Analysis and discussion of single and multi-objective IP formulations for the Truck-to-dock Door Assignment Problem

Nicolas Forget
Johannes Kelper University Linz, Altenberger Straße 69, 4040 Linz – Austria
Elizabeth Gandibleux, Xavier Gandibleux,* Valentin Guy–Deroubaix, Awen Jacq–Bodet
Nantes Université, 2 Rue de la Houssinière BP 92208, 44322 Nantes Cedex 03 – France

September 16, 2024

Abstract:

This paper is devoted to the Truck-to-dock Door Assignment Problem. Two integer programming formulations introduced after 2009 are examined. Our review of the literature takes note of the criticisms and limitations addressed to the seminal work of 2009. Although the published adjustments that followed present strong argument and technical background, we have identified several errors, inaccuracies and incompleteness. This paper therefore brings a rigorous presentation and clarification of these two formulations. In particular, both are described, corrected if necessary, analyzed and discussed. From this study, a bi-objective variant derived from these two formulations is proposed, resulting in four formulations in total.

They have been implemented and the codes are open-source and available online. Numerical experiments on instances from the literature have been conducted. Among the numerical results collected, we underline the advantage shown by the multi-objective formulations for the quality of the solutions produced.

Keywords: Supply Chain Management; Cross Docking; Truck-to-dock Door Assignment Problem; Integer Programming; Multi-objective Optimization

1 Introduction

1.1 Background

In the context of supply chain operations, this paper addresses an operational problem encountered in a cross-docking warehouse. Cross-docking is a logistics technique that aims to accelerate goods delivery and increase supply chain efficiency. Considering a warehouse with a given shape, the term cross-docking expresses the process of receiving products on inbound dock doors and then transferring them directly across the cross-dock to outbound dock doors, with few or no storage time in between (see Figure 1). More precisely, incoming products arrive through means of transportation such as trucks, and are docked on inbound dock doors of the cross-dock terminal. Once incoming trucks have been

*Corresponding author (Xavier.Gandibleux@univ-nantes.fr).

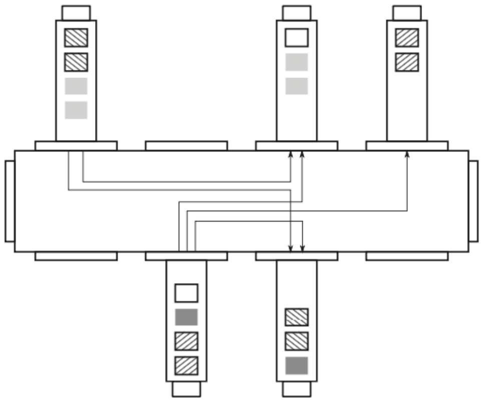


Figure 1: Illustration from Van Belle et al. (2012) of a cross-dock where the layout of the warehouse presents a I-shape. Ten docks are available (flat rectangles) to serve as inbound or outbound of products. Five trucks are docked (long rectangles). A pallet of a given product is represented by a rectangle with a given pattern. Each truck may convoy pallets of different products (represented by rectangles with different patterns). Pallets are moved into the warehouse (represented by arrows) from inbound to outbound docks.

docked, the pallets get unloaded, sorted and screened to identify their end destinations. Afterward, the pallets are moved to outbound dock doors of the cross-dock terminal using e.g. material handling devices such as forklifts.

Cross-docking requires close coordination among a company's supply chain partners, including its suppliers and freight carriers. According Jenkins (2023), this effort often pays off in multiple ways: companies can deliver products faster, minimize the need for warehouse space, optimize inventory control, and reduce transportation and labor costs.

Depending on the horizon of decisions considered (long term, mid term, short-term decisions), the literature presents several decision problems related to cross-docking (see e.g Nduwayo (2020); Van Belle et al. (2012)). They are grouped in strategic (e.g. location of cross-docks and layout design), tactical (e.g. cross-docking networks), and operational (e.g. vehicle routing, dock door assignment, truck scheduling and temporary storage) decision problems.

1.2 The Truck-to-dock Door Assignment Problem

In the context of cross-docking operations, the specific situation tackled is as follows: given a cross-dock warehouse composed of docks, and a planning of trucks where each truck is characterised by an arrival time, a departure time, and a number of pallets to transfert between trucks, the goal is to maximise the volume of cargo transiting through the cross-dock, and to minimize the time required to perform the cargo transfer operations. This statement leads to an assignment subproblem. Indeed, when an inbound or outbound truck arrives at the cross-dock, it has to be decided to which dock door the truck should be assigned. A good assignment can increase the productivity of the cross-dock and can decrease the handling costs. Also, this statement leads to a scheduling subproblem. Effectively, the dock doors are considered as resources (used by the trucks) that have to be scheduled over time. The problem decides on the succession of inbound and outbound trucks at the dock doors of a cross-dock: in short, where and when should the trucks be processed. Thus, a model describing this optimization problem must take into account (1) the arrival and departure of trucks, (2) the assignment of trucks to the docks, (3) the operational time for pallet shipment among the docks, and (4) the maximum amount of pallets that the cross-dock can support.

This optimization problem falls in the class of truck scheduling problem which deal with short-term decisions (operational). Among truck scheduling problems, those where scheduling of inbound and outbound trucks is the central problem are referred to as the *Truck-to-dock Door Assignment Problem (TDAP)* in the literature. It is a combinatorial optimisation problem known to belong to the class of $\mathcal{NP}$-hard problems (Miao, Lim, and Ma, 2009). On the basis of the contributions of Lim, Miao, Rodrigues, and Xu (2005); Lim, Ma, and Miao (2006); Miao et al. (2009), the TDAP is the optimization problem addressed in this paper, and the model introduced in Miao et al. (2009) is our starting reference.

In the literature, the various studies related to the TDAP consider different assumptions and settings, for instance regarding the preemption (allowed or not), the processing time to load or unload a truck (fixed or not for all trucks), intermediate storage (allowed or not), etc. In the following, we consider the case (1) without preemption (loading/unloading operations cannot be interrupted and resumed at a later time), and (2) the processing time to load/unload trucks is fixed for all trucks. The case with/without intermediate storage is discussed in this paper.

1.3 Literature Review

Miao et al. (2009) have extended a truck scheduling problem previously proposed by Lim et al. (2005, 2006) in which the authors assumed that the trucks are loaded or unloaded during a fixed time window. This means that the optimization problem is reduced to determining at which dock door the trucks have to be processed. The length of these time windows can be interpreted as the time needed to load or unload a truck. The trucks can be assigned to any door and the capacity of the cross-dock is limited. Preemption is not allowed and trucks that cannot be served are penalized. The objective here is to minimize the operational cost (based on travel time) plus the cost of unfulfilled shipments. The authors formulate the problem with an *Integer Programming (IP)* model. This formulation is denoted “Formulation M” in the remaining of the paper. Numerical experiments on their mathematical model are carried out using CPLEX. Observing that the exact solver could not solve medium and large instances to optimality within the given time limit of 7200 seconds, they also proposed a heuristic approach using tabu search and a genetic algorithm.

Van Belle et al. (2012) present a comprehensive state-of-the-art of the cross-docking concept. First, the authors discuss on guidelines for the use and implementation of cross-docking. They describe characteristics that can be helpful to distinguish the different cross-dock types and provide an extensive review of the existing literature until 2012 on cross-docking. The papers discussed are classified based on the problem type tackled (e.g. internal transport type, temporary storage allowed or not, etc.), and promising directions to improve and extend the contributions are suggested.

Several PhD thesis related to decision problems on cross-docking have been defended (e.g. Ladier (2014); Nassief (2017); Nduwayo (2020); Zhang (2016); Zhu (2007)). For example, Nduwayo’s contributions (Nduwayo, 2020) are devoted to the *Cross-Docking Assignment Problem (CDAP)*, which consists in finding an assignment of origins to inbound doors and destinations to outbound doors that minimizes the total cost inside the cross-dock platform. The authors propose original *Mixed-Integer*

Programming models (Gelareh, Glover, Guemri, Hanafi, Nduwayo, and Todosijević, 2020), and conduct an extensive comparative analysis on benchmark instances from the literature. Additional contributions related to this particular problem can be found in Tsui and Chang (1992); Zhu, Hahn, Liu, and Guignard-Spielberg (2009) or again Meliàn-Batista (2024).

In a technical report available online, Gelareh, Goncalves, and Monemi (2015) underline weaknesses and shortcomings in Formulation M. They propose a revised IP formulation for the TDAP, denoted "Formulation G" in the rest of the paper. Several valid inequalities are also introduced, and exact separation algorithms are described for separating cuts for those leading to an exponential number of constraints. An efficient branch-and-cut algorithm solving real-life size instances in a reasonable time is provided. Numerical experiments show that in most cases, the optimal solution is identified at the root node without requiring any branching. The main contents of this report have been published in Gelareh, Monemi, Semet, and Goncalves (2016).

Kucukoglu and Ozturk (2017) consider a variant of the TDAP with product placement plans. To solve this problem, they propose an IP model where the goal is to find the truck-door assignment and product placement plans that minimize total travelling distance of the products.

Recently, Daquin, Allaoui, Goncalves, and Hsu (2021) published an adaptation of the Variable Neighborhood Search metaheuristic to solve the TDAP. However, this work is based on Formulation M, which is pointed out as incorrect since 2015 by Gelareh et al. (2015). Thus, an another paper (Gelareh, 2021) which refers to (Gelareh et al., 2015, 2016) has been published which criticizes Daquin et al. (2021) and recalls the shortcomings already discussed about Formulation M.

1.4 Contributions and Organisation of the Paper

Given the points of contention observed along the literature review, a careful reading of the four documents concerned (Gelareh, 2021; Gelareh et al., 2015, 2016; Miao et al., 2009) has been achieved. Inconsistencies in the arguments put forward in Gelareh (2021); Gelareh et al. (2015, 2016), as well as to vagueness and incompleteness in Formulation G were observed. In particular, the proposed amendments in (Gelareh, 2021) does not address accurately all deficiencies identified in Formulation M and, to the best of our capability, we could not replicate the results of the numerical experiment provided in Gelareh et al. (2016).

On this basis, the first contribution of this paper is an analysis and a discussion of Formulations M and G and their optimal solutions. The aim is to provide to the readers (i) an advanced understanding of the formulations, (ii) a corrected formulation of G, and (iii) an understanding of the optimal solutions collected with the two formulations. The second contribution is the proposition of a bi-objective variant of formulations M and G, respectively named 2M and 2G, where the formulation's abilities to simultaneously optimise independently the two conflicting objectives is explored. Finally, the third contribution of this paper consists of numerical experiments conducted with these four formulations over the set of instances found in the literature, which reveal conclusions that were not expected.

The paper is thus organised as follow. Section 2 presents the notations and the definitions of the parameters of the problem. In order to facilitate the discussions about formulations, the notations used for the formulations have been unified across the paper. Next, an illustrating example based on a toy instance is used to review the different values that parameters may take, and the type of solution returned. After that, formulation M is presented and discussed in Section 3, followed in Section 4 by a criticism of this formulation found in the literature. Section 5 introduces the Formulation G, and additional details about it are given in A. Finally, formulations 2M and 2G are introduced in Section 6. These four formulations have been implemented in Julia using JuMP as algebraic modelling language. The corresponding codes are given in B and are available online. Results of the numerical experiments are synthesized in Section 7, and all the quantitative results collected are reported in C. Finally, Section 8 gives a conclusion and draws several perspectives for future research.

2 Notations, Parameters, Example

2.1 Parameters

The following parameters are used in the formulations:

n: number of trucks

m: number of docks

a_i: arrival time of truck i $(1 \leq i \leq n)$

d_i: departure time of truck i $(1 \leq i \leq n)$

t_{kl}: operational time for pallets from dock k to dock l $(1 \leq k, l \leq m)$

f_{ij}: number of pallets transferring from truck i to truck j $(1 \leq i, j \leq n)$

c_{kl}: operational cost per unit time from dock k to dock l $(1 \leq k, l \leq m)$

p_{ij}: penalty cost per unit cargo from truck i to truck j $(1 \leq i, j \leq n)$

C: capacity of crossdock

δ_{ij}: simultaneous presence indicator of trucks i and j $(1 \leq i, j \leq n)$

τ_r: time marker r $(1 \leq r \leq 2n)$

The simultaneous presence indicator indicates whether trucks i and j can be assigned to a same dock (no temporal overlap) or not. It is established as follow[1]: for all $1 \leq i, j \leq n$, if $[a_i, d_i] \cap [a_j, d_j]$ is empty then $\delta_{ij} = 1$ meaning that trucks i and j may be assigned to a same dock, otherwise $\delta_{ij} = 0$. Furthermore, the following restrictions are adopted without loss of generality:

1. $f_{ij} \geq 0$ iff $d_j \geq a_i$ $(1 \leq i, j \leq n)$, otherwise $f_{ij} = 0$
   It means that truck i will transfer some cargo to truck j iff truck j departs no earlier than truck i arrives;

2. $a_i < d_i$ $(1 \leq i \leq n)$
   It implies that the arrival time of a truck is strictly smaller than its departure time;

3. $n > m$
   It considers the over-constrained condition of the cross-dock.

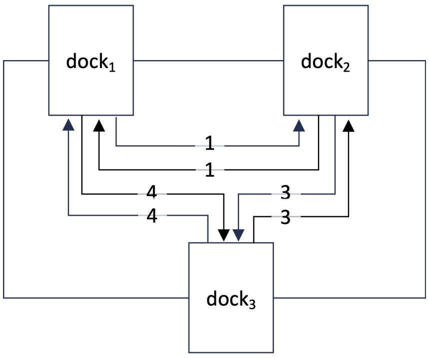
[1] originally in Miao et al. (2009), the authors state "$\delta_{ij} = 1$ iff truck i departs no later than truck j arrives; 0 otherwise".

In order to facilitate the expression of the set of capacity constraints, the vector τ is built from a_i and d_i as follows:

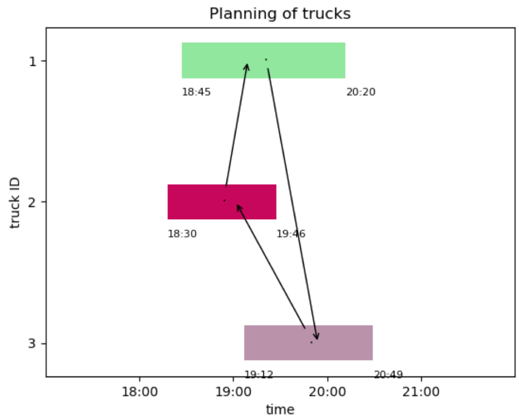
1. sort all a_i and d_i in an increasing order;
2. let τ_r $(1 \leq r \leq 2n)$ be these $2n$ numbers such that $\tau_1 \leq \tau_2 \leq \ldots \leq \tau_{2n}$.

2.2 Illustrative Example

Let's take a toy example illustrated by Figure 2, which shows the parameters to handle.



(a) The layout of the cross-dock. Values on arrows report the transfer time t_{kl} between docks k and l.



(b) A previsional planning of trucks with arrival/departure times. Arrows indicate a transfer of pallets f_{ij} between trucks i and j.

Figure 2: Data of the cross-dock and the planning of trucks for the illustration example

Regarding the cross-dock

- The layout illustrated in Figure 2a is composed of three docks ($m = 3$).
- The times of transfer between docks (in minutes) are given by:

$$t = \begin{pmatrix} 0 & 1 & 4 \\ 1 & 0 & 3 \\ 4 & 3 & 0 \end{pmatrix}$$

- There is no restriction on the capacity of the cross-dock in this example, i.e. $C = \infty$.

Regarding the previsional planning of trucks and transfers of pallets

- The scenario is composed of three trucks ($n = 3$).

- The hours of arrival and departure (in format hour.minute) are known and are, respectively for each truck:

  $a = (18.45,\ 18.30,\ 19.12)$
  $b = (20.20,\ 19.46,\ 20.49)$

- The number of pallets to transfer between the trucks is given by:

$$f = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \text{ i.e. } \begin{cases} \text{truck 1 delivers 1 pallet to truck 3} \\ \text{truck 2 delivers 1 pallet to truck 1} \\ \text{truck 3 delivers 1 pallet to truck 2} \end{cases}$$

Regarding the costs

- The operational costs and the penalty costs (in unit of value) are respectively:

$$c = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 2 \\ 1 & 2 & 0 \end{pmatrix} \text{ and } p = \begin{pmatrix} 0 & 0 & 52 \\ 24 & 0 & 0 \\ 0 & 23 & 0 \end{pmatrix}$$

Regarding the data derived from the parameters

- The matrix δ indicates the simultaneous presence indicator for all pairs of trucks; the value `1` means "no overlap" and the value `0` shows "an overlap":

$$\delta = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ i.e. } \begin{cases} \text{the time window of truck 1 overlaps with 2 and 3} \\ \text{the time window of truck 2 overlaps with 1 and 3} \\ \text{the time window of truck 3 overlaps with 1 and 2} \end{cases}$$

- The vector τ collects all time markers (i.e. hours of arrival and departure to consider) in one vector, where the values are sorted in increasing order:

  $\tau = (18.30,\ 18.45,\ 19.12,\ 19.46,\ 20.20,\ 20.49)$

A feasible solution

- The three trucks can be assigned at one of the three docks, as depicted in Figure 3.
- There is no conflict between the time windows and enough time to make all pallet transfers:
  - trucks 1 and 3 are docked together for 68 minutes (20h20-19h12)
  - trucks 2 and 1 are docked together for 61 minutes (19h46-18h45)
  - trucks 3 and 2 are docked together for 34 minutes (19h46-19h12)
- Thereby, the transfer times of pallets between docks (unit of time) are:
  - 1 pallet of truck 1 (dock 1) goes to truck 3 (dock 3): 4 minutes
  - 1 pallet of truck 2 (dock 2) goes to truck 1 (dock 1): 1 minute
  - 1 pallet of truck 3 (dock 3) goes to truck 2 (dock 2): 3 minutes

  which leads to a total of $4 + 1 + 3 = 8$ units of time (minutes) used to transfer all the pallets. Furthermore, no penalty appear in this example as there is no undelivered pallet.

$$\begin{cases} \text{truck 1 assigned to dock 1} \\ \text{truck 2 assigned to dock 2} \\ \text{truck 3 assigned to dock 3} \end{cases} \longrightarrow$$

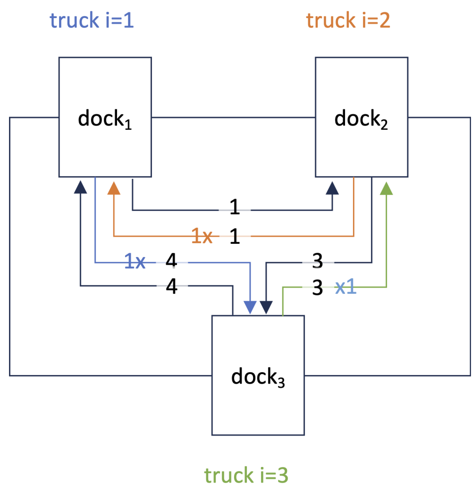


Figure 3: Assignment of trucks to docks and the corresponding transfers of pallets within the cross-dock.

3 Formulation M

The original IP model introduced by Miao et al. (2009) is stated as follow. Two sets of decision variables are defined:

$$y_{ik} = \begin{cases} 1 \text{ if truck } i \text{ is assigned to dock } k \\ \\ 0 \text{ otherwise} \end{cases} \qquad \begin{matrix} (1 \le i \le n; \\ 1 \le k \le m) \end{matrix}$$

$$z_{ijkl} = \begin{cases} 1 \text{ if truck } i \text{ is assigned to dock } k, \text{ truck } j \text{ to dock } l \\ \\ 0 \text{ otherwise} \end{cases} \qquad \begin{matrix} (1 \le i, j \le n; \\ 1 \le k, l \le m) \end{matrix}$$

The model is thus formulated as:

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{kl}\, t_{kl}\, z_{ijkl} \;+\; \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}\, f_{ij} \left(1 - \sum_{k=1}^{m}\sum_{l=1}^{m} z_{ijkl}\right) && && (1)\\
\text{s.t.} \quad & \sum_{k=1}^{m} y_{ik} \le 1 && (1 \le i \le n) && (M.2)\\
& z_{ijkl} \le y_{ik} && (1 \le i, j \le n;\ 1 \le k, l \le m) && (M.3)\\
& z_{ijkl} \le y_{jl} && (1 \le i, j \le n;\ 1 \le k, l \le m) && (M.4)\\
& y_{ik} + y_{jl} - 1 \le z_{ijkl} && (1 \le i, j \le n;\ 1 \le k, l \le m) && (M.5)\\
& \delta_{ij} + \delta_{ji} \ge z_{ijkk} && (1 \le i, j \le n, i \ne j;\ 1 \le k \le m) && (M.6)\\
& \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i \in \{i : a_i \le \tau_r\}}\sum_{j=1}^{n} f_{ij}\, z_{ijkl} - \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j \in \{j : d_j \le \tau_r\}} f_{ij}\, z_{ijkl} \le C && && (M.7)\\
& && (1 \le r \le 2n) && \\
& f_{ij}\, z_{ijkl}\, (d_j - a_i - t_{kl}) \ge 0 && (1 \le i, j \le n;\ 1 \le k, l \le m) && (M.8)\\
& y_{ik} \in \{0,1\},\ y_{jl} \in \{0,1\},\ z_{ijkl} \in \{0,1\} && (1 \le i, j \le n;\ 1 \le k, l \le m) && (M.9)
\end{aligned}
$$

where:

- (1) is the objective function, which is composed of the total operational cost (first term) and the total penalty cost (second term);
- (M.2) ensures that each truck cannot be assigned to more than one dock;
- (M.3-M.5) jointly determine the logic relationship between all y_{ik}, y_{jl} and z_{ijkl} variables;
- (M.6) ensures that one dock cannot be occupied by two trucks simultaneously;
- (M.7) is the capacity constraint, i.e. for each time point τ_r, the total number of pallets inside the warehouse cannot exceed the capacity C;
- (M.8) ensures that the transfer of pallets from truck i on dock k to truck j on dock l takes place within the time window given by the arrival time of truck i and the departure time of truck j;
- (M.9) defines the binary nature of the decision variables.

Constraints (M.2) to (M.9) define $\mathcal{X}_M$ the space of feasible solutions corresponding to the formulation M.

3.1 Discussion

3.1.1 The objective function given by expression (1)

The objective is in minimization and contains two parts: the transfer time of the pallets in the cross-dock and the number of undelivered pallets. These two parts being non-commensurable, they are converted in monetary unit with an operational cost c_{kl} associated with the transfer of pallets between the docks k and l, and a penalty cost p_{ij} associated with a transfer of the truck i to the truck j. The two parts can therefore be aggregated into the single synthesis function described by expression (1).

These costs are an artifact that makes the two parts commensurable and play a major role in the aggregation, as they can be artificially tweaked to obtain different optimal solutions. In Section 6, a bi-objective variant is proposed for the formulation M (and formulation G as expression (1) is the same in both cases). It addresses this issue by considering both parts independently in their respective units, namely time transfert and quantity of cargo transferred.

3.1.2 The notion of capacity in expression (M.7)

The constraint expressed by (M.7) in formulation M (as well as in formulation G as the constraint is the same in both cases) does not help us to relate the C value to the concrete resource limitations in the cross-dock. Indeed, this value could be based on a limited number of forklifts, on a congestion related to pallet traffic, on a limitation of pallet handling spaces near the docks, or on an another resource.

The granularity of the model does not allow us to respond, especially since the time window during which a truck is docked covers three times: unloading, loading and waiting time. No information allows to identify precisely the time when the pallets are handled within this time window, and therefore to measure accurately the required resources for handling the pallets.

This concern is particularly relevant when pallets from i must therefore be stored while awaiting the arrival of j. In order to grasp this aspect more realistically, a perspective could be to consider capacity constraints attached to platforms, as for the Cross-Docking Assignment Problem (CDAP) (Zhu et al., 2009).

3.1.3 The case of pallets not transferred.

When a pallet cannot be transferred, it remains in the truck. However, an another capacity problem may occurs here. Indeed, if this truck is initially planned to leave the cross-dock fully loaded with pallets issued from other trucks, an infeasible situation occurs in practice (truck loaded over its capacity) but is not managed by the formulation M (as well as in formulation G as the situation is identical). This case may appear with situations corresponding to the numerical instances used in papers Gelareh et al. (2016); Miao et al. (2009).

Nevertheless, if the movements of pallet exchanges between trucks correspond to a different modus operandi, this case is no longer relevant. For example, assuming that incoming and outgoing trucks are differentiated into the planning, such that incoming trucks which transport cargo from a production manufacture do not receive pallets, a pallet not transferred is returned to the supplier, which makes sense in practice. To examine this scenario, we have in perspective the project of building a new family of numerical instances.

3.1.4 The multiplication factor f_{ij} in expression (M.8)

In the constraint expressed by (M.8), f_{ij} is the expected number of pallets to be transferred between the trucks i and j, with a_i being truck i's arrival time, d_j truck j's departure time, and t_{kl} a global time to carry these pallets between the docks k and l. The aim of this constraint is to force the variable z_{ijkl} to zero when the conditions to perform this transfer are not satisfied, and (M.8) is valid in this sense.

While the interpretation of the term $f_{ij} \times t_{kl}$ is understandable (although according to the authors, the valuation of t_{kl} includes the number of pallets, and therefore, this product does not represent the time proportional to the number of pallets contrary to what it suggests), this is not the case for $f_{ij} \times d_j$ and $f_{ij} \times a_i$. The meaning behind multiplying a time marker related a truck movement with the number of pallets is not intuitive.

4 Literature criticisms about Formulation M

According to Gelareh (2021); Gelareh et al. (2015), Formulation M has several limitations and issues. They are discussed in the technical report (Gelareh et al., 2015) available online and in the paper (Gelareh, 2021). The mains criticisms and amendments reported in those documents are synthetised hereafter.

4.1 Mains criticisms and amendments

4.1.1 Infeasible bi-directional transfers

There is a bi-directional transfer between two trucks i and j if $f_{ij} > 0$ and $f_{ji} > 0$. If only one of the transfers is not feasible e.g. due to the capacity constraint, then constraints $(M.3) - (M.5)$ invalidates both of them. Indeed, if transfer from i to j is not feasible, then $z_{ijkl} = 0$ and by constraints $(M.3)$ and $(M.4)$, $y_{ik} = y_{jl} = 0$. As a result, by constraint $(M.5)$, $z_{jilk} = 0$, meaning that the opposite transfer also becomes infeasible, even though it may not be w.r.t. the practical problem. Note that a transfer may be feasible in one direction but not the other e.g. because of the global capacity constraint. As a consequence of this, feasible transfers may be canceled, resulting in potentially sub-optimal solutions for the practical problem. In such cases, less pallets may be transferred leading to higher total cost due to increased penalty costs.

4.1.2 No temporary storage

In Formulation M, a bi-direction pallet transfer between two trucks is feasible only if they are docked at a common period of time, which must be long enough for completing the transfer. As a consequence of that, solutions of practical interest may potentially be missed by the model.

Indeed, by definition of the f_{ij} coefficients in Formulation M, it is not possible to have a pair of trucks i, j such that truck i arrives after truck j leaves the cross-dock ($a_i > d_j$) and with a positive flow of pallets ($f_{ij} > 0$). However, such situations may occur in real scenarios. Thus, the authors introduce a notion of “buffer” and come with modifications on the Formulation M to allow such cases.

4.1.3 Extra transportation costs

The authors have also pointed out others unintended consequences over the optimal solutions induced by the Formulation M.

Suppose that two trucks i and j have no common time window, i.e. for example j departs before i arrives. In this case, by definition of f_{ij}, we have $f_{ij} = 0$. Furthermore, since $f_{ij} = 0$, constraint $(M.8)$ is then satisfied without inducing any restriction on z_{ijkl}, as the left-hand side of the constraint becomes 0. However, if for some k, l, we have $z_{jilk} = 1$, then by symmetry $z_{ijkl} = 1$ even though there is no transfer. Because of the first part of the objective function, i.e. $\sum_{l=1}^{m} \sum_{k=1}^{m} \sum_{i=1}^{n} \sum_{j=1}^{n} c_{kl} \, t_{kl} \, z_{ijkl}$, an extra cost is paid for a transfer that does not exist. Consequently, optimal values are inaccurate in this case. Note that the second part of the objective function is not impacted because $f_{ij} = 0$.

4.1.4 Symmetry

There is a link between z_{ijkl} and z_{jilk}. Indeed, there may be a bi-directional transfer between two trucks i and j in an optimal solution of Formulation M, i.e. $z_{ijkl} = z_{jilk}$ holds. This implies that there is an equivalent alternative optimal solution where truck j is assigned to dock l and truck i is assigned to dock k. In such a solution, $z_{ijkl} = z_{jilk}$ holds. As a result, many alternative optimal solutions exist, which may slow down the resolution process significantly.

4.1.5 Proposed modifications

First, to address the infeasible bi-directional transfers and symmetry issue, the definition of the variables z_{ijkl} has been revised so that $z_{ijkl} = 1$ if and only if there is an actual transfer from i to j, and the model is adapted accordingly.

Furthermore, constraint $(M.5)$ is removed and replaced by new constraints that allow for bi-directional transfers for trucks with non-overlapping time windows in the trucks' docking time.

Finally, constraint $(M.8)$ has been rewritten so that $z_{ijkl} = 0$ when $f_{ij} = 0$ or when the time window is too short to make a delivery. This eliminates the extra cost problem.

This led to the revised formulation named G in this paper, proposed by the authors and discussed in Section 5 and A.

4.2 Discussion

Although Gelareh et al. (2015) and Gelareh (2021) propose valid technical solutions to limitations and issues raised in Formulation M, unfortunately these two documents contain themselves inaccuracies such as several inconsistent points and typos.

First, the authors use an example to demonstrate that Formulation M does not compute all the feasible solutions. It is built so that M find no solution and forbid all transfers while Formulation G finds one.

- With Formulation M, they report a value of 16 for the objective function, and none of the variable is activated, which is different from the sum of penalties when no transfer is fulfilled, namely 22.

  Indeed, given the definition of the objective function (1) in Formulation M, when all variables $z_{ijkl} = 0$ with 11 transfers to consider, a penalty value of 1 for each unfulfilled transfer, and 2 pallets per transfer, the penalty term is:

  $\sum_{i=1}^{n} \sum_{j=1}^{n} p_{ij} \, f_{ij} \left(1 - \sum_{k=1}^{m} \sum_{l=1}^{m} z_{ijkl}\right) = 2 \times 1 \times 11 = 22$

- With Formulation G, the value 11 is reported, with 5 variable activated. However, with $c_{kl} = t_{kl} = 1$, the terms corresponding to the transfer and the penalty take respectively values 5 and 6: $1 \times 1 \times 5 \ \ + \ \ 1 \times 2 \times 6 = 17$

  Again, the value reported is different. With the information provided in the documents, to the best of our capabilities, we were not able to understand the origin of these differences.

Second, Formulation G as published by the authors does not allow to reproduce the results reported in the papers (see Section 7.4 for a comment on that matter). Indeed, the description of the formulation given in both documents contains several deficiencies (errors and missing information) to implement it properly, see Section 5 for details.

5 Formulation G

The IP model corresponding to Formulation G uses the same definition of the decision variables as Formulation M, except that the interpretation of the z_{ijkl} variables is slightly changed as described in Section 4.1.5. The formulation is reproduced exactly as described in (Gelareh, 2021; Gelareh et al., 2015, 2016) in A.1, with errors and missing information. They are listed in detail, together with the corrections we propose in A.2, and B.2 gives the code in Julia corresponding to Formulation G after having fixed issues.

While some of these inaccuracies could be easily identified and corrected with certainty, others were not immediately identifiable and are subject to interpretation. They could only be found after a meticulous analysis of the flat formulation, which necessitated testing instances by hand. We have therefore proposed corrections, which have enabled us to obtain a formulation that provides consistent solutions. Note that in particular when the indices are missing or in excess, or when their values are not stated, there is a risk of confusion and ambiguity of the expressions, exposing us to the possibility of deviating from the authors' original intentions. The amended model is thus formulated as:

$$
\begin{aligned}
\min \quad & \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{kl}\, t_{kl}\, z_{ijkl} \;+\; \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}\, f_{ij} \left(1 - \sum_{k=1}^{m}\sum_{l=1}^{m} z_{ijkl}\right) && (1)\\
\text{s.t.} \quad & \sum_{k=1}^{m} y_{ik} \le 1 \qquad (1 \le i \le n) && (G.2)\\
& z_{ijkl} \le y_{ik} \qquad (1 \le i, j \le n;\; 1 \le k, l \le m) && (G.3)\\
& z_{ijkl} \le y_{jl} \qquad (1 \le i, j \le n;\; 1 \le k, l \le m) && (G.4)\\
& y_{ik} + y_{jk} \le 1 + \delta_{ij} + \delta_{ji} \qquad (1 \le i, j \le n, i \neq j;\; 1 \le k \le m) && (G.5)\\
& z_{ijkk} \le \delta_{ij} \qquad (1 \le i, j \le n, i \neq j;\; 1 \le k \le m) && (G.6)\\
& \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i \in \{i: a_i \le \tau_r\}}\sum_{j=1}^{n} f_{ij}\, z_{ijkl} - \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j \in \{j: d_j \le \tau_r\}} f_{ij}\, z_{ijkl} \le C \qquad (1 \le r \le 2n) && (G.7)\\
& z_{ijkl} = 0 \qquad (1 \le i, j \le n, i \neq j;\; 1 \le k, l \le m;\; d_j - a_i - f_{ij}\, t_{kl} \le 0) && (G.8)\\
& y_{ik} \in \{0,1\},\; y_{jl} \in \{0,1\},\; z_{ijkl} \in \{0,1\} (1 \le i, j \le n;\; 1 \le k, l \le m) && (G.9)
\end{aligned}
$$

where:

(G.5) guarantee that if the arrival/departure time windows of two trucks i and j overlap ($\delta_{ij} = \delta_{ji} = 0$), at most one of them can be docked at dock k (not both);

(G.6) ensure that truck i and truck j can use the same dock for realizing the transfers of pallets from i to j, only if their time windows do not overlap;

(G.8) ensure that z_{ijkl} is set to zero if $f_{ij} > 0$ and $(d_j - a_i - t_{kl} < 0)$.

The others parts of the formulation, i.e. (1), $(G.2 - G.4)$, $(G.7)$, and $(G.9)$, share the same definition with Formulation M. Constraints $(G.2)$ to $(G.9)$ define $\mathcal{X}_G$ the space of feasible solutions corresponding to the formulation G.

6 Formulations 2M and 2G, and lexicographic optimal solution

The bi-objective formulations are now presented. They are straitghforward derived from the single objective formulations where expression (1), which corresponds to the objective function in M and G, is spitted in two parts, resulting in the two objectives functions without the costs c_{kl} and p_{ij}: the expression (1.1) minimizes the transfer time of the pallets in the cross-dock, and the expression (1.2) maximizes the quantity of cargos transferred. The IP models are formulated as follow:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{m} t_{kl}\, z_{ijkl} \tag{1.1}$$

$$\max \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{m} f_{ij}\, z_{ijkl} \tag{1.2}$$

subject to constraints defining the space of feasible solutions X_M and X_G, respectively for formulation 2M and 2G.

For these two formulations, a lexicographically optimal solution (see Ehrgott (2005) for more details) will be desired and sufficient. It corresponds to the optimization problem where the main objective to be maximized is the quantity of cargos transferred (objective 1.2), and the secondary objective to be minimized is the total transfer time of the pallets in the cross-dock (objective 1.1). Such an optimal solution makes sense from a practical point of view, as the main goal of a practitioner is to transfer the planned cargo through the cross-dock. Then, once this goal is fulfilled, time or resources necessary to transfer this maximum number of pallets within the cross-dock can be minimized.

A lexicographically optimal solution is obtained at the end of two successive single-objective optimizations. The main objective is firstly optimized giving (if it exists) an optimal value. Next, an additional equality constraint on the optimal value obtained for the first objective function is added, and the secondary objective is optimized giving (if it exists) an optimal value. The couple of optimal values obtained gives the value of the lexicographic optimal solution.

7 Numerical experiments

7.1 Numerical Instances

The experiments reported by Miao et al. (2009) have been conducted with a collection of instances that represent a cross-dock with a I-shape, where dock gates are symmetrically located on each sides. Datasets have been randomly generated (i) with different values of parameters n and m, and (ii) with the following characteristics:

- the time window of a truck on a dock is randomly comprised between 45 and 74 minutes;
- the value f_{ij} giving the number of pallets to be shipped is randomly chosen between 6 and 60.
- the duration of the transfer t_{kl} is based on the Manhattan distance between two docks gates. The authors specifies that a “proportional conversion" is necessary to define t_{kl} correctly but without more detail.

Note that any truck can carry out a pallet transfer to all the other trucks (see Figure 4). It is in this sense that we named them "full-mesh".

On base of their size $n \times m$, the authors distinguish three categories of instances, (1) the *small size instances* (10×3 to 18×6), (2) the *medium size instances* (20×6 to 40×8), and (3) the *large size instances* (50×10 to 80×12). Nevertheless, data files are not provided by the authors.

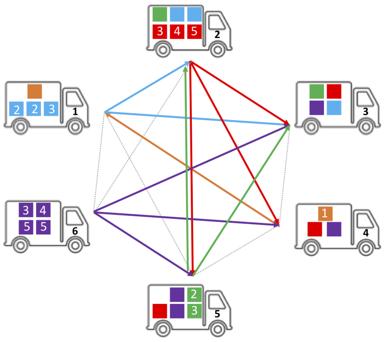


Figure 4: Example of a full-mesh transfer of pallets. Trucks arrives with pallets from suppliers, and leave with pallets to customers. For example, truck numbered 2 arrives with three (red) pallets destined to trucks 3/4/5, and leaves with pallets provided by truck 1 (blue) and truck 5 (green).

Gelareh et al. (2015) have generated instances following generation procedure described below. For the t_{kl}, no proportional conversion is considered, only the Manhattan distance is kept. The data files generated are available online[2] but raise questions about at least 2 aspects.

First, this way of defining transfers from truck k to truck l without "proportional conversion" implies that there is no link between the duration of a pallet transfer t_{kl} and the quantity of pallet in this transfer f_{ij}. These choices can lead to a long transfer time for a small number of pallets to be transferred, which is not in line with a real situation. Second, the units associated to values such as t_{kl} are not stated. The units applied to the data is a potential source of inconsistencies that can lead to misuse of the data, in particular with respect to the truck's time windows.

7.2 Experimental environment

The program is implemented in Julia (version 1.10) and uses JuMP (version 1.21) as algebraic modeling language. All the codes are available online at `https://github.com/xgandibleux/TDAP`, including an implementation of Formulation M, Formulation G corrected as described in this paper, Formulations 2M and 2G. The program is ready to be run with Gurobi or GLPK as MIP solver[3].

The experiment are performed on a MacBook Pro laptop under macOS Ventura (version 13.6) equipped as follow:

- CPU model: 3.5 GHz Intel Core i7 double cores.
- Memory: 16 Go 2133 MHz LPDDR3.

Here, Gurobi Optimizer (version 10.0.3 build v10.0.3rc0 (mac64[x86])) is used.

7.3 Experiment 1: detailed numerical analysis of Formulations M and G on a didactic instance

This experiment examines in detail the optimal solutions obtained by Formulations M and G on a didactic instance composed of 5 trucks and 3 docks. The maximum capacity of the terminal is 813

[2] `https://www.lgi2a.univ-artois.fr/~gelareh/downloads/cross_dock/data.rar`

[3] The program provided online is configured to run with GLPK. Explanations are provided into the `readme` file to switch to Gurobi.

pallets. The previsional arrival and departure time of trucks into the format (hh.mm), transformed in minutes in the interval [00:00;23:59], is given by Table 1a.

The previsional number of pallets f_{ij} to transfer from truck i to truck j, the penalties p_{ij} (in unit of cost) when a transfer from i to j is not achieved, the operational times (minutes) t_{kl} from k to l, and the transportation cost (in unit of cost) c_{kl} from k to l are respectively given in Table 1b, 1c, 1d, and 1e. The Gantt chart depicted in Figure 1f illustrates the previsional planning of trucks and transfers of pallets.

The information deduced from the data are (1) the simultaneous presence indicators δ, and (2) the time markers τ. They are reported in Tables 2a and 2b respectively.

An optimal solution obtained respectively with Formulation M and Formulation G is given in Table 3. Table 3a shows the optimal assignments of trucks to docks and Table 3b reports the transfers of pallets.

The value of the objective function for an optimal solution obtained with Formulation M is equal to 12, which is also the value of the operational cost. Indeed, all the transfers awaited (7 transfers) are fulfilled (ratio of 100.0%), thus the penalty cost is equal to 0.

With Formulation G, the value of the objective function for an optimal solution obtained is equal to 67, with an operational cost of 3 and a penalty cost of 64. Indeed, only 6 transfers on the 7 awaited are fulfilled (ratio of 85.71%).

For the two optimal solutions collected, Figures 3c and 3d illustrates the evolution of the load in the terminal in term of pallets respectively for Formulation M and G. This numerical result obtained with a didactic instance shows that an optimal solution obtained with Formulation G may be dominated by an optimal solution obtained with M.

truck ID (i)	arrival (a_i)		departure (d_i)	
	hh.mm	minutes	hh.mm	minutes
1	17:26	1046	18:17	1097
2	17:14	1034	18:17	1097
3	19:15	1155	20:20	1220
4	18:30	1110	19:16	1156
5	19:47	1187	20:49	1249

(a) Previsional planning of trucks.

f_{ij}		j				
		1	2	3	4	5
	1	.	.	.	.	33
	2	.	.	36	.	.
i	3	.	.	.	8	50
	4	.	.	8	.	52
	5	.	.	24	.	.

(b) Previsional number of pallets f_{ij} to transfer from truck i to truck j.

p_{ij}		j				
		1	2	3	4	5
	1	.	.	.	.	8
	2	.	.	8	.	.
i	3	.	.	.	8	8
	4	.	.	9	.	8
	5	.	.	8	.	.

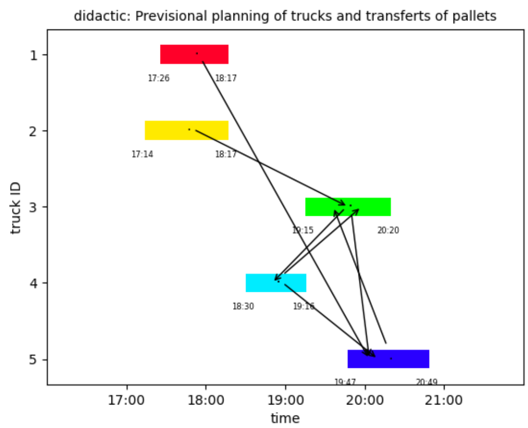
(c) Penalties p_{ij} from i to j.

t_{kl}		l		
		1	2	3
	1	0	1	4
k	2	1	0	3
	3	4	3	0

(d) Operational times (minutes) t_{kl} from k to l.

c_{kl}		l		
		1	2	3
	1	0	1	1
k	2	1	0	2
	3	1	2	0

(e) Transportation cost c_{kl} from k to l.



(f) Overview of the previsionnal planning of trucks and transfers of pallets.

Table 1: Data provided for the didactic instance.

δ_{ij}		j				
		1	2	3	4	5
		0	0	1	1	1
		0	0	1	1	1
	i	1	1	0	0	0
		1	1	0	0	1
		1	1	0	1	0

(a) δ_{ij}, the simultaneous presence indicator between trucks i and j (1 if no overlap, 0 otherwise).

id (r)	time marker (τ_r)		arrivals (`atr[r]`)	departures (`dtr[r]`)
	hh.mm	minutes	list of trucks i	list of trucks j
1	17:14	1034	[2]	[ ]
2	17:26	1046	[1, 2]	[ ]
3	18:17	1097	[1, 2]	[1, 2]
4	18:17	1097	[1, 2]	[1, 2]
5	18:30	1110	[1, 2, 4]	[1, 2]
6	19:15	1155	[1, 2, 3, 4]	[1, 2]
7	19:16	1156	[1, 2, 3, 4]	[1, 2, 4]
8	19:47	1187	[1, 2, 3, 4, 5]	[1, 2, 4]
9	20:20	1220	[1, 2, 3, 4, 5]	[1, 2, 3, 4]
10	20:49	1249	[1, 2, 3, 4, 5]	[1, 2, 3, 4, 5]

(b) Arrivals and departures of trucks at a time marker. For example, at the 2nd ($r = 2$) time marker ($\tau_2 = 17{:}26$), trucks 1 and 2 are arrived (`atr[2]=[1,2]`), none departure of truck (`dtr[2]=[]`).

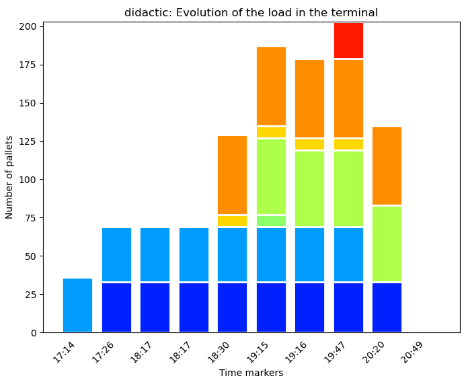
Table 2: Information deduced from the data for the didactic instance.

Formulation M				Formulation G			
truck	dock	arrival	departure	truck	dock	arrival	departure
1	1	17:26	18:17	1	2	17:26	18:17
2	2	17:14	18:17	2	1	17:14	18:17
3	1	19:15	20:20	3	1	19:15	20:20
4	2	18:30	19:16	4	2	18:30	19:16
5	2	19:47	20:49	5	2	19:47	20:49

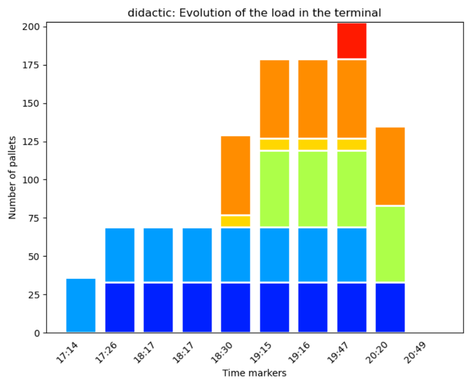
(a) Assignments of trucks to docks.

Formulation M			Formulation G			
trucks	docks	# pallets	trucks		docks	# pallets
$i \to j$	$k \to l$		i	j	$k \to l$	
$1 \to 5$	$1 \to 2$	33	1	5	$2 \to 2$	33
$2 \to 3$	$2 \to 1$	36	2	3	$1 \to 1$	36
$3 \to 4$	$1 \to 2$	8	3	5	$1 \to 2$	50
$3 \to 5$	$1 \to 2$	50	4	3	$2 \to 1$	8
$4 \to 3$	$2 \to 1$	8	4	5	$2 \to 2$	52
$4 \to 5$	$2 \to 2$	52	5	3	$2 \to 1$	24
$5 \to 3$	$2 \to 1$	24				

(b) Transfers of pallets.



(c) Evolution of the load in the terminal in term of pallets for Formulation M.



(d) Evolution of the load in the terminal in term of pallets for Formulation G.

Table 3: Optimal solution collected with each formulation.

7.4 Experiment 2: comparison between Formulations M and G

This experiment focuses on the optimal solutions obtained by Formulations M and G on small and medium size instances. For this experiment, a computation time limit of 600 seconds is given to Gurobi. This value was chosen because it has shown to be sufficient to observe the behaviour of the two formulations given our computer configuration.

In total, 65 instances of increasing size were selected from the available data set. Most instances have been solved to optimality by Gurobi within the time limit for at least one of the formulations. For both formulations and for each instance,(1) the optimal objective value (aggregated cost, operational cost, penalty cost), (2) the CPU time in seconds, (3) the transfers of pallets between docks (number and percentage) achieved, (4) the number of trucks assigned to a dock, and (5) the total time transfert as well as the total quantity of cargo transferred, are gathered together in Tables 5 and 6 in C, and Figures 5 and 6 synthesize graphically these results.

As announced in Section 4, we were unable to reproduce the optimal value reported by the authors for the given instances. For example, we obtained an optimal value of 7911, 4032 and 8556 for instances `data_12_4_1`, `data_12_4_2` and `data_12_4_3` respectively. On the other hand, Gelareh et al. (2016) reported optimal values of 7747, 4032 and 8562. We observe that our values are sometimes equal, greater, or lower than those of Gelareh et al. (2016). To the best of abilities, we could not identify any clear pattern to predict which scenario will occur for which instance. Note that we checked the feasibility of all solutions we obtained, and the code used to do that is available with the source code. Unfortunately, the authors do not provide the optimal solutions they obtained nor do they mention any open access to their codes. Therefore, it was not possible to determine the origin of these differences between both numerical results.

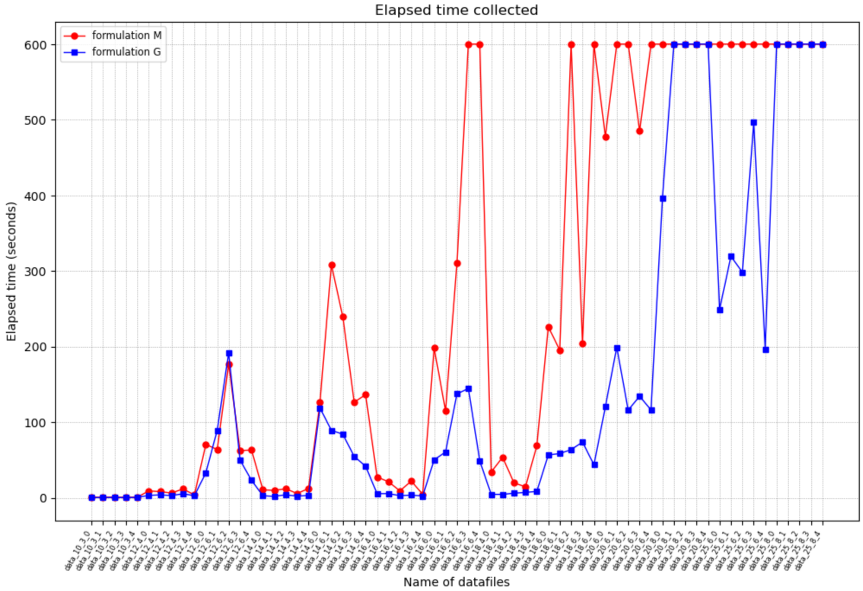
Figure 5a shows that except for two small instances, Formulation M requires more CPU time to solve than Formulation G, and this gap seems to widen with the size of the instance. Moreover, it is important to mention that we did not make use of the valid inequalities presented in Gelareh et al. (2015, 2016). As a result, the CPU time for Formulation G could be further reduced. Consequently, Formulation G appears to be the best one with respect to computation time.

Figure 5b compares the optimal value of the objective function between both formulations. We observe that the optimal value for Formulation G is lower than the one for Formulation M in all but six instances. Thus, empirically, the adjustments proposed by Gelareh et al. (2015) have a positive impact in general, although not systematic.

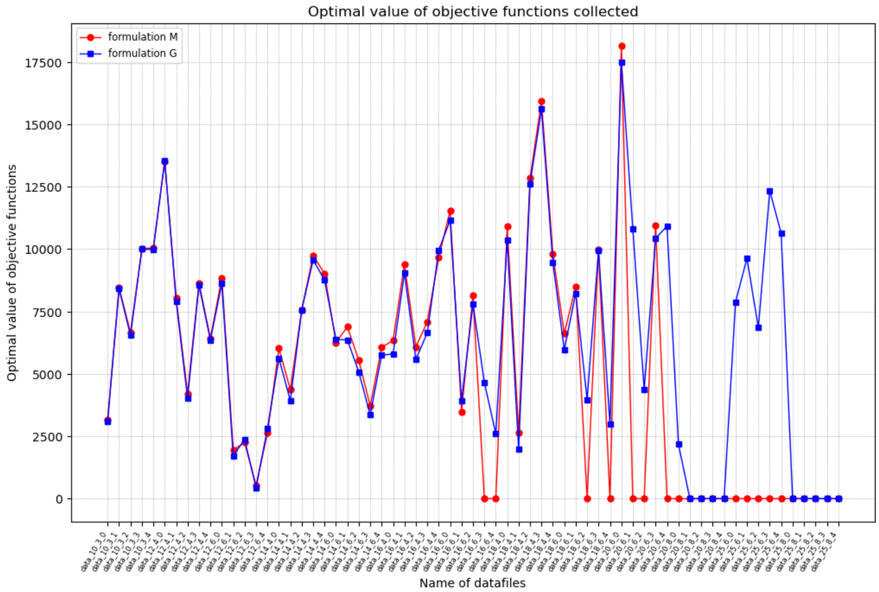
When taking a closer look at Tables 5 and 6, we see that the optimal solutions found by Formulation M often have a larger operational cost. However, the penalty costs are in general equal or lower than those of Formulation G. From that, we conclude that Formulation M finds solutions with a greater number of transfers fulfilled (as illustrated in Figures 6a and 6b), which is surprising considering that Formulation G should be able to find more feasible transfers.

From a practitioner's perspective, this nuance may be relevant as it appears reasonable to put priority first on the number of pallets transferred. Indeed, it may be of greater importance to utilize the cross-dock as much as possible in order to reduce the number of unsatisfied clients due to undelivered pallets. Only then, the total transfer time within the cross-dock are optimized in order to minimize the resources used to perform the deliveries.

Based on these observations, we built the two bi-objective formulation presented in Section 6, which return lexicographically optimal solutions for this problem.

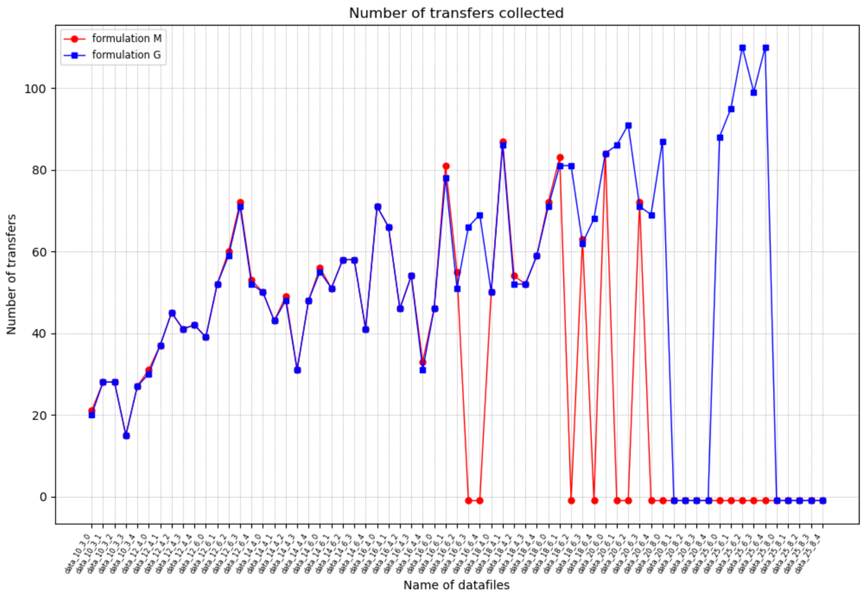


(a) Comparison elapsed time collected for each instance. When a y value is equal to 600, the corresponding optimal solution is not available.

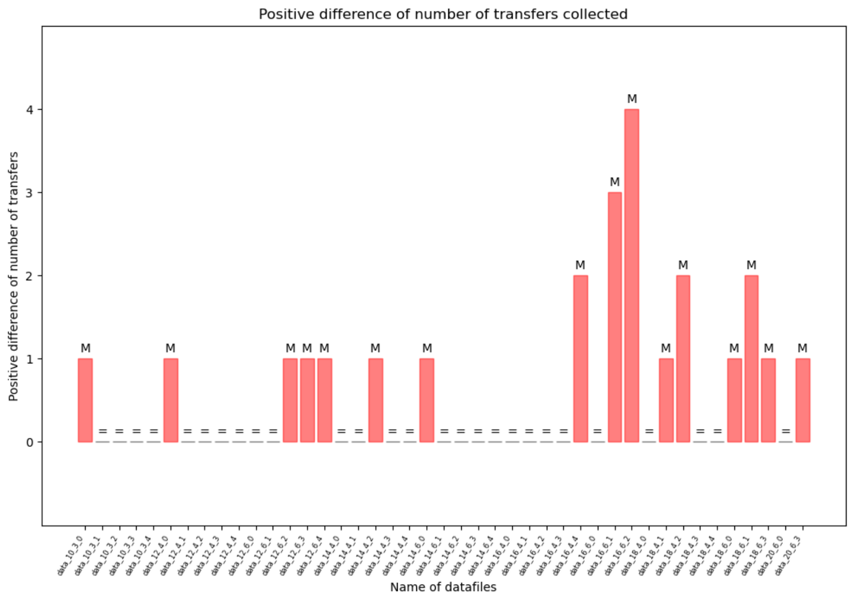


(b) Comparison of objective values collected for each instance. When a y value is equal to 0, the corresponding optimal solution is not available.

Figure 5: Comparison between Formulations M and G on the objective value and the elapsed time.



(a) Comparison on number of transfers collected for each instance. When a y value is equal to 0, the corresponding optimal solution is not available.



(b) Positive difference on number of transfers collected for each instance solved to optimality for both formulations. "M" means the difference is in favor of Formulation M, "G" means the difference is in favor of Formulation G, "=" means there is no difference between the two formulations.

Figure 6: Comparison between Formulations M and G on the number of transfers.

7.5 Experiment 3: results obtained with formulations 2M and 2G

Here, instances are solved using Formulations 2M and 2G. The lexicographic optimization is done in two stages. First, objective (1.2) is maximized, and an optimal value v^* is observed. Second, an additional constraint in the form $\sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{m} f_{ij}\, z_{ijkl} = v^*$ is added to the model, i.e. we force objective (1.2) to be optimal. Then, objective (1.1) is minimized with the new constraint added.

The time limit is kept to 600 seconds for the whole lexicographic optimization process. Detailed results are reported in Table 7 and 8 in C.

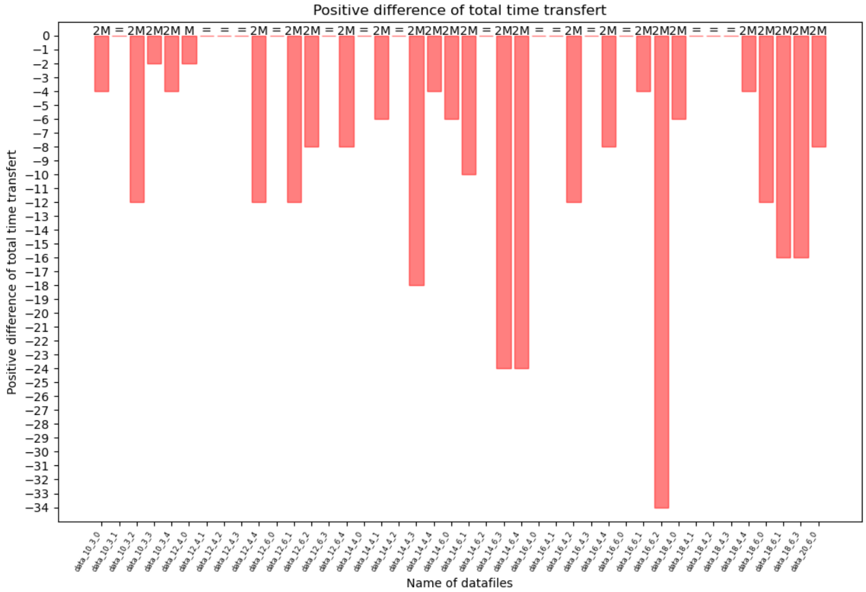
instance	bi-objective		single objective	
	formulation	value	formulation	value
data_12_4_0	2G	(63 ; 980)	G	(64 ; 973)
data_16_4_4	2M	(186 ; 1175)	M	(194 ; 1167)
data_16_6_0	2G	(97 ; 1591)	G	(110 ; 1590)
data_16_6_2	2M	(310 ; 1767)	M	(344 ; 1764)
data_18_4_0	2M	(386 ; 1818)	M	(392 ; 1811)
data_18_4_4	2M	(386 ; 1980)	M	(390 ; 1975)
data_20_6_1	2G	(195 ; 2902)	G	(199 ; 2896)
data_20_6_3	2G	(141 ; 2351)	G	(152 ; 2348)
data_25_6_4	2G	(262 ; 3552)	G	(271 ; 3531)

Table 4: Cases identified where the value of a lexicographic optimal solution dominates strictly the value of a single objective optimal solution. The couple of values between parentheses reported in a column "value" corresponds on the left at the time transfer (to minimize), and on the right the quantity of cargo transferred (to maximize).

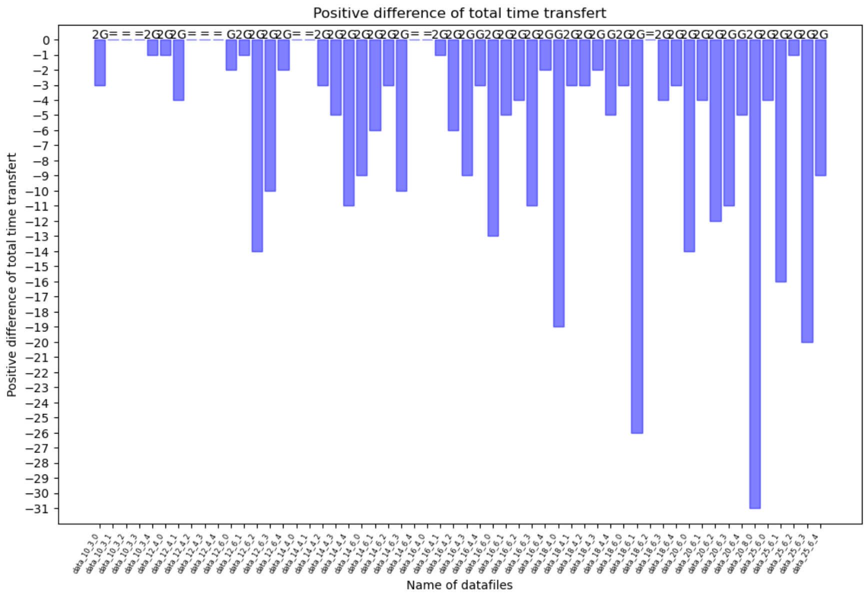
First, solutions obtained with the bi-objective formulations weakly dominate the solutions obtained with the single-objective formulations, i.e. they are equal on one objective and strictly better on the other one. Indeed, the secondary objective (objective (1.1), which minimizes the total transfer time) is improved 26 times for 2M and 45 times for 2G.

Furthermore, Table 4 show that for 9 instances, the lexicographic solutions strictly dominate those from the single-objective formulation, i.e. they are strictly better on all objectives. This indicates that a multi-objective approach where all objectives are considered independently may provide solutions of greater quality for the practical problem, as opposed to a single-objective version where all objectives are aggregated into one complex function.

The CPU time for the lexicographic approach is in general higher than for the single-objective approach, since two IPs have to be solved each time. However, this increase seems to stay within a reasonable proportion, and we even observed the opposite on a few instances. For example, `data_20_6_2` was not solved within 600 seconds for M, but an optimal solution was found in 363.176 seconds for 2M. Similarly, we see that `data_14_6_0` was solved in 116.325 seconds for G, and 102.004 seconds for 2G. This may be explained by the fact that the expression of the objective functions are much simpler in the bi-objective models, as opposed to the complex aggregation function in the single-objective formulations. The practical value of the multi-objective approach for the TDAP is therefore not overshadowed by the computation time required to find a lexicographically optimal solution.



(a) Cases where the value of the time transfert obtained with 2M is better than the value obtained with M.



(b) Cases where the value of the time transfert obtained with 2G is better than the value obtained with G.

Figure 7: Comparisons between values on time transfer collected with M vs 2M, and G vs 2G.

8 Conclusion and perspectives

This paper studies two formulations for the TDAP published since 2009, namely Formulations M and G. Our literature review unveiled several critics and discussions in various papers published since then. Even though the arguments presented are justified, strong, and have a valid technical background, we identified a few errors or inconsistencies in some of these studies.

On this basis, we first aimed at providing a rigorous description of the two formulations. For this purpose, both were written with unified notations, corrected if necessary, discussed, and analyzed numerically. The conclusions drawn led us to propose a bi-objective variant of these models, resulting in four formulations in total.

Finally, we conducted numerical experiments on instances from the literature. All solutions reported are optimal, and all implementations are available online. First, we observed that there was no clear winner between Formulations M and G. Indeed, although Formulation G has a better optimal value in general, Formulation M is still better for some instances. Furthermore, Formulation M tends to generate solutions with more pallets delivered, which may be interesting from a practical point of view, but is also slower to solve.

Experiment 3 showed the practical value of a multi-objective approach in terms of quality of the solution. Indeed, the lexicographic solutions weakly or strictly dominate the solutions from Formulations M and G in terms of number of transfers achieved and total time transfer in the cross-dock. Moreover, although the computation time is slightly higher in general for 2M and 2G, there is again no clear winner as some instances were solved faster with the lexicographic approach.

Various perspective for more realistic models were mentioned in the discussions, such as considering a more detailed management of the capacity constraint, having a more realistic calculation of the transfer time that depends also on the number of pallets to transport, studying of the formulations on different classes of instances, ect. Naturally, the valid inequalities known for Formulation G could also be applied to Formulation 2G and to some extent M and 2M to reduce the computation time. Finally, while already relevant, the multi-objective approach is considered in its simplest form at the moment. Multiple non-dominated points could be computed to provide the practitioner with a better picture of the trade-off between the objectives considered, depending on what makes the most sense for the practical problem at hand.

A Formulation G from Gelareh (2021); Gelareh et al. (2015, 2016)

A.1 The original formulation

The IP model is reproduced exactly as presented in papers, with errors and missing information. It is formulated as:

$$\min \quad \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{kl}\, t_{kl}\, z_{ijk} \;+\; \sum_{i=1}^{n}\left(\sum_{j=1}^{n} p_{ij}\, f_{ij}\left(1-\sum_{k=1}^{m}\sum_{l=1}^{m} z_{ijkl}\right)\right) \tag{1}$$

$$\text{s.t.} \quad \sum_{k=1}^{m} y_{ik} \le 1 \qquad \forall i \tag{2}$$

$$z_{ijkl} \le y_{ik} \qquad \forall i,j,k,l \tag{3}$$

$$z_{ijkl} \le y_{jl} \qquad \forall i,j,k,l \tag{4}$$

$$y_{ik} + y_{jk} \le 1 + \delta_{ij} + \delta_{ji} \qquad \forall i,j,k \tag{5}$$

$$z_{ijkk} \le \delta_{ij} \qquad \forall i, j \ne i, k, l \tag{6}$$

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i\in\{i:a_i\le\tau_r\}}\sum_{j=1}^{n} f_{ij}\, z_{ijkl} \;+\; \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i}^{n}\sum_{j\in\{j:d_j\le\tau_r\}} f_{ij}\, z_{ijkl} \le C \qquad \forall r \in \{1,2,\ldots,2n\} \tag{7}$$

$$z_{ijkl} = 0 \qquad \forall i,j,k,l : j \ne i, (d_j - a_i - t_{kl} \le 0) \tag{8}$$

$$z_{ijkl} \le 1 \qquad \forall i,j,k,l : j \ne i \tag{9}$$

$$z_{ijkl} \ge 0 \qquad \forall i,j,k,l : j \ne i \tag{10}$$

$$y_{ik} \le 1 \qquad \forall i,k \tag{11}$$

$$y_{ik} \ge 0 \qquad \forall i,k \tag{12}$$

$$y_{ik} \in \{0,1\},\; z_{ijkl} \in \{0,1\} \tag{13}$$

A.2 The corrections proposed

A number of inaccuracies and incompleteness were identified in Formulation G, making it non-operational. They are reported hereafter (also coloured in red in the text).

- in (1): index missing into the first term:

  $$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j=1}^{n} c_{kl}\, t_{kl}\, z_{ijkl} + \sum_{i=1}^{n}\left(\sum_{j=1}^{n} p_{ij}\, f_{ij}\left(1-\sum_{k=1}^{m}\sum_{l=1}^{m} z_{ijkl}\right)\right)$$

- in (5): wrong definition of index:

  $$y_{ik} + y_{jk} \leq 1 + \delta_{ij} + \delta_{ji} \qquad \forall i, j \neq i, k$$

- in (6): wrong definition of index:

  $$z_{ijkk} \leq \delta_{ij} \qquad \forall i, j \neq i, k, l$$

- in (7): wrong operation, and initial value missing:

  $$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i\in\{i:a_i\leq\tau_r\}}\sum_{j=1}^{n} f_{ij}\, z_{ijkl} - \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j\in\{j:d_j\leq\tau_r\}} f_{ij}\, z_{ijkl} \leq C \qquad \forall r \in \{1, 2, ..., 2n\}$$

- in (9-12): constraints (9-12) redundant with (13), and domains of indexes not stated in (13):

  $y_{ik} \in \{0, 1\},\ z_{ijkl} \in \{0, 1\}$

B Formulations implemented in Julia with JuMP

B.1 Formulation M

```
using JuMP
mod = Model()
# --- Variables ---------------------------------------------------------------
# 1 if truck i is assigned to dock k, 0 otherwise:
@variable( mod,
y[1:n, 1:m], Bin
)
# 1 if truck i is assigned to dock k and truck j to dock l, 0 otherwise:
@variable( mod,
z[1:n, 1:n, 1:m, 1:m], Bin
)
# --- Objective ---------------------------------------------------------------
# total operational cost + total penalty cost:
@objective( mod,
Min,
sum(c[k,l] * t[k,l] * z[i,j,k,l] for i=1:n, j=1:n, k=1:m, l=1:m)
+
sum( p[i,j] * f[i,j] * ( 1 - sum( z[i,j,k,l] for k=1:m, l=1:m) ) for i=1:n, j=1:n) )
)
# --- (M2-M8): constraints ----------------------------------------------------
@constraint( mod,
cstM2_[i=1:n],
sum(y[i,k] for k=1:m) <= 1
)
@constraint( mod,
cstM3_[i=1:n, j=1:n, k=1:m, l=1:m],
z[i,j,k,l] <= y[i,k]
)
@constraint( mod,
cstM4_[i=1:n, j=1:n, k=1:m, l=1:m],
z[i,j,k,l] <= y[j,l]
)
@constraint( mod,
cstM5_[i=1:n, j=1:n, k=1:m, l=1:m],
y[i,k] + y[j,l] - 1 <= z[i,j,k,l]
)
@constraint( mod,
cstM6_[i=1:n, j=1:n, k=1:m; i!=j],
x[i,j] + x[j,i] >= z[i,j,k,k]
)
@constraint( mod,
cstM7_[r=1:2*n],
sum(f[i,j] * z[i,j,k,l] for i in atr[r], j=1:n, k=1:m, l=1:m)
-
sum(f[i,j] * z[i,j,k,l] for i=1:n, j in dtr[r], k=1:m, l=1:m) <= C
)
@constraint( mod,
cstM8_[i=1:n, j=1:n, k=1:m, l=1:m],
f[i,j] * z[i,j,k,l] * (d[j] - a[i] - t[k,l]) >= 0
)
```

B.2 Formulation G after having fixed issues

```
using JuMP
mod = Model()
# --- Variables ---------------------------------------------------------------
# 1 if truck i is assigned to dock k, 0 otherwise:
@variable( mod,
y[1:n, 1:m], Bin
)
# 1 if truck i is assigned to dock k and truck j to dock l, 0 otherwise:
@variable( mod,
z[1:n, 1:n, 1:m, 1:m], Bin
)
# --- Objective ---------------------------------------------------------------
# total operational cost + total penalty cost
@objective( mod,
Min,
sum(c[k,l] * t[k,l] * z[i,j,k,l] for i=1:n, j=1:n, k=1:m, l=1:m)
+
sum( p[i,j] * f[i,j] * ( 1 - sum( z[i,j,k,l] for k=1:m, l=1:m) ) for i=1:n, j=1:n) )
)
# --- (G2-G8): constraints ----------------------------------------------------
@constraint( mod,
cstG2_[i=1:n],
sum(y[i,k] for k=1:m) <= 1
)
@constraint( mod,
cstG3_[i=1:n, j=1:n, k=1:m, l=1:m],
z[i,j,k,l] <= y[i,k]
)
@constraint( mod,
cstG4_[i=1:n, j=1:n, k=1:m, l=1:m],
z[i,j,k,l] <= y[j,l]
)
@constraint( mod,
cstG5_[i=1:n, j=1:n, k=1:m; i!=j],
y[i,k] + y[j,k] <= 1 + x[i,j] + x[j,i]
)
@constraint( mod,
cstG6_[i=1:n, j=1:n, k=1:m; i!=j],
z[i,j,k,k] <= x[i,j]
)
@constraint( mod,
cstG7_[r=1:2*n],
sum(f[i,j] * z[i,j,k,l] for i in atr[r], j=1:n, k=1:m, l=1:m)
-
sum(f[i,j] * z[i,j,k,l] for i=1:n, j in dtr[r], k=1:m, l=1:m) <= C
)
@constraint( mod,
cstG8_[i=1:n, j=1:n, k=1:m, l=1:m; i!=j && (d[j]-a[i]-t[k,l])<=0],
z[i,j,k,l] == 0
)
```

B.3 Formulation 2M

```
using JuMP
mod = Model()
# --- Variables ---------------------------------------------------------------
# 1 if truck i is assigned to dock k, 0 otherwise:
@variable( mod,
y[1:n, 1:m], Bin
)
# 1 if truck i is assigned to dock k and truck j to dock l, 0 otherwise:
@variable( mod,
z[1:n, 1:n, 1:m, 1:m], Bin
)
# --- Objectives --------------------------------------------------------------
# (1.1): total transfert time of the pallets in the cross-dock
@expression( mod, objFct1_transfertTime,
sum(t[k,l] * z[i,j,k,l] for i=1:n, j=1:n, k=1:m, l=1:m)
)
# (1.2): total quantity transferred
@expression( mod, objFct2_quantityTransferred,
sum(f[i,j] * z[i,j,k,l] for i=1:n, j=1:n, k=1:m, l=1:m)
)
if obj == :obj1
@objective(mod, Min, objFct1_transfertTime)
else
@objective(mod, Max, objFct2_quantityTransferred)
end
# --- (M2-M8): constraints ----------------------------------------------------
@constraint( mod,
cstM2_[i=1:n],
sum(y[i,k] for k=1:m) <= 1
)
@constraint( mod,
cstM3_[i=1:n, j=1:n, k=1:m, l=1:m],
z[i,j,k,l] <= y[i,k]
)
@constraint( mod,
cstM4_[i=1:n, j=1:n, k=1:m, l=1:m],
z[i,j,k,l] <= y[j,l]
)
@constraint( mod,
cstM5_[i=1:n, j=1:n, k=1:m, l=1:m],
y[i,k] + y[j,l] - 1 <= z[i,j,k,l]
)
@constraint( mod,
cstM6_[i=1:n, j=1:n, k=1:m; i!=j],
x[i,j] + x[j,i] >= z[i,j,k,k]
)
@constraint( mod,
cstM7_[r=1:2*n],
sum(f[i,j] * z[i,j,k,l] for i in atr[r], j=1:n, k=1:m, l=1:m)
-
sum(f[i,j] * z[i,j,k,l] for i=1:n, j in dtr[r], k=1:m, l=1:m) <= C
)
@constraint( mod,
cstM8_[i=1:n, j=1:n, k=1:m, l=1:m],
f[i,j] * z[i,j,k,l] * (d[j] - a[i] - t[k,l]) >= 0
)
```

B.4 Formulation 2G

```
using JuMP
mod = Model()
# --- Variables ---------------------------------------------------------------
# 1 if truck i is assigned to dock k, 0 otherwise:
@variable( mod,
y[1:n, 1:m], Bin
)
# 1 if truck i is assigned to dock k and truck j to dock l, 0 otherwise:
@variable( mod,
z[1:n, 1:n, 1:m, 1:m], Bin
)
# --- Objectives --------------------------------------------------------------
# (1.1): total transfert time of the pallets in the cross-dock
@expression( mod, objFct1_transfertTime,
sum(t[k,l] * z[i,j,k,l] for i=1:n, j=1:n, k=1:m, l=1:m)
)
# (1.2): total quantity transferred
@expression( mod, objFct2_quantityTransferred,
sum(f[i,j] * z[i,j,k,l] for i=1:n, j=1:n, k=1:m, l=1:m)
)
if obj == :obj1
@objective(mod, Min, objFct1_transfertTime)
else
@objective(mod, Max, objFct2_quantityTransferred)
end
# --- (G2-G8): constraints ----------------------------------------------------
@constraint( mod,
cstG2_[i=1:n],
sum(y[i,k] for k=1:m) <= 1
)
@constraint( mod,
cstG3_[i=1:n, j=1:n, k=1:m, l=1:m],
z[i,j,k,l] <= y[i,k]
)
@constraint( mod,
cstG4_[i=1:n, j=1:n, k=1:m, l=1:m],
z[i,j,k,l] <= y[j,l]
)
@constraint( mod,
cstG5_[i=1:n, j=1:n, k=1:m; i!=j],
y[i,k] + y[j,k] <= 1 + x[i,j] + x[j,i]
)
@constraint( mod,
cstG6_[i=1:n, j=1:n, k=1:m; i!=j],
z[i,j,k,k] <= x[i,j]
)
@constraint( mod,
cstG7_[r=1:2*n],
sum(f[i,j] * z[i,j,k,l] for i in atr[r], j=1:n, k=1:m, l=1:m)
-
sum(f[i,j] * z[i,j,k,l] for i=1:n, j in dtr[r], k=1:m, l=1:m) <= C
)
@constraint( mod,
cstG8_[i=1:n, j=1:n, k=1:m, l=1:m; i!=j && (d[j]-a[i]-t[k,l])<=0],
z[i,j,k,l] == 0
)
```

C Detailed results

C.1 Results for formulations M

fname	tElapsed	z	zCost	zPenalty	totTimeTrans	totQuanTrans	nTruckAssig	nTransDone	pTransDone
	sec						#	#	%
data_10_3_0	0.509	3163	160	3003	124	789	9	21	67.74
data_10_3_1	0.562	8454	156	8298	84	1030	7	28	46.67
data_10_3_2	0.674	6659	234	6425	114	949	8	28	58.33
data_10_3_3	0.413	10021	60	9961	48	498	6	15	31.25
data_10_3_4	0.727	10047	192	9855	84	968	7	27	45.76
data_12_4_0	8.811	13508	196	13312	126	988	8	31	41.89
data_12_4_1	7.397	8035	220	7815	148	1285	9	37	56.06
data_12_4_2	6.127	4215	344	3871	192	1496	10	45	76.27
data_12_4_3	11.434	8642	208	8434	160	1470	9	41	59.42
data_12_4_4	3.544	6429	198	6231	160	1408	9	42	66.67
data_12_6_0	70.025	8845	406	8439	180	1271	9	39	57.35
data_12_6_1	62.186	1954	408	1546	276	1493	11	52	89.66
data_12_6_2	174.123	2251	532	1719	292	1925	11	60	89.55
data_12_6_3	62.948	528	528	0	344	2387	12	72	100.0
data_12_6_4	61.679	2625	422	2203	272	1846	11	53	88.33
data_14_4_0	10.294	6023	684	5339	228	1636	11	50	75.76
data_14_4_1	9.479	4376	600	3776	286	1473	12	43	78.18
data_14_4_2	11.663	7568	472	7096	224	1720	11	49	62.82
data_14_4_3	5.428	9744	260	9484	192	856	10	31	50.0
data_14_4_4	11.911	9015	420	8595	228	1509	11	48	60.0
data_14_6_0	122.656	6254	760	5494	316	1912	12	56	73.68
data_14_6_1	308.555	6889	778	6111	338	1615	12	51	72.86
data_14_6_2	237.558	5540	796	4744	328	1894	12	58	77.33
data_14_6_3	125.933	3713	568	3145	314	1973	12	58	81.69
data_14_6_4	159.806	6067	468	5599	272	1322	11	41	66.13
data_16_4_0	27.355	6334	800	5534	384	2551	14	71	80.68
data_16_4_1	20.97	9400	560	8840	324	2060	13	66	70.21
data_16_4_2	9.179	6083	680	5403	336	1616	13	46	71.88
data_16_4_3	22.18	7090	606	6484	336	1778	13	54	66.67
data_16_4_4	4.499	9664	286	9378	194	1167	10	33	44.59
data_16_6_0	196.66	11542	556	10986	290	1590	12	46	54.76
data_16_6_1	114.687	3485	806	2679	444	2682	14	81	90.0
data_16_6_2	319.275	8146	528	7618	344	1764	12	55	69.62
data_16_6_3	600.0	.	.	.	.	.	.	.	.
data_16_6_4	600.0	.	.	.	.	.	.	.	.
data_18_4_0	34.252	10910	656	10254	392	1811	14	50	56.82
data_18_4_1	53.99	2653	1352	1301	576	2472	17	87	94.57
data_18_4_2	19.462	12854	368	12486	320	1992	13	54	56.25
data_18_4_3	14.511	15922	522	15400	288	1915	12	52	50.0
data_18_4_4	69.997	9799	486	9313	390	1975	14	59	60.2
data_18_6_0	223.069	6615	1148	5467	536	2526	15	72	76.6
data_18_6_1	195.467	8506	842	7664	524	2943	15	83	75.45
data_18_6_2	600.0	.	.	.	.	.	.	.	.
data_18_6_3	208.451	9988	662	9326	470	2003	14	63	64.95
data_18_6_4	600.0	.	.	.	.	.	.	.	.
data_20_6_0	476.017	18159	976	17183	584	3039	16	84	61.31
data_20_6_1	600.0	.	.	.	.	.	.	.	.
data_20_6_2	600.0	.	.	.	.	.	.	.	.
data_20_6_3	485.055	10963	942	10021	594	2364	16	72	68.57
data_20_6_4	600.0	.	.	.	.	.	.	.	.
data_20_8_0	600.0	.	.	.	.	.	.	.	.
data_20_8_1	600.0	.	.	.	.	.	.	.	.
data_20_8_2	600.0	.	.	.	.	.	.	.	.
data_20_8_3	600.0	.	.	.	.	.	.	.	.
data_20_8_4	600.0	.	.	.	.	.	.	.	.
data_25_6_0	600.0	.	.	.	.	.	.	.	.
data_25_6_1	600.0	.	.	.	.	.	.	.	.
data_25_6_2	600.0	.	.	.	.	.	.	.	.
data_25_6_3	600.0	.	.	.	.	.	.	.	.
data_25_6_4	600.0	.	.	.	.	.	.	.	.
data_25_8_0	600.0	.	.	.	.	.	.	.	.
data_25_8_1	600.0	.	.	.	.	.	.	.	.
data_25_8_2	600.0	.	.	.	.	.	.	.	.
data_25_8_3	600.0	.	.	.	.	.	.	.	.
data_25_8_4	600.0	.	.	.	.	.	.	.	.

Table 5: Numerical results for formulation M (**tElapsed**: CPU time; **z**: aggregated cost; **zCost**: operational cost; **zPenalty**: penalty cost; **totTimeTrans**: total time transfert; **totQuanTrans**: total quantity of cargo transferred; **nTruckAssig**: number of trucks assigned to a dock; **nTransDone**: number of transfers of pallets between docks; **pTransDone**: percentage of transfers of pallets between docks).

C.2 Results for formulations G

fname	tElapsed	z	zCost	zPenalty	totTimeTrans	totQuanTrans	nTruckAssig	nTransDone	pTransDone
	sec						#	#	%
data_10_3_0	0.191	3105	38	3067	32	781	9	20	64.52
data_10_3_1	0.304	8410	112	8298	68	1030	7	28	46.67
data_10_3_2	0.263	6545	120	6425	52	949	8	28	58.33
data_10_3_3	0.245	10005	44	9961	32	498	6	15	31.25
data_10_3_4	0.383	9985	130	9855	56	968	7	27	45.76
data_12_4_0	2.997	13554	92	13462	64	973	8	30	40.54
data_12_4_1	3.997	7911	96	7815	72	1285	9	37	56.06
data_12_4_2	2.826	4032	161	3871	94	1496	10	45	76.27
data_12_4_3	4.875	8556	122	8434	92	1470	9	41	59.42
data_12_4_4	2.617	6353	122	6231	87	1408	9	42	66.67
data_12_6_0	32.222	8630	191	8439	89	1271	9	39	57.35
data_12_6_1	87.18	1722	176	1546	121	1493	11	52	89.66
data_12_6_2	192.302	2366	260	2106	153	1882	11	59	88.06
data_12_6_3	49.625	440	251	189	186	2366	12	71	98.61
data_12_6_4	22.671	2809	186	2623	121	1804	11	52	86.67
data_14_4_0	2.506	5627	288	5339	96	1636	11	50	75.76
data_14_4_1	1.673	3932	156	3776	80	1473	12	43	78.18
data_14_4_2	3.925	7560	192	7368	90	1686	11	48	61.54
data_14_4_3	1.896	9568	84	9484	60	856	10	31	50.0
data_14_4_4	3.489	8762	167	8595	104	1509	11	48	60.0
data_14_6_0	116.38	6380	292	6088	126	1858	12	55	72.37
data_14_6_1	88.715	6363	252	6111	119	1615	12	51	72.86
data_14_6_2	84.006	5076	332	4744	141	1894	12	58	77.33
data_14_6_3	54.847	3361	216	3145	132	1973	12	58	81.69
data_14_6_4	51.71	5756	157	5599	87	1322	11	41	66.13
data_16_4_0	5.437	5793	259	5534	135	2551	14	71	80.68
data_16_4_1	5.525	9050	210	8840	130	2060	13	66	70.21
data_16_4_2	2.787	5579	176	5403	94	1616	13	46	71.88
data_16_4_3	3.646	6649	165	6484	111	1778	13	54	66.67
data_16_4_4	2.219	9951	87	9864	58	1111	10	31	41.89
data_16_6_0	50.924	11162	176	10986	110	1590	12	46	54.76
data_16_6_1	61.577	3915	293	3622	175	2576	14	78	86.67
data_16_6_2	137.286	7799	174	7625	108	1767	12	51	64.56
data_16_6_3	141.809	4654	307	4347	162	2193	14	66	79.52
data_16_6_4	48.062	2597	324	2273	154	2268	15	69	88.46
data_18_4_0	4.472	10376	122	10254	80	1811	14	50	56.82
data_18_4_1	4.463	1987	358	1629	167	2431	17	86	93.48
data_18_4_2	6.018	12626	111	12515	100	2011	13	52	54.17
data_18_4_3	7.478	15611	211	15400	128	1915	12	52	50.0
data_18_4_4	8.419	9467	154	9313	116	1975	14	59	60.2
data_18_6_0	56.468	5972	321	5651	161	2503	15	71	75.53
data_18_6_1	59.063	8231	335	7896	208	2914	15	81	73.64
data_18_6_2	63.318	3965	316	3649	170	2728	16	81	86.17
data_18_6_3	74.137	9956	220	9736	154	1962	14	62	63.92
data_18_6_4	43.723	2979	238	2741	149	2197	16	68	87.18
data_20_6_0	119.347	17504	321	17183	208	3039	16	84	61.31
data_20_6_1	200.197	10825	349	10476	199	2896	16	86	69.92
data_20_6_2	116.617	4365	304	4061	206	2818	17	91	88.35
data_20_6_3	134.26	10432	235	10197	152	2348	16	71	67.62
data_20_6_4	116.442	10924	271	10653	154	2276	16	69	65.09
data_20_8_0	398.406	2176	353	1823	233	2730	19	87	93.55
data_20_8_1	600.0	.	.	.	.	.	.	.	.
data_20_8_2	600.0	.	.	.	.	.	.	.	.
data_20_8_3	600.0	.	.	.	.	.	.	.	.
data_20_8_4	600.0	.	.	.	.	.	.	.	.
data_25_6_0	251.293	7881	301	7580	188	2941	21	88	75.21
data_25_6_1	320.398	9627	267	9360	203	3150	22	95	76.0
data_25_6_2	301.192	6853	477	6376	247	3841	22	110	82.09
data_25_6_3	486.471	12345	456	11889	226	3271	20	99	68.75
data_25_6_4	197.149	10633	510	10123	271	3531	20	110	74.83
data_25_8_0	600.0	.	.	.	.	.	.	.	.
data_25_8_1	600.0	.	.	.	.	.	.	.	.
data_25_8_2	600.0	.	.	.	.	.	.	.	.
data_25_8_3	600.0	.	.	.	.	.	.	.	.
data_25_8_4	600.0	.	.	.	.	.	.	.	.

Table 6: Numerical results for formulation G (**tElapsed**: CPU time; **z**: aggregated cost; **zCost**: operational cost; **zPenalty**: penalty cost; **totTimeTrans**: total time transfert; **totQuanTrans**: total quantity of cargo transferred; **nTruckAssig**: number of trucks assigned to a dock; **nTransDone**: number of transfers of pallets between docks; **pTransDone**: percentage of transfers of pallets between docks).

C.3 Results for formulations 2M

fname	tElapsed sec	totTimeTrans	totQuanTrans	nTruckAssigned #	nTransfertDone #	pTransfertDone %
data_10_3_0	0.812	120	789	9	21	67.74
data_10_3_1	1.082	84	1030	7	28	46.67
data_10_3_2	1.544	102	949	8	28	58.33
data_10_3_3	0.764	46	498	6	15	31.25
data_10_3_4	1.492	80	968	7	27	45.76
data_12_4_0	11.51	128	995	8	30	40.54
data_12_4_1	12.974	148	1285	9	37	56.06
data_12_4_2	10.566	192	1496	10	45	76.27
data_12_4_3	15.674	160	1470	9	41	59.42
data_12_4_4	8.248	148	1408	9	42	66.67
data_12_6_0	116.325	180	1285	9	40	58.82
data_12_6_1	116.227	264	1493	11	52	89.66
data_12_6_2	158.53	284	1925	11	60	89.55
data_12_6_3	129.402	344	2387	12	72	100.0
data_12_6_4	75.455	264	1846	11	53	88.33
data_14_4_0	12.469	228	1636	11	50	75.76
data_14_4_1	13.827	280	1473	12	43	78.18
data_14_4_2	19.076	224	1720	11	49	62.82
data_14_4_3	11.069	174	856	10	31	50.0
data_14_4_4	22.0	224	1509	11	48	60.0
data_14_6_0	179.878	310	1912	12	56	73.68
data_14_6_1	372.014	328	1615	12	51	72.86
data_14_6_2	267.835	328	1894	12	58	77.33
data_14_6_3	155.441	290	1973	12	58	81.69
data_14_6_4	112.403	248	1322	11	41	66.13
data_16_4_0	46.507	384	2551	14	71	80.68
data_16_4_1	39.72	324	2060	13	66	70.21
data_16_4_2	18.136	324	1616	13	46	71.88
data_16_4_3	29.463	336	1778	13	54	66.67
data_16_4_4	11.37	186	1175	10	35	47.3
data_16_6_0	249.185	290	1591	12	47	55.95
data_16_6_1	162.236	440	2682	14	81	90.0
data_16_6_2	205.61	310	1767	12	51	64.56
data_16_6_3	438.476	454	2193	14	66	79.52
data_16_6_4	501.069	524	2268	15	69	88.46
data_18_4_0	45.011	386	1818	14	57	64.77
data_18_4_1	61.248	576	2472	17	87	94.57
data_18_4_2	40.208	320	2011	13	52	54.17
data_18_4_3	32.814	288	1915	12	52	50.0
data_18_4_4	66.205	386	1980	14	62	63.27
data_18_6_0	260.708	524	2526	15	72	76.6
data_18_6_1	392.485	508	2943	15	83	75.45
data_18_6_2	461.688	598	2728	16	81	86.17
data_18_6_3	295.336	454	2003	14	63	64.95
data_18_6_4	600.0	.	.	.	.	.
data_20_6_0	661.723	576	3039	16	84	61.31
data_20_6_1	600.0	.	.	.	.	.
data_20_6_2	363.176	660	2818	17	91	88.35
data_20_6_3	600.0	.	.	.	.	.
data_20_6_4	600.0	.	.	.	.	.
data_20_8_0	600.0	.	.	.	.	.
data_20_8_1	600.0	.	.	.	.	.
data_20_8_2	600.0	.	.	.	.	.
data_20_8_3	600.0	.	.	.	.	.
data_20_8_4	600.0	.	.	.	.	.
data_25_6_0	600.0	.	.	.	.	.
data_25_6_1	600.0	.	.	.	.	.
data_25_6_2	600.0	.	.	.	.	.
data_25_6_3	600.0	.	.	.	.	.
data_25_6_4	600.0	.	.	.	.	.
data_25_8_0	600.0	.	.	.	.	.
data_25_8_1	600.0	.	.	.	.	.
data_25_8_2	600.0	.	.	.	.	.
data_25_8_3	600.0	.	.	.	.	.
data_25_8_4	600.0	.	.	.	.	.

Table 7: Numerical results for formulation 2M (**tElapsed**: CPU time; **totTimeTrans**: total time transfert; **totQuanTrans**: total quantity of cargo transferred; **nTruckAssig**: number of trucks assigned to a dock; **nTransDone**: number of transfers of pallets between docks; **pTransDone**: percentage of transfers of pallets between docks).

C.4 Results for formulations 2G

fname	tElapsed sec	totTimeTrans	totQuanTrans	nTruckAssigned #	nTransfertDone #	pTransfertDone %
data_10_3_0	0.342	29	781	9	20	64.52
data_10_3_1	0.787	68	1030	7	28	46.67
data_10_3_2	0.646	52	949	8	28	58.33
data_10_3_3	0.543	32	498	6	15	31.25
data_10_3_4	0.79	55	968	7	27	45.76
data_12_4_0	9.707	63	980	8	29	39.19
data_12_4_1	4.656	68	1285	9	37	56.06
data_12_4_2	3.027	94	1496	10	45	76.27
data_12_4_3	7.113	92	1470	9	41	59.42
data_12_4_4	4.217	87	1408	9	42	66.67
data_12_6_0	33.861	91	1285	9	40	58.82
data_12_6_1	36.12	120	1493	11	52	89.66
data_12_6_2	84.927	139	1882	11	59	88.06
data_12_6_3	73.181	176	2366	12	46	63.89
data_12_6_4	30.708	119	1804	11	34	56.67
data_14_4_0	4.083	96	1636	11	50	75.76
data_14_4_1	3.177	80	1473	12	43	78.18
data_14_4_2	6.195	87	1686	11	48	61.54
data_14_4_3	3.611	55	856	10	31	50.0
data_14_4_4	7.948	93	1509	11	48	60.0
data_14_6_0	102.004	117	1858	12	55	72.37
data_14_6_1	152.031	113	1615	12	51	72.86
data_14_6_2	79.687	138	1894	12	58	77.33
data_14_6_3	51.298	122	1973	12	58	81.69
data_14_6_4	36.628	87	1322	11	41	66.13
data_16_4_0	10.421	135	2551	14	71	80.68
data_16_4_1	11.445	129	2060	13	66	70.21
data_16_4_2	3.997	88	1616	13	46	71.88
data_16_4_3	6.654	102	1778	13	54	66.67
data_16_4_4	3.852	61	1119	10	33	44.59
data_16_6_0	208.463	97	1591	12	47	55.95
data_16_6_1	91.301	170	2576	14	78	86.67
data_16_6_2	44.587	104	1767	12	51	64.56
data_16_6_3	83.354	151	2193	14	66	79.52
data_16_6_4	46.325	152	2268	15	69	88.46
data_18_4_0	9.194	99	1812	14	56	63.64
data_18_4_1	10.164	164	2431	17	86	93.48
data_18_4_2	11.879	97	2011	13	52	54.17
data_18_4_3	11.257	126	1915	12	52	50.0
data_18_4_4	11.493	121	1980	14	62	63.27
data_18_6_0	98.589	158	2503	15	71	75.53
data_18_6_1	91.453	182	2914	15	82	74.55
data_18_6_2	70.69	170	2728	16	81	86.17
data_18_6_3	79.34	150	1962	14	62	63.92
data_18_6_4	53.487	146	2197	16	68	87.18
data_20_6_0	227.273	194	3039	16	84	61.31
data_20_6_1	353.924	195	2902	16	88	71.54
data_20_6_2	97.591	194	2818	17	91	88.35
data_20_6_3	103.752	141	2351	16	40	38.1
data_20_6_4	194.63	159	2278	16	69	65.09
data_20_8_0	599.659	202	2730	19	87	93.55
data_20_8_1	600.0	.	.	.	.	.
data_20_8_2	600.0	.	.	.	.	.
data_20_8_3	600.0	.	.	.	.	.
data_20_8_4	600.0	.	.	.	.	.
data_25_6_0	191.679	184	2941	21	88	75.21
data_25_6_1	251.288	187	3150	22	95	76.0
data_25_6_2	232.982	246	3841	22	110	82.09
data_25_6_3	579.702	206	3271	20	99	68.75
data_25_6_4	231.332	262	3552	20	112	76.19
data_25_8_0	600.0	.	.	.	.	.
data_25_8_1	600.0	.	.	.	.	.
data_25_8_2	600.0	.	.	.	.	.
data_25_8_3	600.0	.	.	.	.	.
data_25_8_4	600.0	.	.	.	.	.

Table 8: Numerical results for formulation 2G (**tElapsed**: CPU time; **totTimeTrans**: total time transfert; **totQuanTrans**: total quantity of cargo transferred; **nTruckAssig**: number of trucks assigned to a dock; **nTransDone**: number of transfers of pallets between docks; **pTransDone**: percentage of transfers of pallets between docks).

References

Cécilia Daquin, Hamid Allaoui, Gilles Goncalves, and Tienté Hsu. Variable neighborhood search based algorithms for crossdock truck assignment. *RAIRO-Oper. Res.*, 55:S2291–S2323, 2021. doi: 10.1051/ro/2020087. URL https://doi.org/10.1051/ro/2020087.

Matthias Ehrgott. *Multicriteria Optimization*. Springer Berlin, Heidelberg, 2nd edition, 2005. ISBN 3540213988.

Shahin Gelareh. A note on "variable neighborhood search based algorithms for crossdock truck assignment". *RAIRO-Operations Research*, 55(5):2763–2768, 2021. doi: 10.1051/ro/2021132. URL https://doi.org/10.1051/ro/2021132.

Shahin Gelareh, Gilles Goncalves, and Rahimeh Neamatian Monemi. On truck dock assignment problem with operational time constraint within cross docks, 2015. Optimization Online. 18 pages. https://optimization-online.org/2014/11/4626/.

Shahin Gelareh, Rahimeh Neamatian Monemi, Frédéric Semet, and Gilles Goncalves. A branch-and-cut algorithm for the truck dock assignment problem with operational time constraints. *European Journal of Operational Research*, 249(3):1144–1152, 2016. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2015.09.049. URL https://www.sciencedirect.com/science/article/pii/S0377221715008917.

Shahin Gelareh, Fred Glover, Oualid Guemri, Saïd Hanafi, Placide Nduwayo, and Raca Todosijević. A comparative study of formulations for a cross-dock door assignment problem. *Omega*, 91: 102015, 2020. ISSN 0305-0483. doi: https://doi.org/10.1016/j.omega.2018.12.004. URL https://www.sciencedirect.com/science/article/pii/S0305048317311908.

Abby Jenkins. What is Cross-Docking? Definition, Types & Advantages, 2023. https://www.netsuite.com/portal/resource/articles/inventory-management/cross-docking.shtml.

Ilker Kucukoglu and Nursel Ozturk. A mathematical model for truck-door assignment and product placement problem in cross-docking center. *Global Journal of Business, Economics and Management: Current Issues*, 7(1):135–142, 2017.

Anne-Laure Ladier. *Planification des opérations de cross-docking : prise en compte des incertitudes opérationnelles et de la capacité des ressources internes.* PhD thesis, Grenoble-INP (France), 2014.

Andrew Lim, Zhaowei Miao, Brian Rodrigues, and Zhou Xu. Transshipment through crossdocks with inventory and time windows. *Naval Research Logistics (NRL)*, 52(8):724–733, 2005. doi: https://doi.org/10.1002/nav.20113. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20113.

Andrew Lim, Hong Ma, and Zhaowei Miao. Truck dock assignment problem with time windows and capacity constraint in transshipment network through crossdocks. In Marina Gavrilova, Osvaldo Gervasi, Vipin Kumar, C. J. Kenneth Tan, David Taniar, Antonio Laganá, Youngsong Mun, and Hyunseung Choo, editors, *Computational Science and Its Applications - ICCSA 2006*, pages 688–697, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-34076-8.

Belén Meliàn-Batista. Variable neighborhood search: Application in cross-docking optimization. In *MIC-2024 - 15th Metaheuristics International Conference (Lorient, France)*. 2024.

Zhaowei Miao, Andrew Lim, and Hong Ma. Truck dock assignment problem with operational time constraint within crossdocks. *European Journal of Operational Research*, 192(1):105–115, 2009. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2007.09.031. URL https://www.sciencedirect.com/science/article/pii/S0377221707009393.

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

Placide Nduwayo. *Mathematical Programming Formulations and Algorithms for the Cross-dock Door Assignment Problem*. PhD thesis, Université Polytechnique Hauts-de-France (France), 2020.

Louis Y. Tsui and Chia-Hao Chang. An optimal solution to a dock door assignment problem. *Computers & Industrial Engineering*, 23(1):283–286, 1992. ISSN 0360-8352. doi: https://doi.org/10.1016/0360-8352(92)90117-3. URL https://www.sciencedirect.com/science/article/pii/036083529290117З.

Jan Van Belle, Paul Valckenaers, and Dirk Cattrysse. Cross-docking: State of the art. *Omega*, 40(6): 827–846, 2012. ISSN 0305-0483. doi: https://doi.org/10.1016/j.omega.2012.01.005. URL https://www.sciencedirect.com/science/article/pii/S0305048312000060. Special Issue on Forecasting in Management Science.

Lijuan Zhang. *Optimization and simulation of a cross-docking terminal*. PhD thesis, Centrale Lille (France), 2016.

Yi-Rong Zhu. *Recent Advances and Challenges in Quadratic Assignment and Related Problems*. PhD thesis, University of Pennsylvania (USA), 2007.

Yi-Rong Zhu, Peter M. Hahn, Ying Liu, and Monique Guignard-Spielberg. New approach for the cross-dock door assignment problem. In *XLI SBPO 2009 - Pesquisa Operacional na Gestão do Conhecimento*, pages 1226–1236. 2009.