suppliers in the network),
$\mathcal{D}$:	Set of destinations (consumers in the network),
$\mathcal{N}$:	Node in the network, $\mathcal{N} = \mathcal{O} \cup \mathcal{D}$,
$\mathcal{C}$:	Set of types of goods,
$\mathcal{T}_e$:	Set of exchange time periods,
$\mathcal{T}_r$:	Set of routing time periods,
$\mathcal{T}$:	Set of time periods in the planning horizon, $\mathcal{T} = \mathcal{T}_e \cup \mathcal{T}_r$,
c_{ij} :	Travel cost (distance) between points (nodes) i and j in the distribution network,
d_{ic} :	Demand from consumer $i \in \mathcal{D}$ for type of good $c \in \mathcal{C}$,
$\mathcal{V}$:	Set of vehicles,
$v(i)$:	Vehicle associated with supplier $i \in \mathcal{O}$,
CAP_v :	Capacity of vehicle $v \in \mathcal{V}$.

Let $\mathcal{N} = \mathcal{O} \cup \mathcal{D}$ be the set of nodes in the network. The set of nodes is partitioned into suppliers ($\mathcal{O}$) and consumers ($\mathcal{D}$). Let c_{ij} be the cost (distance) of travelling from $i \in \mathcal{N}$ to $j \in \mathcal{N}$. For each consumer node $i \in \mathcal{D}$, there is an associated demand for a particular product $c \in \mathcal{C}$, d_{ic} . Each supplier $i \in \mathcal{O}$ has a vehicle $w(i)$ with capacity $CAP_{w(i)}$ that can be used either for exchanging product with another supplier, via the selected hub, or for visiting a consumer. Each vehicle may perform at most one trip per time period $t \in \mathcal{T}$. The periods in the planning horizon, $\mathcal{T}$, are partitioned into *exchange periods* ($\mathcal{T}_e$) and *routing periods* ($\mathcal{T}_r$). In the set of exchange periods, $\mathcal{T}_e$, only visits to (or from) the

hub are allowed, in the set of routing periods, $\mathcal{T}_r$, only visits to the consumers are allowed. In summary, we have the parameters listed in Table 24.

The goal is to select routes and exchanges with the hub that minimise the aggregated distance (cost) of distribution and exchanges. For this, let variables $y_i \in \{0, 1\}$, $i \in \mathcal{O}$, indicate whether supplier i is selected as a hub. Variables $\delta_{ijvt} \in \{0, 1\}$ indicate whether vehicle $v \in \mathcal{V}$ goes from node $i \in \mathcal{N}$ to $j \in \mathcal{N}$ at routing period $t \in \mathcal{T}_r$. For the exchange periods, we have binary variables σ_{ijt} indicating whether a vehicle goes from node $i \in \mathcal{O}$ to $j \in \mathcal{O}$ in time period $t \in \mathcal{T}_e$. We also have binary variables ψ_{it} indicating whether consumer $i \in \mathcal{D}$ is visited at time period $t \in \mathcal{T}_r$. For the continuous variables, the amount of product $c \in \mathcal{C}$ send from (to) a supplier $i \in \mathcal{O}$ to (from) the hub $j \in \mathcal{O}$ at time period $t \in \mathcal{T}_e$ is defined by variables λ_{ijct} . Variables g_{ijct} give the amount of product $c \in \mathcal{C}$ transported in a routing period $t \in \mathcal{T}_r$ from node $i \in \mathcal{N}$ to $j \in \mathcal{N}$. Finally, variables θ_{ict} provide the level of stock at supplier $i \in \mathcal{O}$ of product $c \in \mathcal{C}$ at time period $t \in \mathcal{T}$. In summary, we have the decision variables in Table 25 and the model formulation in (5.1)–(5.14).

Table 25 – Summary of variables used in modelling the hub selection problem.

y_i :	Binary variables indicating whether supplier $i \in \mathcal{O}$ is selected as a hub,
ψ_{it} :	Binary variables indicating whether consumer $i \in \mathcal{D}$ is visited in time period $t \in \mathcal{T}_r$,
δ_{ijvt} :	Binary variables indicating whether vehicle $v \in \mathcal{V}$ goes from $i \in \mathcal{N}$ to $j \in \mathcal{N}$ in time period $t \in \mathcal{T}_r$,
σ_{ijt} :	Binary variables indicating whether a vehicle goes from $i \in \mathcal{O}$ to $j \in \mathcal{O}$ in time period $\mathcal{T}_e$,
λ_{ijct} :	Amount of product $c \in \mathcal{C}$ moved from $i \in \mathcal{O}$ to $j \in \mathcal{O}$ in time period $t \in \mathcal{T}_e$,
g_{ijct} :	Amount of product $c \in \mathcal{C}$ moved from $i \in \mathcal{D}$ to $j \in \mathcal{D}$ in time period $t \in \mathcal{T}_r$,
θ_{ict} :	Inventory level of product $c \in \mathcal{C}$ in supplier $i \in \mathcal{D}$ in time period $t \in \mathcal{T}$.

$$\text{Min} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} \sum_{v \in \mathcal{V}} \sum_{t \in \mathcal{T}_r} c_{ij} \delta_{ijvt} + \sum_{i \in \mathcal{O}} \sum_{j \in \mathcal{O}} \sum_{t \in \mathcal{T}_e} c_{ij} \sigma_{ijt}, \quad (5.1)$$

subject to:

$$\sum_{c \in \mathcal{C}} \lambda_{ijct} \leq \max\{CAP_{v(i)}, CAP_{v(j)}\} \sigma_{ijt}, \quad i \in \mathcal{O}, j \in \mathcal{O}, t \in \mathcal{T}_e, \quad (5.2)$$

$$\sum_{i \in \mathcal{O}} y_i = 1, \quad (5.3)$$

$$\sum_{c \in \mathcal{C}} \lambda_{jict} \leq CAP_{v(j)} y_i + CAP_{v(i)} y_j, \quad i \in \mathcal{O}, j \in \mathcal{O}, t \in \mathcal{T}_e, \quad (5.4)$$

$$\theta_{ic,t-1} + \sum_{j \in \mathcal{O}} (\lambda_{jict} - \lambda_{ijct}) = \theta_{ict}, \quad i \in \mathcal{O}, c \in \mathcal{C}, t \in \mathcal{T}_e, \quad (5.5)$$

$$\theta_{ic,t-1} - \sum_{j \in \mathcal{D}} g_{ijct} = \theta_{ict}, \quad i \in \mathcal{O}, c \in \mathcal{C}, t \in \mathcal{T}_r, \quad (5.6)$$

$$\sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{V}} \sigma_{ijvt} = \psi_{jt}, \quad j \in \mathcal{C}, t \in \mathcal{T}_r, \quad (5.7)$$

$$\sum_{t \in \mathcal{T}_r} \psi_{it} = 1, \quad i \in \mathcal{D}, \quad (5.8)$$

$$\sum_{j \in \mathcal{D}} \delta_{ijv(i)t} \leq 1, \quad i \in \mathcal{C}, t \in \mathcal{T}_r, \quad (5.9)$$

$$\sum_{j \in \mathcal{D}} \delta_{ijv(j)t} = 0, \quad i \in \mathcal{O}, t \in \mathcal{T}_r, \quad (5.10)$$

$$\sum_{j \in \mathcal{N}} \sum_{c \in \mathcal{C}} g_{ijct} \leq C_{v(i)}, \quad i \in \mathcal{O}, t \in \mathcal{T}_r, \quad (5.11)$$

$$g_{ijct} \leq \max_{v \in \mathcal{V}} \{CAP_v\} \sum_{v \in \mathcal{V}} \delta_{ijvt}, \quad i \in \mathcal{N}, j \in \mathcal{N}, t \in \mathcal{T}_r, c \in \mathcal{C}, \quad (5.12)$$

$$\sum_{j \in \mathcal{N}} (\delta_{ijvt} - \delta_{jivt}) = 0, \quad i \in \mathcal{N}, v \in \mathcal{V}, t \in \mathcal{T}_r, \quad (5.13)$$

$$\sum_{j \in \mathcal{N}} (g_{jict} - g_{ijct}) = d_{ic} \psi_{it}, \quad i \in \mathcal{D}, c \in \mathcal{C}, t \in \mathcal{T}_r, \quad (5.14)$$

Domain of the variables.

The objective function (5.1) minimises routing and exchange distances. Constraints (5.2) define the exchange routes. Constraints (5.3) determine that a single hub is selected, while constraints (5.4) limit the flow of products between suppliers to the capacity of the vehicle being used. Constraints (5.5) and (5.6) manage the stock of each supplier along the time horizon. They act as linking constraints between the exchange phase and the routing phase. Constraints (5.7) define the visit of a customer. To ensure every customer is visited, we have constraints (5.8). Constraints (5.9) and (5.10) guarantee that only the vehicle associated with supplier i , $v(i)$, departs from supplier i . Constraints (5.11) limit the flow of products by the capacity of the truck used. Constraints (5.12) define the distribution routes. Constraints (5.13) account for the continuity of the routes. Finally, demand satisfaction is ensured by constraints (5.14).

5.2 A heuristic method: Relax-and-fix and proximity search

The relax-and-fix heuristic (Pochet and Wolsey, 2006) is a constructive heuristic, which builds a solution using a series of relaxations of the model. It has been applied to different problems, some recent example are: Scheduling (Kroer *et al.*, 2018), three-dimensional packing (Paquay *et al.*, 2018b) and production planning (Aouam *et al.*; Akartunali and Miller, 2018, 2009). As mentioned in some of these references, one challenge of using the relax-and-fix framework is that there is no guarantee that it will provide a feasible solution. To handle this, we added a feasibility repair step. Using proximity search (Fischetti and Monaci, 2016), allowed us to search for feasible integer solutions close to a relaxed feasible solution provided by the relax-and-fix. Examples of application for proximity search can be seen in (Boland *et al.*; Fischetti and Monaci, 2016, 2016).

We applied the relax-and-fix framework to the problem as follows. For each $t \in \mathcal{T}_r$, we partitioned variables δ_{ijvt} and ψ_{it} . At a current $t = \bar{t}$, we relax the integrality constraints for the set of variables mentioned for all $t > \bar{t}$. Then, after the optimisation, we fix variables related to $t = \bar{t}$ and repeat the process for the next time period in $\mathcal{T}_r$. By means of example, consider $\mathcal{T} = \{1, 2, 3, 4, 5\}$ with $\mathcal{T}_r = \{2, 4, 5\}$ and $\mathcal{T}_e = \{1, 3\}$. In the first iteration, $t = 2$ (the first routing period), we have variables δ_{ijv2} and ψ_{i2} as binary, others with the binary constraints relaxed. After solving this relaxed instance, we fix variables δ_{ijv2} and ψ_{i2} and repeat the process for $t = 4$. Figure 18 provides a graphical representation of the algorithm. Note that we could include variables σ_{ijt} in the procedure, however, initial results suggested that this was less efficient.

Figure 18 – Graphical representation of the relax-and-fix strategy. At each iteration $1, \dots, |T|$, only the correspondent subset of variables are considered binary. Others are either fixed or relaxed.

5.2.1 Repairing feasibility

The relax-and-fix algorithm proceeds in a rolling horizon fashion without ever revisiting its previous decisions. This may lead to an infeasible solution even for viable instances due to the fixing of the variables in previous steps. Whenever an infeasibility is encountered, we activate a Proximity Search procedure: We solve a model with constraints (5.2)–(5.14) and objective function (5.15). In other words, we look for a solution that is the closest (Hamming distance) from a relaxed solution (given by the relax-and-fix).

$$\text{Min} \sum_{\delta_{ijvt} \in \mathcal{F}_\delta^-} \delta_{ijvt} + \sum_{\delta_{ijvt} \in \mathcal{F}_\delta^+} (1 - \delta_{ijvt}) + \sum_{\psi_{it} \in \mathcal{F}_\psi^-} \psi_{it} + \sum_{\psi_{it} \in \mathcal{F}_\psi^+} (1 - \psi_{it}). \quad (5.15)$$

In which $\mathcal{F}_\delta^-$ ($\mathcal{F}_\delta^+$) is the set of variables δ_{ijvt} which had a value of zero (one) in the previous (feasible) step. In a similar fashion, we have the sets $\mathcal{F}_\psi^-$ and $\mathcal{F}_\psi^+$ for the ψ_{it} variables.

5.3 Computational experiments for the design of hub exchange networks

The goal of the experiments is to (i) evaluate potential savings when using the collaborative approach against the non-collaborative one, (ii) gain insights on the effectiveness of the relax-and-fix in achieving high quality solutions compared to the model and (iii) understand how the trucks capacity might affect solution quality. For this, we generated a set of instances based on instances for the traditional Vehicle Routing Problem. We begin by describing this set of instances and the computational environment of the experiments. Then, Subsection 5.3.1 presents results comparing solutions from the model and the relax-and-fix to the non-collaborative case. Subsection 5.3.2 analyse the effect of changing the capacity of the trucks and, in Subsection 5.3.3, we present a case study based on a retailer in Melbourne area (Australia).

We generated a set of random instances based on the benchmark for a capacitated vehicle routing problem published by Augerat *et al.*¹ For each generated instance, we randomly chose half (rounded up) of the nodes in a instance from Augerat *et al.* The number of suppliers was set to 20% (rounded up) of the network, except when this lead to just one supplier. In such cases, we used two suppliers. Suppliers and consumers were chosen randomly. Truck capacity was set to $\lceil \frac{|C||D|}{T_r} \rceil$ items. We assumed that each consumer demands one item from each supplier. Also, we consider the planning horizon $\mathcal{T} = \{1, 2, 3, 4, 5\}$ (five days of a week), with $\mathcal{T}_e = \{1, 3\}$ and $\mathcal{T}_r = \{2, 4, 5\}$. Moreover, each supplier provides its own unique product. For all the experiments, we used Gurobi 7.0 in Ubuntu 12.04. Regarding hardware settings, we used four cores of an Intel[®] Xeon CPU E5-2620 at a frequency of 2 GHz each.

First, in Subsection 5.3.1, we present general results comparing the non-collaborative approach, the default branch-and-cut from Gurobi and the relax-and-fix method. For the non-collaborative approach, we assume every supplier is responsible for satisfying its own demand. For that, we solve a “classical” vehicle routing model for each supplier (using the MTZ subtour elimination constraints (Miller, Tucker and Zemlin, 1960), with a time limit of 3600 seconds each). For the branch-and-cut results, we use a time limit of 3600 for each instance and for the relax-and-fix we set the time limit for each subproblem as $\frac{3600}{|\mathcal{T}_r|}$. After the general results, in Subsection 5.3.2, we provide some insight on how different instance characteristics might affect the solution. Particularly, we show some results that suggest that a network in which products are interchangeable could be easier to solve and that using trucks with a bigger capacity could lead to a higher level of inventory at the hub facility. Finally, Subsection 5.3.3 presents a case study based on an Australian hardware retailer network. For the case study, we also compare non-collaborative and collaborative approaches (branch-and-cut and relax-and-fix).

¹ Available online on neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances

5.3.1 Evaluating model and heuristic

The collaborative approach yields better results even for suboptimal solutions (tables 26, 27 and 28). Although the non-collaborative strategy resulted in instances which were more easily solved (we could find an optimal solution for almost all cases), the collaborative case could achieve solutions with roughly one-third (one-fourth), on average, of the distance when using pure branch-and-cut (relax-and-fix). Note that, for the majority of cases, the time limit was reached for the branch-and-cut, thus we do not have proof of optimality. A Wilcoxon signed-rank test computed a p-value of zero when comparing solution quality of the non-collaborative against either of the collaborative approaches (branch-and-cut and relax-and-fix).

The relax-and-fix heuristics is more efficient, on average, than the pure branch-and-cut regarding solution quality and time (tables 26, 27 and 28). Using around one-third of the time, the heuristics provide results with an 80% reduction, on average, of the distance compared to the branch-and-cut solution. However, the relax-and-fix was not always the best alternative. For instance 57, for example, the solution without the relax-and-fix was about 53% of the value obtained when using the heuristics. Also, the branch-and-cut solver provided better solutions more frequently than the relax-and-fix. Moreover, a Wilcoxon signed-rank test does not support difference in solution quality when comparing pure branch-and-cut and the relax-and-fix alternative (p-value of 81.01%).

5.3.2 Insights on different scenarios

One point of stress in the network might be the warehousing capacity of the hub facility. To analyse this, we run the same set of instances with the same experimental settings but with trucks having their capacities doubled. The results suggest that the greater the capacity of the trucks the higher the volume utilisation at the hub facility. On average, the maximum inventory levels, along the planning horizon, for the doubled capacity case were 14% higher than the baseline case. Also, as expected, the total distance is reduced (by around 85%, on average) when using trucks with doubled capacity. Regarding solver performance, doubled capacities made the problem easier to solve, with an optimality gap of 88% of the baseline case, on average. The optimality gap is the difference between best upper and lower bounds related to the upper bound found within the time limit.

Another characteristic of the distribution that might affect the solutions is the mix of products supplied. In Subsection 5.3.1, we supposed that each supplier has its own unique type of product. If all suppliers have the same type of product, the total distance is about 35%, on average, of the distance travelled in the case with a more diverse mix of products. From the solver's performance point of view, it is also easier to solve the problem with just one type of product – the optimality gap is about 33% of the case with more variety of products. Also, in the case with just one product type, collaboration is often not required for a good solution. This suggests that a multi-commodity environments may

Table 26 – Results comparing non-collaborative, collaborative with branch-and-cut and collaborative with relax-and-fix approaches (part 1). For each approach, we report the solution value and time to reach that solution (TL if the time limit was reached). For the branch-and-cut we also report the best lower bound. Bold values represent the best solution between collaborative model for branch-and-cut with and without relax-and-fix.

	Non-collaborative		Branch-and-cut			Branch-and-cut with Relax-and-fix	
	Distance	Time (s)	Distance	Lower bound	Time (s)	Distance	Time (s)
1	1373.43	< 1	759.00	563.48	TL	760.34	1202
2	1188.17	1	549.22	379.95	TL	605.60	1201
3	1256.15	1	559.47	415.29	TL	646.73	1201
4	1374.51	1	580.52	440.38	TL	629.75	1201
5	1269.24	1	602.03	390.11	TL	653.47	1202
6	1384.38	1	601.21	486.86	TL	630.94	1202
7	1203.77	2	606.09	482.27	TL	629.36	1202
8	1207.32	1	570.28	378.84	TL	595.19	1202
9	1470.14	1	713.71	461.27	TL	739.78	1202
10	1353.15	2	650.92	457.47	TL	690.74	1202
11	1541.06	44	673.55	393.28	TL	795.76	1205
12	1913.17	1762	810.96	481.79	TL	915.05	1205
13	1699.02	1	629.73	382.31	TL	611.58	1205
14	1571.92	1	674.48	374.68	TL	668.99	1205
15	2036.45	2	750.64	436.57	TL	777.97	1206
16	2497.30	8	775.01	410.37	TL	749.97	1212
17	2024.72	6	895.33	330.29	TL	727.82	1212
18	2083.96	14	1895.63	347.09	TL	870.16	1215
19	2143.19	1016	1017.71	389.88	TL	1024.02	1232
20	2202.43	6	958.81	356.68	TL	996.45	1244
21	2261.67	10	993.55	368.71	TL	1040.26	1228
22	2320.91	655	1041.04	395.07	TL	832.66	1246
23	2380.15	16	941.86	427.04	TL	1021.05	1237
24	2439.38	7	948.66	433.35	TL	848.10	1241
25	2498.62	35	1323.92	406.97	TL	1227.17	1231

benefit more from collaboration.

5.3.3 Case study: Melbourne hardware retailer

We used the branch-and-cut (with model in Section 5.1) and the relax-and-fix method (Section 5.2) to investigate potential savings when using collaboration on a network associated with the distribution of goods from hardware retailers in the metropolitan area of Melbourne, Australia. This involved the distribution of goods from five suppliers to 41 retail stores. Grouping retailers into local areas (Figure 19 and Table 29), we have the estimated weekly demands (Table 30), which were based on the population, demographics and types of goods shipped from each supplier. Each supplier is assumed to be operating a medium sized truck with a payload capacity of 10,000kg.

For the non-collaborative approach, each supplier distribute its own demand. For that, we solved a vehicle routing problem to optimality, for each retailer, and the total aggregated distance was 1466. The results for the collaborative approach are shown in Table 31. Also, when using pure branch-and-cut, the first solution was found after roughly

Table 27 – Results comparing non-collaborative, collaborative with branch-and-cut and collaborative with relax-and-fix approaches (part 2). For each approach, we report the solution value and time to reach that solution (TL if the time limit was reached). For the branch-and-cut we also report the best lower bound. Bold values represent the best solution between collaborative model for branch-and-cut with and without relax-and-fix.

	Non-collaborative		Branch-and-cut			Branch-and-cut with Relax-and-fix	
	Distance	Time (s)	Distance	Lower bound	Time (s)	Distance	Time (s)
26	2557.86	43	1332.62	455.62	TL	951.39	1265
27	2617.10	77	5792.42	491.20	TL	1891.55	1292
28	2676.33	77	272.39	240.63	TL	274.33	153
29	2735.57	16	297.43	210.46	TL	351.15	1202
30	2794.81	0	593.04	329.44	TL	631.87	1202
31	2854.05	1206	396.02	295.07	TL	396.02	1202
32	2913.28	36	498.75	265.28	TL	501.98	1203
33	2972.52	29	708.92	255.42	TL	709.61	1204
34	3031.76	1708	527.71	277.31	TL	577.66	1205
35	3091.00	61	578.23	257.89	TL	571.91	1205
36	3150.23	129	840.63	325.16	TL	885.98	1205
37	3209.47	67	366.28	219.66	TL	376.79	1205
38	3268.71	362	580.68	217.00	TL	625.18	1206
39	3327.95	3734	599.57	281.44	TL	655.92	1207
40	3387.19	4734	1024.85	211.53	TL	920.19	1210
41	3446.42	183	680.60	193.70	TL	655.55	1211
42	3505.66	TL	591.81	190.68	TL	756.01	1213
43	3564.90	6133	722.27	227.01	TL	665.14	1217
44	3624.14	674	702.54	186.39	TL	811.86	1217
45	3683.37	4059	1934.10	290.70	TL	1171.91	1234
46	3742.61	5460	689.54	211.89	TL	693.93	1228
47	3801.85	6543	777.96	234.63	TL	739.78	1238
48	3861.09	3624	1195.20	316.85	TL	789.08	1245
49	3920.32	8	967.74	242.42	TL	778.10	1246
50	3979.56	761	1712.71	296.64	TL	1013.57	1307

1970 seconds and reached an optimality gap of around 48% after 20 hours, which is the reported result. The relax-and-fix algorithm, converged after 4900 seconds. We can see that the pure branch-and-cut might be more suitable given an appropriate time limit, while if there are tight time constraints relax-and-fix is recommended. In both cases, the hub was identified to be Braeside. Moreover, again for this case study, it is possible to see that finding an efficient collaboration strategy can be more complicating them operating alone, however, even sub-optimal solutions for the collaborative case can provide savings of more than 60% when comparing to the optimal strategy for operating alone.

Overall, the experiments showed that the distances travelled can be greatly reduced (up to roughly one-third, on average) when suppliers collaborate with each other to better utilise the already available infrastructure. These savings were usually not related to an optimal solution for the collaborative case, suggesting that even higher reductions could be achieved.

Table 28 – Results comparing non-collaborative, collaborative with branch-and-cut and collaborative with relax-and-fix approaches (part 3). For each approach, we report the solution value and time to reach that solution (TL if the time limit was reached). For the branch-and-cut we also report the best lower bound. Bold values represent the best solution between collaborative model for branch-and-cut with and without relax-and-fix.

	Non-collaborative		Branch-and-cut			Branch-and-cut with Relax-and-fix	
	Distance	Time (s)	Distance	Lower bound	Time (s)	Distance	Time (s)
51	4038.80	3726	6735.88	326.02	TL	1264.53	1717
52	4098.04	< 1	172.03	172.03	1	172.28	< 1
53	4157.28	< 1	193.47	193.47	23	198.01	1
54	4216.51	< 1	247.47	247.47	33	263.21	1
55	4275.75	< 1	206.50	206.50	755	486.96	31
56	4334.99	< 1	218.83	218.83	1142	516.84	26
57	4394.23	< 1	397.59	397.59	3193	750.92	30
58	4453.46	< 1	204.41	204.41	766	261.33	142
59	4512.70	1	353.61	263.51	TL	368.85	1202
60	4571.94	1	408.14	285.95	TL	401.04	1205
61	4631.18	1	434.59	249.41	TL	451.67	1207
62	1254.31	199	462.81	276.30	TL	490.97	1206
63	1258.89	5	507.87	274.71	TL	500.94	1207
64	1627.83	3	632.97	270.66	TL	645.13	1211
65	1537.04	26	540.38	288.67	TL	498.67	1214
66	1456.19	7	681.96	264.60	TL	558.72	1217
67	1417.72	968	672.21	269.22	TL	566.14	1213
68	1358.45	47	584.65	240.06	TL	488.16	1216
69	1673.89	19	480.32	281.90	TL	487.75	1218
70	1538.13	47	524.51	264.14	TL	609.56	1219
71	2194.64	14	2138.79	309.27	TL	620.93	1236
72	2371.48	13	677.43	307.48	TL	809.77	1261
73	2695.65	29	646.56	344.42	TL	643.13	1278
74	2653.76	50	913.35	329.17	TL	761.69	1281
Average	2663.98	898.46	864.10	318.92	3339.36	687.87	1100.03

Table 29 – Areas allocated to suppliers.

Suppliers	Area
Laverton	West
Somerton	North
Scoresby	East
Braeside	South
Dandenong	South east

Table 30 – Average weekly requirements from suppliers to stores in each of the areas (kg).

Supplier	West	North	East	South east	South
Laverton	150	150	250	250	50
Somerton	150	200	250	150	100
Scoresby	150	200	250	380	150
Braeside	100	150	200	300	100
Dandenong	150	250	250	200	50

Figure 19 – Hardware stores, suppliers and local distribution areas in Melbourne (Australia).

Table 31 – Comparison between non-collaborative approach and collaborative approach (branch-and-cut and relax-and-fix). Savings related to the non-collaborative approach are also reported. Bold values indicate best result. The sign “-” indicate that the cell is not applicable to that particular case.

	Non-collaborative	Branch-and-cut	Relax-and-fix
Exchange costs	-	125.10	114.21
Routing costs	-	365.88	421.96
Total costs	1466	490.98	536.17
Saving	-	67%	63%

5.4 Remarks on design of hub exchange networks

Suppliers operating their own independent distribution systems in large urban areas are contributing towards increasing environmental costs due to service frequency requirements and low vehicle utilisation. Independent distribution systems operated by number of suppliers to common stores in large urban areas can be transformed into collaborative distribution systems to substantially reduce the distance travelled as well as the associated environmental impacts. Optimisation techniques can be used to determine a supplier to be used for transshipment of goods as well as allocating retail stores to suppliers for local routes.

A number of extensions to the model presented in this paper can be envisaged, depending on the nature of the independent distribution networks. The existing model assumes a single vehicle per supplier, however multiple vehicles can be handled by changing the resource constraint. Also, only a single hub is selected per period was considered here.

The model can be extended to incorporate multiple hubs. Currently, routes starting at one supplier cannot visit another supplier. This constraint could be relaxed, giving the problem a *pick-up and delivery* structure. In addition, other costs, such as inventory or transportation costs proportional to the load of trucks, can also be added to the model.

We show that collaborative urban distribution systems can be designed to substantially reduce transport distances travelled as well as the number of vehicles required. Our experiments show substantial savings and the application of the techniques presented to a large retail distribution network in Melbourne, Australia, showed savings of approximately 67% in travel distance by using a collaborative distribution network. This will lead to substantial savings in fuel consumption and emissions from freight vehicles.

CONCLUSIONS AND FUTURE WORK

Freight distribution systems are an essential component of our daily lives and may affect from people to countries. If they are poorly designed or operated, there are economical, societal and/or environmental impacts. The costs of a badly designed system might lead to higher costs for the consumers, if it does not make the business itself impractical due to the intense competitive nature of logistic industry. Also, trucks might be mostly empty for good portions of these long journeys. Moreover, an inefficient system can increase the number of vehicles on roads and in the city: increasing traffic, noise and air pollution and the possibility of incidents.

In this thesis, we explore the technique of consolidation to increase the efficiency of freight distribution systems. More broadly, we focus reducing the unused space inside trucks when they are travelling. Increasing rates of consolidation (or reducing unused space) is a known idea to reduce the number of trucks (or trips) needed to perform distribution tasks. This increase is intended to reduce the negative impacts aforementioned. We explore consolidation in three separate optimisation scenarios.

The first one concerns Container Loading Problems (CLPs). In these problems, the goal is to position a set of boxes inside containers (or trucks) to maximise the volume/value loaded (or minimise the cost of the containers used). CLP is a classical three-dimensional packing problem. In this thesis, we explore the container loading problem as part of a more dynamic and time sensitive logistic environment (e.g., cross-docking). In such environments, there is the need to account for the schedule of arrival of the boxes, since they are not always available for loading, we name this situation as the *Container Loading Problem with Time Availability Constraints* (CLPTAC). We explore two variations of CLPTAC: Output maximisation (CLPTAC-Om) and Input minimisation (CLPTAC-Im). In CLPTAC-Om, the goal is to select a subset of boxes to load into a limited number of trucks maximising the loaded volume and minimising the ready time of the trucks. For this variation, we propose a dynamic programming algorithm able to handle deterministic and stochastic cases. The uncertainty refers to the arrival time of the boxes. In CLPTAC-Im, the goal is to

minimise the number of trucks and to dispatch as much volume of goods as soon as possible, ensuring all boxes are loaded. For this, we propose a mixed integer linear programming model and a constructive heuristic which is also able to handle uncertainty in arrival time of boxes. For future work, besides incorporating other practical constraints (e.g., stability and weight distribution) which are already explored for the traditional Container Loading Problem in the literature, we can incorporate aspects related to the time sensitiveness of the operation. For example, a more detailed modelling of the processing capacity of the facility might consider the allocation of people to perform loading operations, which directly affect loading time of trucks.

In the second scenario, we explore a network scheduling problem with cross-docking. For this, we consider an open network in which suppliers have to satisfy their consumers but, instead of direct supplier-consumer links, the goods must go through a consolidation centre (cross-docking facility). In these consolidation centres, the goods might be recombined into (often smaller) trucks that are dispatched to their final destinations (customers). The goal is to minimise total transportation costs. For this problem, we propose a mixed integer linear programming model. The model is reformulated with an exponential number of constraints and is solved using a logic-based Benders algorithm. Many characteristics of distribution systems may be incorporated in the network scheduling problem with cross-docking. This flexibility is, in part, due to the decomposition framework and some characteristics might be contemplated without changing it. Some examples of these characteristics are: (i) Closed routes for visiting multiple customers, (ii) internal resources of the cross-dock and (iii) product interchangeability. It is of our particular interest to incorporate the three-dimensional aspects of the truck loading into the network scheduling problem with cross-docking, specially because it is related to CLPTAC.

In the third scenario, we considered a collaborative distribution problem. Traditionally, each supplier handles its own demand. Alternatively, they can use the already available infrastructure to design a hub exchange network. In this network, one of the suppliers depot acts as a hub. Different suppliers might exchange goods in the hub, allowing consolidation on the delivery routes. For this, we propose a mixed integer linear programming model which selects the hub and designs routes for exchange and distribution of goods minimising total exchange and distribution costs. Also, a math-heuristic algorithm is proposed to reduce computational time. As future work, it would be interesting to extend the model to contemplate multiple vehicles for suppliers and explore different solution methods (exact and heuristic). Moreover, some practical characteristics can be incorporated such as time windows for the clients and heterogeneous fleets. Finally, a game theory framework can be used to share distribution costs among the suppliers.

The three different cases we explore have in common the fact that they intend to increase the efficiency of logistic systems. In some sense, we explore from a very local and operational process (container loading) to a broader network scheduling case to a more tactical network design problem. The three cases have contributions on their own, but

they potentially lead to new research and new class of applications. We hope this thesis encourage this new exploration and help to increase the efficiency of freight distribution systems because we depend on logistics.

BIBLIOGRAPHY

ABAD, H. K. E.; VAHDANI, B.; SHARIFI, M.; ETEBARI, F. A bi-objective model for pickup and delivery pollution-routing problem with integration and consolidation shipments in cross-docking system. **Journal of Cleaner Production**, v. 193, p. 784–801, 2018. Citation on page [54](#).

ABDOU, G.; ELMASRY, M. 3D random stacking of weakly heterogeneous palletization problems. **International Journal of Production Research**, v. 37, p. 1505–1524, 1999. Citation on page [46](#).

AGUSTINA, D.; LEE, C.; PIPLANI, R. A Review: Mathematical Modles for Cross Docking Planning. **International Journal of Engineering Business Management**, v. 2, p. 47–54, 2010. Citations on pages [30](#) and [86](#).

AHKAMIRAAD, A.; WANG, Y. Capacitated and multiple cross-docked vehicle routing problem with pickup, delivery, and time windows. **Computers & Industrial Engineering**, v. 119, p. 76–84, 2018. Citations on pages [55](#) and [86](#).

AHMADIZAR, F.; ZEYNIVAND, M.; ARKAT, J. Two-level vehicle routing with cross-docking in a three-echelon supply chain: A genetic algorithm approach. **Applied Mathematical Modelling**, v. 39, p. 7065–7081, 2015. Citation on page [55](#).

AKARTUNALI, K.; MILLER, A. J. A heuristic approach for big bucket multi-level production planning problems. **European Journal of Operational Research**, Elsevier, v. 193, p. 396–411, 2009. Citation on page [107](#).

ALONSO, M. T.; ALVAREZ-VALDES, R.; IORI, M.; PARREÑO, F.; TAMARIT, J. M. Mathematical models for multicontainer loading problems. **Omega**, Elsevier, v. 66, p. 106–117, 2017. Citation on page [47](#).

ALONSO, M. T.; ALVAREZ-VALDES, R.; PARREÑO, F.; TAMARIT, J. M. Algorithms for pallet building and truck loading in an interdepot transportation problem. **Mathematical Problems in Engineering**, v. 2016, 2016. Citation on page [47](#).

AMINI, A.; TAVAKKOLI-MOGHADDAM, R. A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks. **Computers & Industrial Engineering**, v. 96, p. 180–191, 2016. Citations on pages [33](#), [38](#), [39](#), and [85](#).

AOUAM, T.; GERYL, K.; KUMAR, K.; BRAHIMI, N. Production planning with order acceptance and demand uncertainty. **Computers & Operations Research**, Elsevier, v. 91, p. 145–159, 2018. Citation on page [107](#).

ARABANI, A. B.; GHOMI, S.; ZANDIEH, M. A multi-criteria cross-docking scheduling with just-in-time approach. **International Journal of Advanced Manufacturing Technology**, v. 49, p. 741–756, 2010. Citations on pages [33](#), [39](#), and [85](#).

_____. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. **Expert Systems with Applications**, v. 38, p. 1964–1979, 2011. Citations on pages [32](#), [33](#), [35](#), [38](#), [39](#), and [85](#).

ARABANI, A. B.; ZANDIEH, M.; GHOMI, S. Multi-objective genetic-based algorithms for a cross-docking scheduling problem. **Applied Soft Computing Journal**, Elsevier B.V., v. 11, p. 4954–4970, 2011. Citations on pages [35](#), [38](#), and [39](#).

ARAYA, I.; GUERRERO, K.; NUÑEZ, E. VCS: A new heuristic function for selecting boxes in the single container loading problem. **Computers & Operations Research**, Elsevier Ltd, v. 82, p. 27–35, 2017. Citations on pages [46](#), [51](#), and [53](#).

ARAYA, I.; RIFF, M. C. A beam search approach to the container loading problem. **Computers & Operations Research**, Elsevier, v. 43, p. 100–107, 2014. Citations on pages [46](#) and [53](#).

ASSADI, M. T.; BAGHERI, M. Differential evolution and Population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems. **Computers & Industrial Engineering**, v. 96, p. 149–161, 2016. Citations on pages [33](#), [38](#), [39](#), and [85](#).

BANIAMERIAN, A.; BASHIRI, M.; TAVAKKOLI-MOGHADDAM, R. Modified variable neighborhood search and genetic algorithm for profitable heterogeneous vehicle routing problem with cross-docking. **Applied Soft Computing**, v. 75, p. 441–460, 2019. Citation on page [56](#).

BAZGOSHA, A.; RANJBAR, M.; JAMILI, N. Scheduling of loading and unloading operations in a multi stations transshipment terminal with release date and inventory constraints. **Computers & Industrial Engineering**, v. 106, p. 20–31, 2017. Citations on pages [36](#), [39](#), and [86](#).

BEASLEY, J. E. An Exact Two-Dimensional Non-Guillotine Cutting Tree Search Procedure. **Operations Research**, v. 33, p. 49–64, 1985. Citation on page [45](#).

BELLANGER, A.; HANAFI, S.; WILBAUT, C. Three-stage hybrid-flowshop model for cross-docking. **Computers & Operations Research**, v. 40, p. 1109–1121, 2013. Citations on pages [34](#), [39](#), and [85](#).

BELLE, J. V.; VALCKENAERS, P.; CATTRYSSSE, D. Cross-docking: State of the art. **Omega**, Elsevier, v. 40, p. 827–846, 2012. Citations on pages [12](#), [28](#), [29](#), [30](#), and [86](#).

BELLE, J. V.; VALCKENAERS, P.; Vanden Berghe, G.; CATTRYSSSE, D. A tabu search approach to the truck scheduling problem with multiple docks and time windows. **Computers & Industrial Engineering**, v. 66, p. 818–826, 2013. Citations on pages [33](#) and [39](#).

BISCHOFF, E. E.; RATCLIFF, M. S. W. Issues in the development of approaches to container loading. **Omega**, Elsevier, v. 23, p. 377–390, 1995. Citations on pages [50](#) and [51](#).

BOLAND, N.; FISCHETTI, M.; MONACI, M.; SAVELSBERGH, M. Proximity benders: a decomposition heuristic for stochastic programs. **Journal of Heuristics**, Springer, v. 22, p. 181–198, 2016. Citation on page [107](#).

BOLDUC, M.-C.; RENAUD, J.; BOCTOR, F. A heuristic for the routing and carrier selection problem. **European Journal of Operational Research**, Elsevier, v. 183, p. 926–932, 2007. Citation on page [59](#).

BOLDUC, M.-C.; RENAUD, J.; BOCTOR, F.; LAPORTE, G. A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. **Journal of the Operational Research Society**, v. 59, p. 776–787, 2008. Citation on page [59](#).

- BORTFELDT, A. A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. **Computers & Operations Research**, Elsevier, v. 39, p. 2248–2257, 2012. Citations on pages [49](#) and [50](#).
- BORTFELDT, A.; WÄSCHER, G. Constraints in container loading – A state-of-the-art review. **European Journal of Operational Research**, Elsevier B.V., v. 229, p. 1–20, 2013. Citations on pages [41](#), [43](#), and [52](#).
- BOYSEN, N.; FLIEDNER, M. Cross dock scheduling: Classification, literature review and research agenda. v. 38, p. 413–422, 2010. Citation on page [30](#).
- BRUNETTA, L.; GRÉGOIRE, P. A General Purpose Algorithm for Three-Dimensional Packing. **INFORMS Journal on Computing**, v. 17, p. 328–338, 2005. Citations on pages [44](#) and [53](#).
- BUIJS, P.; VIS, I. F. A.; CARLO, H. J. Synchronization in cross-docking networks: A research classification and framework. **European Journal of Operational Research**, v. 239, p. 593–608, 2014. Citations on pages [12](#), [28](#), [29](#), [30](#), and [86](#).
- BUTTON, K. J.; STOUGH, R. **Air transport networks: Theory and policy implications**. [S.l.]: Edward Elgar Publishing, 2000. Citation on page [103](#).
- CESCHIA, S.; SCHAERF, A. Local search for a multi-drop multi-container loading problem. **Journal of Heuristics**, v. 19, p. 275–294, 2013. Citations on pages [48](#) and [53](#).
- CHAN, F. T. S.; BHAGWAT, R.; KUMAR, N.; TIWARI, M. K.; LAM, P. Development of a decision support system for air-cargo pallets loading problem: A case study. **Expert Systems with Applications**, v. 31, p. 472–485, 2006. Citations on pages [47](#) and [52](#).
- CHE, C. H.; HUANG, W.; LIM, A.; ZHU, W. The multiple container loading cost minimization problem. **European Journal of Operational Research**, v. 214, p. 501–511, 2011. Citations on pages [46](#), [47](#), [51](#), and [53](#).
- CHEN, C. S.; LEE, S. M.; SHEN, Q. S. An analytical model for the container loading problem. **European Journal of Operational Research**, v. 80, p. 68–76, 1995. Citations on pages [42](#), [43](#), [53](#), [68](#), and [72](#).
- CHEN, F.; LEE, C.-Y. Minimizing the makespan in a two-machine cross-docking flow shop problem. **European Journal of Operational Research**, v. 193, p. 59–72, 2009. Citations on pages [33](#), [34](#), [38](#), and [39](#).
- CHEN, F.; SONG, K. Minimizing makespan in two-stage hybrid cross docking scheduling problem. **Computers & Operations Research**, v. 36, p. 2066–2073, 2009. Citations on pages [34](#), [38](#), [39](#), and [85](#).
- CHEN, M. C.; HSIAO, Y. H.; REDDY, R. H.; TIWARI, M. K. The Self-Learning Particle Swarm Optimization approach for routing pickup and delivery of multiple products with material handling in multiple cross-docks. **Transportation Research Part E: Logistics and Transportation Review**, v. 91, p. 208–226, 2016. Citations on pages [55](#) and [86](#).
- CHEN, P.; GUO, Y.; LIM, A.; RODRIGUES, B. Multiple crossdocks with inventory and time windows. **Computers & Operations Research**, v. 33, p. 43–63, 2006. Citations on pages [57](#) and [86](#).

- CHIEN, C.-F.; LEE, C.-Y.; HUANG, Y.-C.; WU, W.-T. An efficient computational procedure for determining the container-loading pattern. **Computers & Industrial Engineering**, Elsevier, v. 56, p. 965–978, 2009. Citation on page [44](#).
- CHINH, N. Q.; KIM, H. C.; JIANG, S.; ZHANG, N. Collaborative vehicle routing problem for urban last-mile logistics. In: **IEEE International Conference on Systems, Man, and Cybernetics**. Budapest, Hungary: [s.n.], 2016. p. 1531–1536. Citation on page [60](#).
- CHMIELEWSKI, A.; NAUJOKS, B.; JANAS, M.; CLAUSEN, U. Optimizing the Door Assignment in LTL-Terminals. **Transportation Science**, v. 43, p. 198–210, 2009. Citations on pages [35](#), [39](#), and [86](#).
- CHU, C. W. A heuristic algorithm for the truckload and less-than-truckload problem. **European Journal of Operational Research**, v. 165, p. 657–667, 2005. Citation on page [59](#).
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations research**, v. 12, p. 568–581, 1964. Citation on page [59](#).
- CLEOPHAS, C.; COTTRILL, C.; EHMKE, J. F.; TIERNEY, K. Collaborative urban transportation: Recent advances in theory and practice. **European Journal of Operational Research**, Elsevier, v. 273, p. 801–816, 2019. Citations on pages [57](#), [58](#), and [59](#).
- COOK, R. L.; GIBSON, B.; MACCURDY, D. A lean approach to cross docking. **Supply Chain Management review**, p. 54–59, 2005. Citation on page [28](#).
- CORRECHER, J.; ALONSO, M.; PARREÑO, F.; ALVAREZ-VALDES, R. Solving a large multicontainer loading problem in the car manufacturing industry. **Computers & Operations Research**, v. 82, p. 139–152, 2017. Citations on pages [48](#) and [53](#).
- COSTA, A. M. A survey on benders decomposition applied to fixed-charge network design problems. **Computers & operations research**, v. 32, p. 1429–1450, 2005. Citation on page [86](#).
- COTA, P. M.; GIMENEZ, B. M. R.; ARAÚJO, D. P. M.; NOGUEIRA, T. H.; SOUZA, M. C. de; RAVETTI, M. G. Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre. **Computers & Industrial Engineering**, v. 95, p. 135–143, 2016. Citations on pages [34](#), [38](#), [39](#), and [85](#).
- CÔTÉ, J.-F.; POTVIN, J.-Y. A tabu search heuristic for the vehicle routing problem with private fleet and common carrier. **European Journal of Operational Research**, v. 198, p. 464–469, 2009. Citation on page [59](#).
- CRUIJSSEN, F.; DULLAERT, W.; FLEUREN, H. Horizontal Cooperation in Transport and Logistics : A Literature Review. **Transportation Journal**, v. 46, p. 22–39, 2007. Citations on pages [58](#) and [59](#).
- DAGANZO, C. F. **Logistics systems analysis**. [S.l.]: Springer Science & Business Media, 2005. Citation on page [103](#).
- DAGANZO, C. F.; NEWELL, G. F. Configuration of physical distribution networks. **Networks**, Wiley Online Library, v. 16, p. 113–132, 1986. Citation on page [103](#).

- DAVIES, A. P.; BISCHOFF, E. E. Weight distribution considerations in container loading. **European Journal of Operational Research**, Elsevier, v. 114, p. 509–527, 1999. Citations on pages [42](#), [43](#), [51](#), and [53](#).
- DEFRYN, C.; SÖRENSEN, K.; CORNELISSENS, T. The selective vehicle routing problem in a collaborative environment. **European Journal of Operational Research**, v. 250, n. 2, p. 400–411, 2016. Citation on page [60](#).
- DIRK, M.; BORTFELDT, A. A Hybrid Algorithm for the Vehicle Routing Problem with Pickup and Delivery and Three-dimensional Loading Constraints. **European Journal of Operational Research**, v. 254, p. 840–858, 2016. Citation on page [49](#).
- DOMINGO, B.; PONNAMBALAM, S.; KANAGARAJ, G. A differential evolution based algorithm for single container loading problem. p. 105–111, 2013. Citations on pages [42](#), [45](#), and [53](#).
- DONDO, R.; MÉNDEZ, C. A.; CERDÁ, J. The multi-echelon vehicle routing problem with cross docking in supply chain management. **Computers & Chemical Engineering**, v. 35, p. 3002–3024, 2011. Citations on pages [55](#) and [86](#).
- DUIN, R. V.; MUÑOZURI, J. City logistics: Mapping the future. In: _____. [S.l.]: CRC Press, Taylor & Francis, 2015. chap. 8. Citation on page [104](#).
- EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: IEEE. **Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on**. [S.l.], 1995. p. 39–43. Citation on page [33](#).
- EGEBLAD, J.; GARAVELLI, C.; LISI, S.; PISINGER, D. Heuristics for container loading of furniture. **European Journal of Operational Research**, v. 200, p. 881–892, 2010. Citation on page [41](#).
- EGEBLAD, J.; NIELSEN, B. K.; BRAZIL, M. Translational packing of arbitrary polytopes. **Computational Geometry**, Elsevier, v. 42, p. 269–288, 2009. Citation on page [41](#).
- EGEBLAD, J.; NIELSEN, B. K.; ODGAARD, A. Fast neighborhood search for two-and three-dimensional nesting problems. **European Journal of Operational Research**, Elsevier, v. 183, p. 1249–1266, 2007. Citation on page [41](#).
- ELEY, M. Solving container loading problems by block arrangement. **European Journal of Operational Research**, v. 141, p. 393–409, 2002. Citations on pages [48](#) and [53](#).
- _____. A bottleneck assignment approach to the multiple container loading problem. **OR Spectrum**, v. 25, p. 45–60, 2003. Citations on pages [46](#), [47](#), [48](#), [51](#), and [53](#).
- ENDERER, F.; CONTARDO, C.; CONTRERAS, I. Integrating dock-door assignment and vehicle routing with cross-docking. **Computers & Operations Research**, v. 88, p. 30–43, 2017. Citations on pages [56](#) and [86](#).
- FANSLAU, T.; BORTFELDT, A. A tree search algorithm for solving the container loading problem. **INFORMS Journal on Computing**, v. 22, p. 222–235, 2010. Citations on pages [44](#), [46](#), [49](#), and [53](#).
- FARD, S. S.; VAHDANI, B. Assignment and scheduling trucks in cross-docking system with energy consumption consideration and trucks queuing. **Journal of Cleaner Production**, v. 213, p. 21–41, 2019. Citation on page [37](#).

FASANO, G. Cargo Analytical Integration in Space Engineering: A Three-dimensional Packing Model. In: CIRIANI, T.; GLIOZZI, S.; JOHNSON, E.; TADEI, R. (Ed.). **Operational Research in Industry**. [S.l.]: Palgrave Macmillan UK, 1999. p. 232–246. Citations on pages [43](#) and [45](#).

FISCHER, K.; KUHN, N.; MÜLLER, H. J.; MÜLLER, J. P.; PISCHEL, M. Sophisticated and distributed: The transportation domain. In: SPRINGER. **European Workshop on Modelling Autonomous Agents in a Multi-Agent World**. [S.l.], 1993. p. 122–138. Citation on page [103](#).

FISCHETTI, M.; MONACI, M. Proximity search heuristics for wind farm optimal layout. **Journal of Heuristics**, Springer, v. 22, p. 459–474, 2016. Citation on page [107](#).

FLEURY, P. F. Logística no brasil: situação atual e transição para uma economia verde. Fundação Brasileira para o Desenvolvimento Sustentável, 2012. Citation on page [21](#).

FORGER, G. UPS starts world's premiere cross-docking operation. **Modern Material Handling**, v. 36, p. 36–38, 1995. Citation on page [28](#).

FRASER, H. J.; GEORGE, J. A. Integrated container loading software for pulp and paper industry. **European Journal of Operational Research**, v. 77, p. 466–474, 1994. Citation on page [47](#).

Freight Transport Association. **The Logistics Report 2014**. [S.l.], 2014. 68 p. Citation on page [21](#).

GANSTERER, M.; HARTL, R. F. Collaborative vehicle routing: A survey. **European Journal of Operational Research**, Elsevier B.V., v. 268, p. 1–12, 2018. Citations on pages [57](#), [58](#), and [59](#).

GAREY, M. R.; JOHNSON, D. S.; SETHI, R. The complexity of flowshop and jobshop scheduling. **Mathematics of operations research**, v. 1, p. 117–129, 1976. Citation on page [33](#).

GELAREH, S.; MONEMI, R. N.; SEMET, F.; GONCALVES, G. A branch-and-cut algorithm for the truck dock assignment problem with operational time constraints. **European Journal of Operational Research**, v. 249, p. 1144–1152, 2016. Citations on pages [36](#), [39](#), and [86](#).

GENDREAU, M.; IORI, M.; LAPORTE, G.; MARTELLO, S. A Tabu Search Algorithm for a Routing and Container Loading Problem. **Transportation Science**, v. 40, p. 342–350, 2006. Citations on pages [48](#) and [49](#).

GEOFFRION, A. M.; GRAVES, G. W. Multicommodity Distribution System Design by Benders Decomposition. **Management Science**, p. 822–844, 1974. Citations on pages [57](#) and [86](#).

GEORGE, J. A.; ROBINSON, D. F. A heuristic for packing boxes into a container. **Computers & Operations Research**, v. 7, p. 147–156, 1980. Citations on pages [42](#), [46](#), [53](#), and [74](#).

GILMORE, P. C.; GOMORY, R. E. Multistage cutting stock problems of two and more dimensions. **Operations research**, INFORMS, v. 13, p. 94–120, 1965. Citation on page [43](#).

GONÇALVES, J. F.; RESENDE, M. G. C. A parallel multi-population biased random-key genetic algorithm for a container loading problem. **Computers & Operations Research**, v. 39, p. 179–190, 2012. Citations on pages [45](#) and [53](#).

GRANGIER, P.; GENDREAU, M.; LEHUÉDÉ, F.; ROUSSEAU, L. M. An adaptive large neighborhood search for the two-echelon multiple-trip vehicle routing problem with satellite synchronization. **European Journal of Operational Research**, v. 254, p. 80–91, 2016. Citation on page [56](#).

GRANGIER, P.; GENDREAU, M.; LEHUÉDÉ, F.; ROUSSEAU, L.-M. A Matheuristic Based on Large Neighborhood Search for the Vehicle Routing Problem with Cross-Docking. **Computers & Operations Research**, v. 84, p. 116–126, 2017. Citation on page [55](#).

GUAJARDO, M.; RÖNNQVIST, M. A review on cost allocation methods in collaborative transportation. **International Transactions in Operational Research**, v. 23, p. 371–392, 2016. Citation on page [58](#).

GUASTAROBA, G.; SPERANZA, M. G.; VIGO, D. Intermediate Facilities in Freight Transportation Planning : A Survey. **Transportation Science**, v. 50, p. 763–789, 2016. Citations on pages [30](#) and [86](#).

HAYASHI, K.; NEMOTO, T.; VISSER, J. E-commerce and City Logistics Solution. In: TANIGUCHI, E.; THOMPSON, R. G. (Ed.). **City logistics: mapping the future**. [S.l.]: CRC Press, 2014. chap. 5, p. 55–78. Citation on page [22](#).

HEMMINKI, J.; LEIPALA, T.; NEVALAINEN, O. On-line packing with boxes of different sizes. **International Journal of Production Research**, v. 36, p. 2225–2245, 1998. Citation on page [46](#).

HERMEL, D.; HASHEMINIA, H.; ADLER, N.; FRY, M. J. A solution framework for the multi-mode resource-constrained cross-dock scheduling problem. **Omega (United Kingdom)**, Elsevier, v. 59, p. 157–170, 2016. Citations on pages [37](#), [38](#), and [39](#).

HERNÁNDEZ, S.; PEETA, S.; KALAFATAS, G. A less-than-truckload carrier collaboration planning problem under dynamic capacities. **Transportation Research Part E: Logistics and Transportation Review**, v. 47, p. 933–946, 2011. Citations on pages [57](#), [86](#), and [87](#).

HIGINO, W.; CHAVES, A. Biased Random-key Genetic Algorithm Applied to the Vehicle Routing Problem with Private Fleet and Common Carrier. **2018 IEEE Congress on Evolutionary Computation (CEC)**, p. 1–8, 2018. Citation on page [59](#).

HOOKE, J. N.; OTTOSSON, G. Logic-based Benders decomposition. **Mathematical Programming**, v. 96, p. 33–60, 2003. Citation on page [87](#).

HOSSEINI, S. D.; SHIRAZI, M.; KARIMI, B. Cross-docking and milk run logistics in a consolidation network: A hybrid of harmony search and simulated annealing approach. **Journal of Manufacturing Systems**, The Society of Manufacturing Engineers, v. 33, p. 567–577, 2014. Citation on page [28](#).

HU, Z. H.; ZHAO, Y.; CHOI, T. M. Vehicle routing problem for fashion supply chains with cross-docking. **Mathematical Problems in Engineering**, v. 2013, 2013. Citation on page [55](#).

IVANCIC, N.; MATHUR, K.; MOHANTY, B. B. An Integer Programming Based Heuristic Approach to the Three-dimensional Packing Problem. **Journal of manufacturing and operations management**, n. 4, p. 268–289, 1989. Citations on pages [50](#), [68](#), and [74](#).

JAMRUS, T.; CHIEN, C.-F. Extended priority-based hybrid genetic algorithm for the less-than-container loading problem. **Computers & Industrial Engineering**, Elsevier Ltd, v. 96, p. 227–236, 2016. Citations on pages [45](#) and [53](#).

JOHNSON, S. M. Optimal two-and three-stage production schedules with setup times included. **Naval research logistics quarterly**, Wiley Online Library, v. 1, p. 61–68, 1954. Citation on page [34](#).

JR, W. V. B. International space station overview. 1999. Citation on page [43](#).

JUNQUEIRA, L.; MORABITO, R.; YAMASHITA, D. S. Mip-based approaches for the container loading problem with multi-drop constraints. **Annals of Operations Research**, v. 199, n. 1, p. 51–75, 2012. Citation on page [45](#).

JUNQUEIRA, L.; MORABITO, R.; YAMASHITA, S. D. Three-dimensional container loading models with cargo stability and load bearing constraints. **Computers & Operations Research**, v. 39, p. 74–85, 2012. Citation on page [45](#).

JUNQUEIRA, L.; OLIVEIRA, J. F.; CARRAVILLA, M. A.; MORABITO, R. An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. **International Transactions in Operational Research**, v. 20, p. 645–666, 2013. Citations on pages [49](#) and [53](#).

KAEWPUANG, R.; NIYATO, D.; TAN, P. S.; WANG, P. Cooperative Management in Full-Truckload and Less-Than-Truckload Vehicle System. **IEEE Transactions on Vehicular Technology**, v. 66, p. 5707–5722, 2017. Citation on page [60](#).

KESHTZARI, M.; NADERI, B.; MEHDIZADEH, E. An improved mathematical model and a hybrid metaheuristic for truck scheduling in cross-dock problems. **Computers & Industrial Engineering**, v. 91, p. 197–204, 2016. Citations on pages [32](#), [39](#), and [85](#).

KINNEAR, E. Is there any magic in cross-docking? **Supply Chain Management: An International Journal**, v. 2, p. 49–52, 2006. Citation on page [28](#).

KONUR, D.; GOLIAS, M. M. Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty. **Computers & Industrial Engineering**, v. 65, p. 663–672, 2013. Citations on pages [35](#), [39](#), and [86](#).

KOSTER, R. de; LE-DUC, T.; ROODBERGEN, K. J. Design and control of warehouse order picking: A literature review. **European Journal of Operational Research**, v. 182, p. 481–501, 2007. Citation on page [28](#).

KRAJEWSKA, M. A.; KOPFER, H. Transportation planning in freight forwarding companies: Tabu search algorithm for the integrated operational transportation planning problem. **European Journal of Operational Research**, v. 197, p. 741–751, 2009. Citation on page [60](#).

KROER, L. R.; FOVERSKOV, K.; VILHELMSSEN, C.; HANSEN, A. S.; LARSEN, J. Planning and scheduling operating rooms for elective and emergency surgeries with uncertain duration. **Operations Research for Health Care**, Elsevier, 2018. Citation on page [107](#).

LADIER, A.-L.; ALPAN, G. Robust cross-dock scheduling with time windows. **Computers & Industrial Engineering**, v. 99, p. 16–28, 2016. Citations on pages 34, 39, and 86.

_____. Crossdock truck scheduling with time windows: earliness, tardiness and storage policies. **Journal of Intelligent Manufacturing**, v. 29, p. 569–583, 2018. Citations on pages 34, 39, and 85.

LAPORTE, G. What You Should Know about the Vehicle Routing Problem. **Naval Research Logistics**, n. August, p. 811–819, 2007. Citation on page 28.

LEE, Y. H.; JUNG, J. W.; LEE, K. M. Vehicle routing scheduling for cross-docking in the supply chain. **Computers & Industrial Engineering**, v. 51, p. 246–256, 2006. Citations on pages 54 and 86.

LIAO, T. W.; EGBELU, P. J.; CHANG, P. C. Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. **Applied Soft Computing Journal**, v. 12, p. 3683–3697, 2012. Citations on pages 35, 39, and 85.

LIM, A.; MA, H.; QIU, C.; ZHU, W. The single container loading problem with axle weight constraints. **International Journal of Production Economics**, Elsevier, v. 144, p. 358–369, 2013. Citations on pages 45 and 53.

LIM, A.; ZHANG, X. The container loading problem. **SAC '05: Proceedings of the 2005 ACM symposium on Applied computing**, p. 913–917, 2005. Citations on pages 48 and 53.

LIU, R.; JIANG, Z.; LIU, X.; DEPARTMENT, F. C. Task selection and routing problems in collaborative truckload transportation Ran. **Transportation Research Part E**, v. 46, p. 1071–1085, 2010. Citation on page 59.

LOH, T. H.; NEE, A. Y. C. A packing algorithm for hexahedral boxes. In: **Proceedings of the Conference of Industrial Automation, Singapore**. [S.l.: s.n.], 1992. p. 115–126. Citations on pages 50 and 51.

MACAL, C. M.; NORTH, M. J. Proceedings of the 2005 Winter Simulation Conference M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds. In: **37th Winter Simulation Conference**. [S.l.: s.n.], 2005. p. 1613–1620. Citation on page 31.

MADANI-ISFAHANI, M.; TAVAKKOLI-MOGHADDAM, R.; NADERI, B. Multiple cross-docks scheduling using two meta-heuristic algorithms. **Computers & Industrial Engineering**, v. 74, p. 129–138, 2014. Citations on pages 36 and 39.

MAGABLEH, G. M.; ROSSETTI, M. D.; MASON, S. Modeling and analysis of a generic cross-docking facility. In: WINTER SIMULATION CONFERENCE. **Proceedings of the 37th conference on Winter simulation**. [S.l.], 2005. p. 1613–1620. Citations on pages 31 and 38.

MAKNOON, M.; SOUMIS, F.; BAPTISTE, P. An integer programming approach to scheduling the transshipment of products at cross-docks in less-than-truckload industries. **Computers & Operations Research**, v. 82, p. 167–179, 2017. Citations on pages 37 and 86.

MAKNOON, Y.; LAPORTE, G. Vehicle routing with cross-dock selection. **Computers & Operations Research**, v. 77, p. 254–266, 2017. Citations on pages 55 and 86.

MARTELLO, S.; PISINGER, D.; VIGO, D. The three-dimensional bin packing problem. **Operations Research**, v. 48, p. 256–267, 2000. Citations on pages [46](#) and [53](#).

MCWILLIAMS, D. Simulation-based scheduling for parcel consolidation terminals: a comparison of iterative improvement and simulated annealing. In: **37th Winter Simulation Conference**. [S.l.: s.n.], 2005. p. 2087–2093. Citations on pages [31](#), [38](#), and [39](#).

MERVIS, J. The information highway gets physical. **Science**, n. 6188, p. 1104–1107, 2014. Citation on page [21](#).

MIAO, Z.; LIM, A.; MA, H. Truck dock assignment problem with operational time constraint within crossdocks. **European Journal of Operational Research**, v. 192, p. 105–115, 2009. Citations on pages [36](#), [39](#), and [86](#).

MILLER, C. E.; TUCKER, A. W.; ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. **Journal of the ACM (JACM)**, ACM, v. 7, p. 326–329, 1960. Citation on page [109](#).

MOGHADAM, S. S.; GHOMI, S. M. T. F.; KARIMI, B. Vehicle routing scheduling problem with cross docking and split deliveries. **Computers & Chemical Engineering**, Elsevier, v. 69, p. 98–107, 2014. Citations on pages [55](#) and [86](#).

MOHANTY, B. B.; MATHUR, K.; IVANCIC, N. Value considerations in three-dimensional packing — a heuristic procedure using the fractional knapsack problem. **European Journal of Operational Research**, Elsevier, v. 74, p. 143–151, 1994. Citations on pages [43](#), [50](#), and [53](#).

MOHTASHAMI, A.; TAVANA, M.; SANTOS-ARTEAGA, F. J.; FALLAHIAN-NAJAFABADI, A. A novel multi-objective meta-heuristic model for solving cross-docking scheduling problems. **Applied Soft Computing Journal**, v. 31, p. 30–47, 2015. Citation on page [56](#).

MOLAVI, D.; SHAHMARDAN, A.; SAJADIEH, M. S. Truck scheduling in a cross docking systems with fixed due dates and shipment sorting. **Computers & Industrial Engineering**, v. 117, p. 29–40, 2018. Citations on pages [37](#) and [39](#).

MONMA, C. L. Sequencing to minimize the maximum job cost. **Operations Research**, INFORMS, v. 28, p. 942–951, 1980. Citation on page [34](#).

MONTREUIL, B. Toward a Physical Internet: meeting the global logistics sustainability grand challenge. **Logistics Research**, n. 2-3, p. 71–87, 2011. Citations on pages [15](#), [21](#), and [22](#).

MONTREUIL, B.; MELLER, R. D.; BALLOT, E. **Physical Internet Foundations**. [S.l.], 2012. 11 p. Citation on page [21](#).

MORABITO, R.; ARENALES, M. Staged and constrained two-dimensional guillotine cutting problems: A new approach. In: **Technical Report: Notas do ICMSC 126**. [S.l.]: Universidade de São Paulo São Carlos, SP, Brazil, 1992. Citation on page [43](#).

_____. An AND/OR-graph approach to the Container Loading Problem. **International Transactions in Operational Research**, v. 1, p. 59–73, 1994. Citations on pages [43](#) and [53](#).

- MORABITO, R.; MORALES, S. A simple and effective recursive procedure for the manufacturer's pallet loading problem. **Journal of the Operational Research Society**, v. 49, p. 819–828, 1998. Citation on page 44.
- MORAIS, V. W. C.; MATEUS, G. R.; NORONHA, T. F. Iterated local search heuristics for the Vehicle Routing Problem with Cross-Docking. **Expert Systems with Applications**, v. 41, p. 7495–7506, 2014. Citation on page 55.
- MOURA, A.; OLIVEIRA, J. F. A grasp approach to the container-loading problem. **IEEE Intelligent Systems**, v. 20, n. 4, p. 50–57, 2005. Citations on pages 42, 45, 46, and 53.
- _____. An integrated approach to the vehicle routing and container loading problems. **OR Spectrum**, v. 31, p. 775–800, 2009. Citation on page 49.
- NAPOLITANO, M. Cross dock fuels growth at Dots. **Logistics Management**, v. 50, p. 54–9, 2011. Citation on page 28.
- NGUYEN, L. X.; CHAMBERS, M.; GOWOROWSKA, J.; RICK, C.; SEDOR, J.; BERG, J.; FARBER-DEANDA, M.; FORD, C.; JONES, C.; KATSIKIDES, N.; LIU, M.; MALLIK, A.; MENEGUS, D.; SMALLEN, D.; TORRES, C. **Freight Facts and Figures 2015**. [S.l.], 2015. 85 p. Citation on page 21.
- OPTIMIZATION, G. Inc., “gurobi optimizer reference manual,” 2017. URL: www.gurobi.com/documentation/7.5/refman, 2018. Citation on page 92.
- ÖZENER, O. Ö.; ERGUN, Ö.; SAVELSBERGH, M. Lane-Exchange Mechanisms for Truckload Carrier Collaboration. **Transportation Science**, v. 45, p. 1–17, 2011. Citation on page 59.
- PADBERG, M. Packing small boxes into a big box. **Mathematical Methods of Operations Research**, v. 52, p. 1–21, 2000. Citations on pages 43 and 45.
- PAQUAY, C.; LIMBOURG, S.; SCHYNS, M.; OLIVEIRA, J. F. Mip-based constructive heuristics for the three-dimensional bin packing problem with transportation constraints. **International Journal of Production Research**, v. 56, p. 1581–1592, 2018. Citations on pages 48 and 53.
- _____. Mip-based constructive heuristics for the three-dimensional bin packing problem with transportation constraints. **International Journal of Production Research**, Taylor & Francis, v. 56, p. 1581–1592, 2018. Citation on page 107.
- PARRAGH, S. N.; DOERNER, K. F.; HARTL, R. F. A survey on pickup and delivery problems. **Journal für Betriebswirtschaft**, Springer, v. 58, p. 21–51, 2008. Citation on page 49.
- PESKIR, G.; SHIRYAEV, A. **Optimal stopping and free-boundary problems**. [S.l.]: Springer, 2006. Citation on page 65.
- PISINGER, D. A tree search algorithm for the container loading problem. **Ricerca Operativa**, p. 394–410, 1998. Citation on page 44.
- _____. Heuristics for the container loading problem. **European journal of Operational Research**, v. 141, p. 382–392, 2002. Citations on pages 23, 42, 44, and 53.
- POCHET, Y.; WOLSEY, L. A. **Production planning by mixed integer programming**. [S.l.]: Springer Science & Business Media, 2006. Citation on page 107.

POLLARIS, H.; BRAEKERS, K.; CARIS, A.; JANSSENS, G. K.; LIMBOURG, S. Vehicle routing problems with loading constraints: state-of-the-art and future directions. **OR Spectrum**, v. 37, p. 297–330, 2015. Citation on page [50](#).

PONGCHAIRERKS, P.; KACHITVICHYANUKUL, V. A two-level particle swarm optimisation algorithm on job-shop scheduling problems. **International Journal of Operational Research**, Inderscience Publishers, p. 390–411, 2009. Citation on page [33](#).

POTVIN, J.-Y.; NAUD, M.-A. Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. **Journal of the operational research society**, v. 62, p. 326–336, 2011. Citation on page [59](#).

RAMOS, A. G.; SILVA, E.; OLIVEIRA, J. F. A new load balance methodology for container loading problem in road transportation. **European Journal of Operational Research**, v. 266, p. 1140–1152, 2018. Citation on page [45](#).

REN, J.; TIAN, Y.; SAWARAGI, T. A tree search method for the container loading problem with shipment priority. **European Journal of Operational Research**, Elsevier B.V., v. 214, p. 526–535, 2011. Citations on pages [44](#), [45](#), [51](#), and [53](#).

ROHRER, M. Simulation and cross docking. In: **Winter Simulation Conference Proceedings, 1995**. [S.l.: s.n.], 1995. p. 846–849. Citation on page [31](#).

ROPKE, S.; PISINGER, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. **Transportation science**, Informs, v. 40, p. 455–472, 2006. Citations on pages [50](#) and [103](#).

SANTOS, F. A.; MATEUS, G. R.; CUNHA, A. S. D. The pickup and delivery problem with cross-docking. **Computers & Operations Research**, v. 40, p. 1085–1093, 2013. Citation on page [55](#).

SAVELSBERGH, M.; WOENSEL, T. V. 50th anniversary invited article – city logistics: Challenges and opportunities. **Transportation Science**, INFORMS, v. 50, p. 579–590, 2016. Citation on page [103](#).

SAVELSBERGH, M.; WOENSEL, T. V. City Logistics: Challenges and Opportunities. **Transportation Science**, p. 1–19, 2016. Citation on page [22](#).

SAVELSBERGH, M. W.; SOL, M. The general pickup and delivery problem. **Transportation science**, INFORMS, v. 29, p. 17–29, 1995. Citation on page [103](#).

SHAKERI, M.; LOW, M. Y. H.; TURNER, S. J.; LEE, E. W. A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. **Computers & Operations Research**, v. 39, p. 2564–2577, 2012. Citations on pages [36](#), [37](#), [38](#), [39](#), and [86](#).

SHENG, L.; HONGXIA, Z.; XISONG, D.; CHANGJIAN, C. A heuristic algorithm for container loading of pallets with infill boxes. **European Journal of Operational Research**, v. 252, p. 728–736, 2016. Citations on pages [46](#) and [53](#).

SHENG, L.; WEI, T.; ZHIYUAN, X.; XIWEI, L. A tree search algorithm for the container loading problem. **Computers & Industrial Engineering**, Elsevier Ltd, v. 75, n. 2014, p. 20–30, 2013. Citations on pages [44](#) and [53](#).

SHIGUEMOTO, A. L.; NETTO, U. S. C.; BAUAB, G. H. S. An efficient hybrid meta-heuristic for a cross-docking system with temporary storage. **International Journal of Production Research**, v. 52, p. 1231–1239, 2014. Citation on page [37](#).

SIGURD, M.; PISINGER, D.; SIG, M. **The pickup and delivery problem with time windows and precedences**. [S.l.]: Datalogisk Institut, Københavns Universitet, 2000. Citation on page [103](#).

SIMS, R.; SCHAEFFER, R.; CREUTZIG, F.; CRUZ-NÚÑEZ, X.; D'AGOSTO, M.; DIMITRIU, D.; MEZA, M. J. F.; FULTON, L.; KOBAYASHI, S.; LAH, O.; MCKINNON, A.; NEWMAN, P.; OUYANG, M.; SCHAUER, J. J.; SPERLING, D.; TIWARI, G. Climate Change 2014: Mitigation of Climate Change. In: EDENHOFER, O.; PICHES-MADRUGA, R.; SOKONA, Y.; FARAHANI, E.; KADNER, S.; SEYBOTH, K.; ADLER, A.; BAUM, I.; BRUNNER, S.; EICKEMEIER, P.; KRIEMANN, B.; SAVOLAINEN, J.; SCHLÖMER, S.; STECHOW, C. von; ZWICKEL, T.; MINX, J. (Ed.). **Working Group III Contribution to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change**. Cambridge, United Kingdom and New York, NY, USA.: Cambridge University Press, 2014. chap. 8, p. 599–670. Citation on page [21](#).

STALK, G.; EVANS, P.; SHULMAN, L. E. Competing on capabilities: the new rules of corporate strategy. **Harvard Business Review**, v. 70, p. 57–69, 1992. Citation on page [28](#).

STENGER, A.; VIGO, D.; ENZ, S.; SCHWIND, M. An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. **Transportation Science**, v. 47, p. 64–80, 2013. Citation on page [59](#).

STEPHAN, K.; BOYSEN, N. Cross-docking. **Journal of Management Control**, v. 22, p. 129–137, 2011. Citations on pages [28](#), [30](#), and [86](#).

STORN, R.; PRICE, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. **Journal of global optimization**, Springer, n. 4, p. 341–359, 1997. Citation on page [45](#).

SWEENEY, P. E.; PATERNOSTER, E. R. Cutting and Packing Problems: A Categorized, Application-Orientated Research Bibliography. **Journal of the Operational Research Society**, v. 43, p. 691–706, 1992. Citation on page [40](#).

TAKAHARA, S. A Multi-start Local Search Approach to the Multiple Container Loading Problem. In: BEDNORZ, W. (Ed.). **Advances in Greedy Algorithms**. Vienna Austria: I-TECH Education and Publishing, 2008. p. 55–68. Citations on pages [43](#), [51](#), and [53](#).

TAKAHARA, S.; MIYAMOTO, S. An Evolutionary Approach for the Multiple Container Loading Problem. In: **Hybrid Intelligent Systems, Fifth International Conference on**. [S.l.]: IEEE, 2005. p. 1–6. Citation on page [53](#).

TANIGUCHI, E.; THOMPSON, R. G. **City logistics: Mapping the future**. [S.l.]: CRC Press, Taylor & Francis, 2015. Citation on page [103](#).

TARANTILIS, C.; ZACHARIADIS, E.; KIRANOUDIS, C. A hybrid metaheuristic algorithm for the multi-depot vehicle routing problem with time windows. **Intelligent Transportation Systems, IEEE Transactions on**, n. 2, p. 255–271, 2009. Citation on page [49](#).

TAVANA, M.; KHALILI-DAMGHANI, K.; SANTOS-ARTEAGA, F. J.; ZANDI, M.-H. Drone shipping versus truck delivery in a cross-docking system with multiple fleets and products. **Expert Systems with Applications**, v. 72, p. 93–107, 2017. Citations on pages 37 and 39.

TIAN, T.; ZHU, W.; LIM, A.; WEI, L. The multiple container loading problem with preference. **European Journal of Operational Research**, v. 248, p. 84–94, 2016. Citations on pages 47 and 53.

TINARELLI, G. U.; ADDONIZIO, M. Un problema di caricamento di containers. In: **Proc. AIRO congress**. [S.l.: s.n.], 1978. Citation on page 46.

TOOTKALEH, S. R.; GHOMI, S. M. T. F.; SAJADIEH, M. S. Cross dock scheduling with fixed outbound trucks departure times under substitution condition. **Computers & Industrial Engineering**, v. 92, p. 50–56, 2016. Citations on pages 36, 39, and 86.

TSAI, D. M. **Modeling and analysis of three-dimensional robotic palletizing systems for mixed carton sizes**. 388 p. Phd Thesis (PhD) — Iowa State University, 1987. Citation on page 43.

TSAI, J.-F.; LI, H.-L. A global optimization method for packing problems. **Journal of the Operational Research Society**, v. 38, p. 419–425, 2006. Citation on page 43.

VAHDANI, B.; TAVAKKOLI-MOGHADDAM, R.; ZANDIEH, M.; RAZMI, J.; REZA, T.-M. Vehicle routing scheduling using an enhanced hybrid optimization approach. **Journal of Intelligent Manufacturing**, v. 23, p. 759–774, 2012. Citations on pages 54 and 86.

VAHDANI, B.; ZANDIEH, M. Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. **Computers & Industrial Engineering**, Elsevier Ltd, v. 46, p. 650–666, 2010. Citations on pages 32, 33, 38, 39, and 85.

VIDAL, T.; MACULAN, N.; OCHI, L. S.; PENNA, P. H. V. Large neighborhoods with implicit customer selection for vehicle routing problems with profits. **Transportation Science**, v. 50, p. 720–734, 2015. Citation on page 59.

VIS, I. F. A.; ROODBERGEN, K. J. Positioning of goods in a cross-docking environment. **Computers & Industrial Engineering**, v. 54, p. 677–689, 2008. Citations on pages 31, 32, and 38.

WALLER, M. A.; CASSADY, C. R.; OZMENT, J. Impact of cross-docking on inventory in a decentralized retail supply chain. **Transportation Research Part E: Logistics and Transportation Review**, v. 42, p. 359–382, 2006. Citation on page 28.

WANG, N.; LIM, A.; ZHU, W. A multi-round partial beam search approach for the single container loading problem with shipment priority. **International Journal of Production Economics**, Elsevier, v. 145, p. 531–540, 2013. Citations on pages 42, 44, 45, 51, and 53.

WANG, X.; KOPFER, H. Collaborative transportation planning of less-than-truckload freight. **OR spectrum**, v. 36, p. 357–380, 2014. Citation on page 60.

WANG, X.; KOPFER, H.; GENDREAU, M. Operational transportation planning of freight forwarding companies in horizontal coalitions. **European Journal of Operational Research**, v. 237, p. 1133–1141, 2014. Citations on pages 59 and 60.

- WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European Journal of Operational Research**, v. 183, p. 1109–1130, 2007. Citations on pages [12](#), [23](#), [40](#), and [41](#).
- WEI, L.; ZHU, W.; LIM, A. A goal-driven prototype column generation strategy for the multiple container loading cost minimization problem. **European Journal of Operational Research**, v. 241, p. 39–49, 2015. Citations on pages [47](#) and [53](#).
- WEN, M.; LARSEN, J.; CLAUSEN, J.; CORDEAU, J.-F.; LAPORTE, G. Vehicle routing with cross-docking. **Journal of the Operational Research Society**, v. 60, p. 1708–1718, 2009. Citations on pages [54](#) and [86](#).
- WISITTIPANICH, W.; HENGMEECHAI, P. Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization. **Computers & Industrial Engineering**, v. 113, p. 793–802, 2017. Citations on pages [33](#) and [39](#).
- WITT, C. Crossdocking: concepts demand choice. **Material Handling Engineering**, v. 53, p. 44–49, 1998. Citation on page [28](#).
- WOOLSON, R. Wilcoxon signed-rank test. **Wiley encyclopedia of clinical trials**, p. 1–3, 2007. Citation on page [69](#).
- YU, V. F.; JEWPNYA, P.; REDI, A. A. N. P. Open vehicle routing problem with cross-docking. **Computers & Industrial Engineering**, v. 94, p. 6–17, 2016. Citations on pages [57](#), [86](#), and [87](#).
- YU, W. **Operational strategies for cross-docking systems**. 279 p. Phd Thesis (Ph. D.) — Iowa State University, 2002. Citations on pages [32](#), [35](#), [38](#), and [39](#).
- YU, W.; EGBELU, P. J. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. **European Journal of Operational Research**, v. 184, p. 377–396, 2008. Citations on pages [32](#), [33](#), [35](#), [38](#), [39](#), and [85](#).
- ZACHARIADIS, E.; TARANTILIS, C.; KIRANOUDIS, C. A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. **European Journal of Operational Research**, Elsevier, v. 195, p. 729–743, 2009. Citation on page [49](#).
- ZHANG, T.; SAHARIDIS, G. K. D.; THEOFANIS, S.; BOILE, M. Scheduling of Inbound and Outbound Trucks at Cross-Docks. **Transportation Research Record: Journal of the Transportation Research Board**, p. 9–16, 2010. Citations on pages [35](#), [39](#), and [86](#).
- ZHAO, X.; BENNELL, J. A.; BEKTAŞ, T.; DOWSLAND, K. A comparative review of 3D container loading algorithms. **International Transactions in Operational Research**, v. 23, p. 287–320, 2016. Citation on page [41](#).
- ZHENG, J.-N.; CHIEN, C.-F.; GEN, M. Multi-objective multi-population biased random-key genetic algorithm for the 3-D container loading problem. **Computers & Industrial Engineering**, v. 89, p. 80–87, 2015. Citations on pages [43](#) and [53](#).
- ZHU, W.; HUANG, W.; LIM, A. A prototype column generation strategy for the multiple container loading problem. **European Journal of Operational Research**, Elsevier B.V., v. 223, p. 27–39, 2012. Citations on pages [46](#), [47](#), and [53](#).

ZHU, W.; LIM, A. A new iterative-doubling Greedy-Lookahead algorithm for the single container loading problem. **European Journal of Operational Research**, v. 222, p. 408–417, 2012. Citations on pages [51](#) and [53](#).

ZHU, W.; OON, W.-C.; LIM, A.; WENG, Y. The six elements to block-building approaches for the single container loading problem. **Applied Intelligence**, v. 37, p. 431–445, 2012. Citations on pages [42](#) and [44](#).

ZIBAEI, S.; HAFEZALKOTOB, A.; GHASHAMI, S. S. Cooperative vehicle routing problem: an opportunity for cost saving. **Journal of Industrial Engineering International**, v. 12, p. 271–286, 2016. Citation on page [60](#).

ZISSIS, D.; AKTAS, E.; BOURLAKIS, M. Collaboration in urban distribution of online grocery orders. **International Journal of Logistics Management**, v. 29, p. 1196–1214, 2018. Citation on page [60](#).

DETAILS ON THE EXPERIMENTS FOR THE NETWORK SCHEDULING PROBLEM

Here, we detail some results related to Chapter 4. Particularly, our experiments showed that the difference in solver performance for *narrow* and *wide* instances is not expressive, although it consistently favours the *wide* case (see Table 32). The table also shows primal and dual bounds and the results are as expected, *wide* instances present better values, since the time window are more flexible. Gaps are consistently better, with an additional proof of optimality for the *wide* case. Moreover, optimal solutions were found faster in every case for the *wide* set.

Tables (33)–(38) details results regarding primal bound, gap and time to solve for *narrow* and *wide* instances. The results compare the decomposition variations NSCD-B-MP-I, II and III with and without the pre-processing of the time windows in the sub-problems. In general, incorporating the pre-processing was ineffective.

Table 32 – Comparison between *narrow* and *wide* instances solved by the model NSCD. We present primal bound (UB) and dual bound (LB), gap and time to solve (s). The gap is presented as $100 \frac{UB-LB}{UB}$. NA indicates that no feasible solution was found within the time limit.

Inst	Primal bound		Dual bound		Gap (%)		Time (s)	
	Narrow	Wide	Narrow	Wide	Narrow	Wide	Narrow	Wide
A-n32-k5	3794.20	3638.50	2775.84	2714.68	26.84	25.39	3600	3600
A-n33-k5	2800.54	2140.62	1225.24	1203.67	56.25	43.77	3600	3600
A-n33-k6	3547.39	3156.36	2120.98	2133.38	40.21	32.41	3600	3600
A-n34-k5	2940.39	2816.15	2099.73	2206.74	28.59	21.64	3600	3600
A-n36-k5	3273.17	3068.82	1802.86	1766.72	44.92	42.43	3600	3600
A-n37-k5	2270.46	2114.36	2270.46	2114.36	0.00	0.00	2287	1291
A-n37-k6	3077.35	2900.62	2097.83	2110.49	31.83	27.24	3600	3600
A-n38-k5	1721.78	1670.34	1721.78	1670.34	0.00	0.00	2	1
A-n39-k5	4800.77	4314.89	1583.29	1574.07	67.02	63.52	3600	3600
A-n39-k6	3998.34	3675.63	2660.90	2533.98	33.45	31.06	3600	3600
A-n44-k6	4728.43	4487.17	1458.72	1458.78	69.15	67.49	3600	3600
A-n45-k6	6140.23	5522.96	1671.98	1671.80	72.77	69.73	3600	3600
A-n45-k7	2659.36	2387.25	2659.36	2387.25	0.00	0.00	369	16
A-n46-k7	2891.16	2646.16	2099.27	2011.61	27.39	23.98	3600	3600
A-n48-k7	7400.74	6817.61	2271.29	2271.63	69.31	66.68	3600	3600
A-n53-k7	7050.52	6421.32	1976.97	1976.48	71.96	69.22	3600	3600
A-n54-k7	4043.69	3670.20	3244.25	3670.20	19.77	0.00	3600	1152
A-n55-k9	5784.16	5199.58	2816.31	2737.06	51.31	47.36	3600	3600
A-n60-k9	13547.31	12163.89	2867.97	2867.03	78.83	76.43	3600	3600
A-n61-k9	2404.64	2208.98	2404.64	2208.98	0.00	0.00	76	50
A-n62-k8	3068.42	2928.40	2739.79	2735.71	10.71	6.58	3600	3600
A-n63-k10	2235.76	2161.94	2235.76	2161.94	0.00	0.00	461	458
A-n63-k9	NA	NA	NA	NA	NA	NA	NA	NA
A-n64-k9	7581.08	6834.80	2316.02	2316.31	69.45	66.11	3600	3600
A-n65-k9	14679.93	13342.53	2677.62	2677.85	81.76	79.93	3600	3600
A-n69-k9	5601.93	5118.32	2461.49	2461.40	56.06	51.91	3600	3600
A-n80-k10	3371.30	3045.58	2811.66	2788.84	16.60	8.43	3600	3600
Average	4823.58	4402.04	2272.00	2247.36	39.39	35.44	3031	2884

Table 33 – Comparison of primal bound of *narrow* instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III *with* time window preprocessing and *without* time window preprocessing.

Primal bound	NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III	
	With	Without	With	Without	With	Without
A-n32-k5	3947.23	3947.23	4024.87	3965.68	4024.87	4024.87
A-n33-k5	2723.88	2723.88	2973.65	2973.65	2857.49	2973.65
A-n33-k6	3547.39	3547.39	3547.39	3547.39	3547.39	3547.39
A-n34-k5	2940.39	2940.39	2940.39	2940.39	2940.39	2940.39
A-n36-k5	3412.40	3412.40	3410.09	3412.40	3273.17	3410.09
A-n37-k5	2563.78	2616.31	2270.46	2270.46	2270.46	2270.46
A-n37-k6	3169.23	3156.42	3080.17	3080.17	3159.14	3132.25
A-n38-k5	1721.78	1721.78	1721.78	1721.78	1721.78	1721.78
A-n39-k5	4742.26	4742.26	4742.26	4742.26	4742.26	4742.26
A-n39-k6	4598.81	4706.31	3998.34	3998.34	3998.34	3998.34
A-n44-k6	4694.89	4694.89	4694.89	4694.89	4694.89	4694.89
A-n45-k6	6128.45	6128.45	6128.45	6128.45	6128.45	6128.45
A-n45-k7	2659.36	2659.36	2659.36	2659.36	2659.36	2659.36
A-n46-k7	2957.11	2957.11	2846.12	2860.40	2812.97	2812.97
A-n48-k7	6674.33	6674.33	6676.82	6676.82	6652.78	6676.82
A-n53-k7	8195.81	8177.51	7813.81	7590.15	6978.83	6978.83
A-n54-k7	4043.69	4043.69	4043.69	4043.69	4043.69	4043.69
A-n55-k9	6902.91	6990.96	5777.38	5777.38	5777.38	5777.38
A-n60-k9	8123.59	8123.59	9431.36	9461.29	9304.47	9225.94
A-n61-k9	3256.68	3256.68	2404.64	2404.64	2404.64	2404.64
A-n62-k8	3496.14	3213.13	3068.42	3068.42	3068.42	3068.42
A-n63-k10	2308.40	2308.40	2235.76	2235.76	2235.76	2235.76
A-n63-k9	8452.01	8326.47	8558.73	8282.84	7788.15	7770.14
A-n64-k9	8148.94	8121.01	7560.72	6951.84	7548.57	7113.77
A-n65-k9	14340.45	14340.45	12616.89	12125.72	12958.34	9853.34
A-n69-k9w	9172.66	8971.65	5579.52	5579.52	5579.52	5579.52
A-n80-k10	3371.30	3371.30	3371.30	3371.30	3371.30	3371.30
Average	5047.92	5032.35	4747.31	4687.59	4686.77	4561.36

Table 34 – Comparison of primal bound of *wide* instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III *with* time window pre-processing and *without* time window pre-processing.

Primal bound	NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III	
	With	Without	With	Without	With	Without
A-n32-k5	3754.98	3754.98	3825.06	3825.06	3754.43	3825.06
A-n33-k5	2278.35	2206.33	2398.80	2416.14	2416.14	2319.56
A-n33-k6	3156.36	3156.36	3156.36	3156.36	3156.36	3156.36
A-n34-k5	2816.15	2816.15	2816.15	2816.15	2816.15	2816.15
A-n36-k5	3202.68	3167.73	3068.82	3068.82	3068.82	3068.82
A-n37-k5	2422.11	2422.11	2114.36	2114.36	2114.36	2114.36
A-n37-k6	3006.09	2996.87	2984.52	2989.47	2900.62	2900.62
A-n38-k5	1670.34	1670.34	1670.34	1670.34	1670.34	1670.34
A-n39-k5	4310.87	4310.87	4310.87	4310.87	4310.87	4310.87
A-n39-k6	4151.10	4157.02	3675.63	3675.63	3675.63	3675.63
A-n44-k6	4487.17	4487.17	4487.17	4487.17	4487.17	4487.17
A-n45-k6	5506.91	5506.91	5506.91	5506.91	5506.91	5506.91
A-n45-k7	2387.25	2387.25	2387.25	2387.25	2387.25	2387.25
A-n46-k7	2745.75	2762.44	2615.76	2627.77	2601.34	2580.97
A-n48-k7	6199.65	6199.65	6239.22	6239.22	6126.17	6192.52
A-n53-k7	7236.66	7376.03	6351.91	6351.91	7052.89	6351.91
A-n54-k7	3670.20	3670.20	3670.20	3670.20	3670.20	3670.20
A-n55-k9	7178.84	6438.25	5199.58	5199.58	5199.58	5199.58
A-n60-k9	8457.56	8467.17	8786.71	8944.93	8367.63	8497.30
A-n61-k9	2909.16	3097.61	2208.98	2208.98	2208.98	2208.98
A-n62-k8	3110.37	3110.37	2928.40	2928.40	2928.40	2928.40
A-n63-k10	2232.40	2232.40	2161.94	2161.94	2161.94	2161.94
A-n63-k9	7846.91	7928.52	8386.74	8131.00	7404.12	7897.20
A-n64-k9	7216.10	7307.22	6758.06	6622.96	6857.76	6668.17
A-n65-k9	13165.12	12304.30	10300.62	12368.39	10667.70	10990.59
A-n69-k9	8075.45	8314.35	5118.32	5118.32	5118.32	5118.32
A-n80-k10	3045.58	3045.58	3045.58	3045.58	3045.58	3045.58
Average	4675.56	4640.53	4302.75	4371.99	4284.28	4287.07

Table 35 – Comparison of gap, $(100 \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}})$, of *narrow* instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III *with* time window preprocessing and *without* time window preprocessing.

Gap	NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III	
	With	Without	With	Without	With	Without
A-n32-k5	63.17	63.17	57.28	54.59	62.06	61.51
A-n33-k5	57.63	57.63	54.19	54.24	54.68	57.67
A-n33-k6	0.00	0.00	0.00	0.00	0.00	0.00
A-n34-k5	37.76	38.20	0.00	0.00	0.00	0.00
A-n36-k5	80.34	80.13	21.71	24.76	0.00	31.81
A-n37-k5	73.33	74.05	0.00	0.00	0.00	0.00
A-n37-k6	77.89	77.89	12.40	12.93	27.19	24.39
A-n38-k5	32.20	31.24	0.00	0.00	0.00	0.00
A-n39-k5	0.00	0.00	0.83	0.83	0.41	0.41
A-n39-k6	77.65	78.20	0.01	0.00	0.01	0.00
A-n44-k6	3.34	4.34	3.34	4.34	18.02	18.02
A-n45-k6	4.53	4.60	4.78	4.91	22.82	20.10
A-n45-k7	0.00	0.00	0.00	0.00	0.00	0.00
A-n46-k7	73.43	73.43	21.65	14.23	16.45	18.98
A-n48-k7	88.37	88.34	66.97	66.47	70.48	68.84
A-n53-k7	95.53	95.52	51.51	41.67	51.08	38.71
A-n54-k7	0.00	0.00	0.00	0.00	0.00	0.00
A-n55-k9	84.72	84.92	0.01	0.00	0.00	0.01
A-n60-k9	84.59	84.56	76.25	76.10	75.40	78.43
A-n61-k9	75.27	75.38	0.00	0.00	0.00	0.00
A-n62-k8	69.16	66.25	0.00	0.01	0.01	0.01
A-n63-k10	32.48	32.50	0.01	0.01	0.01	0.01
A-n63-k9	85.67	85.36	73.59	72.52	75.26	75.13
A-n64-k9	91.09	91.24	50.17	46.17	55.55	50.91
A-n65-k9	96.96	96.96	67.16	76.60	81.01	75.70
A-n69-k9	90.75	90.50	8.30	7.26	6.91	12.41
A-n80-k10	0.00	0.00	0.00	0.00	0.00	0.00
Average	54.66	54.61	21.12	20.65	22.86	23.45

Table 36 – Comparison of gap, $(100 \frac{\text{upper bound} - \text{lower bound}}{\text{upper bound}})$, of *wide* instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III *with* time window pre-processing and *without* time window pre-processing.

Gap	NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III	
	With	Without	With	Without	With	Without
A-n32-k5	61.35	61.38	51.90	53.26	55.13	57.24
A-n33-k5	49.70	47.78	36.20	37.09	45.13	42.50
A-n33-k6	0.00	0.00	0.00	0.00	0.00	0.00
A-n34-k5	37.41	38.65	0.00	0.01	0.00	0.00
A-n36-k5	79.05	78.87	0.00	0.00	13.51	24.91
A-n37-k5	72.02	72.08	0.00	0.00	0.00	0.00
A-n37-k6	76.70	76.65	23.10	20.60	14.28	15.16
A-n38-k5	27.04	26.95	0.00	0.00	0.00	0.00
A-n39-k5	0.00	0.00	0.46	0.46	0.00	0.00
A-n39-k6	74.74	75.04	0.01	0.00	0.01	0.01
A-n44-k6	0.24	0.24	0.24	0.24	0.27	0.27
A-n45-k6	3.40	3.40	2.67	2.73	22.76	17.97
A-n45-k7	0.00	0.00	0.00	0.00	0.00	0.00
A-n46-k7	70.89	71.04	15.62	17.06	11.15	12.12
A-n48-k7	86.55	86.55	68.44	67.96	72.30	72.99
A-n53-k7	94.68	94.76	31.85	30.97	38.83	26.19
A-n54-k7	0.00	0.00	0.00	0.00	0.00	0.00
A-n55-k9	87.39	85.39	0.01	0.00	0.00	0.00
A-n60-k9	85.67	85.72	71.01	72.30	76.80	76.28
A-n61-k9	72.49	74.34	0.00	0.00	0.00	0.00
A-n62-k8	63.95	64.06	0.00	0.01	0.01	0.00
A-n63-k10	30.01	30.02	0.01	0.01	0.01	0.01
A-n63-k9	84.35	84.61	75.69	73.57	74.15	77.47
A-n64-k9	89.86	89.73	52.49	50.38	52.91	51.36
A-n65-k9	96.67	96.43	55.12	60.56	58.32	87.38
A-n69-k9	89.51	89.78	8.36	8.48	14.43	11.41
A-n80-k10	0.00	0.00	0.00	0.00	0.00	0.00
Average	53.10	53.09	18.27	18.36	20.37	21.23

Table 37 – Comparison of time to solve (s) *narrow* instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III *with* time window preprocessing and *without* time window preprocessing.

Time	NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III	
	With	Without	With	Without	With	Without
A-n32-k5	3600	3600	3600	3600	3600	3600
A-n33-k5	3600	3600	3449	3462	3600	3600
A-n33-k6	2370	2379	495	511	552	586
A-n34-k5	3600	3600	458	761	677	582
A-n36-k5	3600	3600	3599	3596	3510	3599
A-n37-k5	3600	3600	72	73	49	52
A-n37-k6	3600	3600	3598	3598	3600	3600
A-n38-k5	3600	3600	4	4	5	5
A-n39-k5	1193	1229	650	646	533	518
A-n39-k6	3600	3600	2147	2189	3187	2454
A-n44-k6	296	301	416	463	319	394
A-n45-k6	1961	1817	1104	1142	1528	1663
A-n45-k7	42	47	6	5	6	6
A-n46-k7	3600	3600	3597	3597	3599	3599
A-n48-k7	3600	3600	3600	3600	3600	3600
A-n53-k7	3600	3600	3572	3562	3600	3600
A-n54-k7	318	316	17	20	22	19
A-n55-k9	3600	3600	1672	1763	1440	1873
A-n60-k9	3600	3600	3573	3600	3600	3600
A-n61-k9	3600	3600	8	8	9	9
A-n62-k8	3600	3600	128	149	92	105
A-n63-k10	3600	3601	502	520	532	550
A-n63-k9	3600	3600	3600	3596	3600	3600
A-n64-k9	3600	3600	3591	3600	3600	3600
A-n65-k9	3600	3600	3586	3588	3669	4069
A-n69-k9	3600	3600	3577	3577	3599	3599
A-n80-k10	478	590	21	24	24	28
Average	2913	2914	1876	1898	1931	1945

Table 38 – Comparison of time to solve (s) *wide* instances for decomposition strategies NSCD-B-MP-I, NSCD-B-MP-II and NSCD-B-MP-III *with* time window pre-processing and *without* time window pre-processing.

Time	NSCD-B-MP-I		NSCD-B-MP-II		NSCD-B-MP-III	
	With	Without	With	Without	With	Without
A-n32-k5	3600	3600	3600	3600	3600	3600
A-n33-k5	3600	3600	3579	3586	3600	3600
A-n33-k6	2179	1958	399	411	382	377
A-n34-k5	3599	3600	683	582	362	390
A-n36-k5	3600	3600	3460	3548	3600	3599
A-n37-k5	3600	3600	44	50	72	74
A-n37-k6	3600	3600	3598	3599	3599	3599
A-n38-k5	3600	3600	5	5	6	7
A-n39-k5	977	871	496	518	585	594
A-n39-k6	3600	3600	1605	1512	2093	2117
A-n44-k6	248	262	277	283	335	336
A-n45-k6	1216	1278	746	713	814	1072
A-n45-k7	29	31	5	5	6	6
A-n46-k7	3600	3599	3598	3597	3599	3599
A-n48-k7	3600	3600	3589	3589	3600	3600
A-n53-k7	3600	3600	3600	3585	3600	3600
A-n54-k7	316	319	16	20	17	18
A-n55-k9	3600	3600	1255	1225	1256	1123
A-n60-k9	3600	3600	3600	3578	3600	3600
A-n61-k9	3600	3600	7	8	8	8
A-n62-k8	3599	3600	118	138	106	132
A-n63-k10	3600	3600	518	570	487	560
A-n63-k9	3600	3600	3600	3598	3600	3600
A-n64-k9	3600	3600	3572	3600	3600	3600
A-n65-k9	3600	3600	3589	3540	3600	3600
A-n69-k9	3600	3600	3580	3582	3599	3599
A-n80-k10	354	463	20	26	18	22
Average	2864	2858	1821	1817	1842	1853