Graphical Abstract

The Truck-to-dock Door Assignment Problem: analysis and discussion of IP formulations

Nicolas Forget, Elizabeth Gandibleux, Xavier Gandibleux, Valentin Guy–Deroubaix, Awen Jacq–Bodet

Highlights

The Truck-to-dock Door Assignment Problem:
analysis and discussion of IP formulations

Nicolas Forget, Elizabeth Gandibleux, Xavier Gandibleux, Valentin Guy–Deroubaix, Awen Jacq–Bodet

- Research highlight 1
- Research highlight 2

The Truck-to-dock Door Assignment Problem: analysis and discussion of IP formulations

Nicolas Forget[a], Elizabeth Gandibleux[b], Xavier Gandibleux[b,*], Valentin Guy–Deroubaix[b], Awen Jacq–Bodet[b]

[a] *Johannes Kelper University Linz, Altenberger Straße 69, Linz, 4040, Austria*
[b] *Nantes University, 2, rue de la Houssiniere BP 92208, Nantes, 44322 Cedex 03, France*

Abstract

This paper is devoted to (résumé posé à adapter)...

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

*Corresponding author

Email addresses: `Nicolas.Forget@jku.at` (Nicolas Forget), `Elizabeth.Gandibleux@etu.univ-nantes.fr` (Elizabeth Gandibleux), `Xavier.Gandibleux@univ-nantes.fr` (Xavier Gandibleux), `valentin.Guy-Deroubaix@etu.univ-nantes.fr` (Valentin Guy–Deroubaix), `Awen.Jacq--Bodet@etu.univ-nantes.fr` (Awen Jacq–Bodet)

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

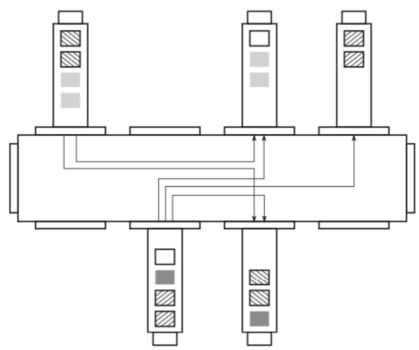


Figure 1: Illustration from Van Belle et al. (2012) of a cross-dock where the layout of the warehouse presents a I-shape. Ten docks are available (flat rectangles) to serve as inbound or outbound of products. Five trucks are docked (long rectangles). A pallet of a given product is represented by a rectangle 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.

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 docked, the pallets get unloaded, sorted and screened to identify their end destinations. Afterward, the pallets are moved to outbound dock doors of 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 Van Belle et al. (2012); Nduwayo (2020)). 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 number of pallets transfered, and to minimize the time required to perform the 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 a short, where and when should the trucks be processed. Thus a model describing this optimization problem must take into account at minima (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 crossdock 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 et al., 2009). On the basis of the contributions of Lim et al. (2005, 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 it is 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 are reporting using CPLEX. Observing that CPLEX faces resolution difficulties, they have also proposed a tabu search and a genetic algorithm approach to solve it.

Van Belle et al. (2012) present a vast 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 promizing directions to improve and extend the contributions are suggested.

Several PhD thesis (e.g. Zhu (2007); Ladier (2014); Zhang (2016); Nassief (2017); Nduwayo (2020)) are devoted to decision problems related to cross-docking. For example, Nduwayo's contributions (Gelareh et al., 2020; Nduwayo, 2020) are devoted to the "Cross-Docking Assignment Problem (CDAP)", i.e. the problem to assign origins to inbound doors and destinations to outbound doors so that the total cost inside the cross-dock platform is minimized (see e.g. Tsui and Chang (1992); Zhu et al. (2009) or recently Meliàn-Batista (2024)). Nduwayo proposes original Mixed-Integer Programming models, and he conductes an extensive comparative analysis on benchmark instances from the literature.

Gelareh et al. (2015) underline in a technical report available online weaknesses and shortcomings in the formulation M. On the basis of the latter, they propose a revised IP formulation for the TDAP denoted "formulation G" in the rest of the paper. Several classes of valid inequalities are also introduced, and exact separation algorithms are described for separating cuts for classes with 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 et al. (2016).

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

Daquin et al. (2021) have published recently a paper devoted to the TDAP where the algorithms presented are based on the Variable Neighborhood Search metaheuristic. However, this work is based on the 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 the 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 (Miao et al., 2009; Gelareh et al., 2015, 2016; Gelareh, 2021) has been achieved. It has revealed inconsistencies in the arguments put forward in (Gelareh et al., 2015, 2016; Gelareh, 2021), as well as to observe several vagueness and incompleteness in the formulation G. In particular, the proposed amendments in (Gelareh, 2021) does not address accurately all deficiencies raised in the formulation M, and the numerical results provided in (Gelareh et al., 2016) are not replicable.

On base of these four scientific documents questioned on several aspects, the first contribution of this paper is devoted to an analysis and a discussion of the M and G formulations and the optimal solutions obtained by them, aiming to provide to the readers (1) an advanced understanding of the formulations, (2) a corrected formulation of G, and (3) an understanding of the optimal solutions collected with the two formulations. Also, an alternative formulation derived from G and considered less simplifying assumptions in the operations, named 2R, is proposed. 2R is a a bi-objective variant where (1) one constraint of G is modified and (2) the formulation's abilities to simultaneously optimize independently the two conflicting objectives is explored.

The second contribution of this paper concerns numerical experiments conducted with these formulations. Miao et al. (2009) describe the characteristics used for creating numerical instances generated to evaluate the algorithms proposed in their paper, but the corresponding datafiles are not provided. Gelareh et al. (2016) have generated datafiles according the indications provided in (Miao et al., 2009) for evaluating their algorithms. They are available online. We refer to these instances in the following as "full mesh" instances. In addition, a new set of numerical instances inspired from a real crossdock warehouse and plannings of trucks where the management of pallets is more realistic have been generated. These instances will be referred to in the following as "bipartite" instances. Numerical experiments have been conducted using these two distinct sets of instances, allowing the evaluation of the formulation's performance across various scenarios.

The rest of the paper is organised as follow. Section 2 presents the notations and the definitions of the parameters. In order to facilitate the discussions about formulations, the notations used for the formulations have been unified. 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. Formulation M is presented in Section 3, followed by criticism found in the literature in Section 3.1. Next, formulations G and 2R are respectively introduced in Section 5 and Section 6. The three formulations are rigorously described, and come with a comments and a discussion. Numerical experiments are reported in Section 7. The two sets of instances are presented and the results collected are reported and analysed. Finally, Section 8 gives a conclusion and draws several perspectives. The formulations are implemented in Julia (Bezanson et al., 2017) with JuMP (Lubin et al., 2023), and they are provided in Appendix B. Also, in order to allow the readers to reproduce all results presented in this paper, all the material produced, i.e. codes in Julia, formulations in JuMP, and datasets in raw text files, are available online on GitHub at `https://github.com/xgandibleux/TDAP`.

2. Notations, Parameters, Example

2.1. Parameters

The following parameters are used in the formulations:

- N: set of trucks arriving at and/or departing from the crossdock
- M: set of docks available in the crossdock
- n: total number of trucks, that is $|N|$
- m: total number of docks, that is $|M|$
- 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
- $\hat{x}_{ij}$: 1 iff truck i departs no later than truck j arrives; 0 otherwise

It is reasonable and no restrictive to adopt the following considerations:

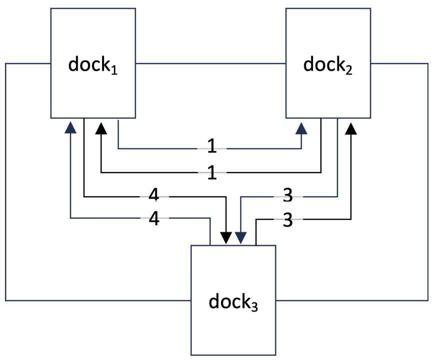
1. $f_{ij} \geq 0$ iff $d_j \geq a_i$ $(1 \leq i, j \leq n)$, otherwise $f_{ij} = 0$
   It means that truck i will transfer some cargo to truck j iff truck j departs no earlier than truck i arrives;
2. $a_i < d_i$ $(1 \leq i \leq n)$
   It means that for each truck, the arrival time should strictly smaller than its departure time;
3. $n > m$
   It considers the over-constrained condition.

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

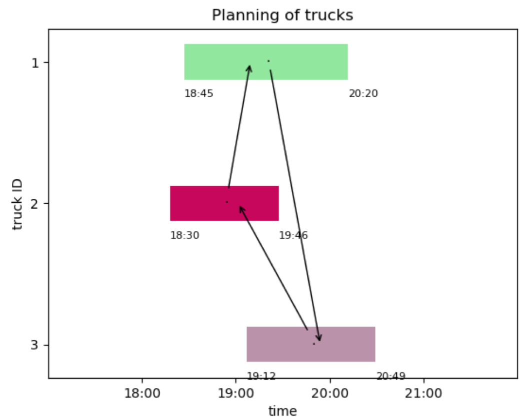
1. sort all a_i and d_i in an increasing order;
2. let τ_r $(r = 1, 2, ..., 2n)$ be these $2n$ numbers such that $\tau_1 \leq \tau_2 \leq ... \leq \tau_{2n}$.

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 crossdock. Values on arrows report the transfert time t_{kl} between docks k and l.



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

Figure 2: Data of the crossdock and the planning of trucks for the illustration example

Regarding the cross-dock.

- It is composed of:

  $M = \{1, 2, 3\}$, three docks ($m = 3$) according to the layout illustrated in Figure 2a.

- The times of transfer between docks (in minutes, $0 \leq \text{minute} \leq 59$) are given by:

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

- There is no restriction on the capacity (in number of pallets) in this example:

  $C = \infty$

Regarding the previsional planning of trucks and transfers of pallets.

- The scenario is composed of:

  $N = \{1, 2, 3\}$, three trucks ($n = 3$)

- The hours of arrival and of departure (in format hour.minute, $0 \leq$ hour ≤ 23, $0 \leq$ minute ≤ 59) are known and are, respectively for each trucks:

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

- The transfer of pallets between trucks:

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

Regarding the costs.

- The operational costs and the penalty costs (in euros, at the format unit.cents) are respectively:

$$c = \begin{pmatrix} 0.0 & 1.0 & 1.0 \\ 1.0 & 0.0 & 2.0 \\ 1.0 & 2.0 & 0.0 \end{pmatrix} \quad \text{and} \quad p = \begin{pmatrix} 0.0 & 0.0 & 52.0 \\ 24.0 & 0.0 & 0.0 \\ 0.0 & 23.0 & 0.0 \end{pmatrix}$$

Regarding the derivated data.

- The vector $\hat{x}$ indicates if a pair of trucks has a conflict or not between their times windows; the value `1` means "no conflict" and the value `0` shows "a conflict":

$$\hat{x} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ states } \begin{cases} \text{the time window of truck 1 is in conflict with trucks 2 and 3} \\ \text{the time window of truck 2 is in conflict with trucks 1 and 3} \\ \text{the time window of truck 3 is in conflict with trucks 1 and 2} \end{cases}$$

- The vector τ collects all the hours of arrival and departure in one sorted vector:

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

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

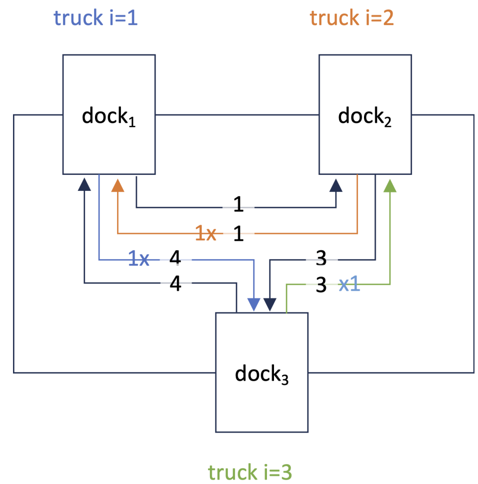


Figure 3: Assignment of trucks to docks and the corresponding transfers of pallets within the cross dock.

A feasible solution.

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

– 1 pallet of truck 3 (dock 3) to truck 2 (dock 2):
transfer time of 3 minutes (= 3 minutes x 1 pallet)

which make a total of $4 + 1 + 3 = 8$ unit of time (minutes) for all the pallets. All the pallets have been transferred, no penalty appears in this example.

3. Formulation M

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

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

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

The model is thus formulated as:

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

$$\text{s.t.} \quad \sum_{k=1}^{m} y_{ik} \leq 1 \qquad (1 \leq i \leq n) \tag{2}$$

$$z_{ijkl} \leq y_{ik} \qquad (1 \leq i, j \leq n;\ 1 \leq k, l \leq m) \tag{3}$$

$$z_{ijkl} \leq y_{jl} \qquad (1 \leq i, j \leq n;\ 1 \leq k, l \leq m) \tag{4}$$

$$y_{ik} + y_{jl} - 1 \leq z_{ijkl} \qquad (1 \leq i, j \leq n;\ 1 \leq k, l \leq m) \tag{5}$$

$$\hat{x}_{ij} + \hat{x}_{ji} \geq z_{ijkk} \qquad (1 \leq i, j \leq n, i \neq j;\ 1 \leq k \leq m) \tag{6}$$

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i \in \{i : a_i \leq \tau_r\}}\sum_{j=1}^{n} f_{ij}\, z_{ijkl} - \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j \in \{j : d_j \leq \tau_r\}} f_{ij}\, z_{ijkl} \leq C \qquad (1 \leq r \leq 2n) \tag{7}$$

$$f_{ij}\, z_{ijkl}\,(d_j - a_i - t_{kl}) \geq 0 \qquad (1 \leq i, j \leq n;\ 1 \leq k, l \leq m) \tag{8}$$

$$y_{ik} \in \{0,1\},\ y_{jl} \in \{0,1\},\ z_{ijkl} \in \{0,1\} \qquad (1 \leq i, j \leq n;\ 1 \leq k, l \leq m) \tag{9}$$

where:

(1) is the objective function, which is composed by the total operational cost (first term) and the total penalty cost (second term);

(2) ensures that each truck cannot be assigned to more than one dock;

(3-5) jointly determine the variable z which represent the logic relationship among y_{ik}, y_{jl} and z_{ijkl};

(6) assures that one dock cannot be occupied by two trucks simultaneously;

(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;

(8) assures that the transfers 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;

(9) states the integrality of decision variables.

The implementation of this formulation in Julia with JuMP is given in Appendix B.1 and on a GitHub repository available online at `https://github.com/xgandibleux/TDAP`.

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. Those two parts being non-commensurable, they are converted in monetary unit, considering an operationnal 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 of the two parts. Also, only a single solution may be obtained at the end of the resolution as output of this single objective problem. However, the two parts are by nature conflictual. Indeed, to minimize transfer times, just don't transfer a pallet, which goes against minimizing the number of undelivered pallets. Section 6 gives the formulation 2R of the TDAP from the perspective of multi-objective optimization. The optimal solutions in the sense of multiple objective optimization (named the efficient solutions and represent compromises regarding the two objectives) are discussed in Section 7.

3.1.2. The notion of capacity in expression (7)

The formulation does not help to put in touch the C value with the concrete resource limitations in the cross-dock. Is this value 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.

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

The expression (8) raises questions. Indeed, 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 are not gathered to perform this transfer, and (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 $f_{ij} \times a_i$. What is the meaning of multiplying a time marker related a truck movement with the number of pallets? In the model 2R described in Section 6, this constraint is modified to clarify this issue.

4. Literature criticisms about M formulation

According to Gelareh et al. (2015); Gelareh (2021), the M formulation 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 dedicated corrections are reported hereafter.

4.1. Mains criticisms and given corrections

4.1.1. No temporary storage

If there are transfers from a truck i to another truck j and from j to i, and one of them is not feasible, then the constraint (5) invalidates both of them. As a consequence of this, if given their time windows truck i can transfer pallets to the truck j, these transfers are also not achieved. This is due to the fact that, to make a pallet transfer feasible between two trucks in formulation M, they must be docked with a common period of time which must be long enough for completing the transfer.

The authors consider this restriction as a limitation. For them, pallets from truck i may be temporary stored into the cross-dock in waiting for the arrival of truck j to pick them up, even if i has already left the cross-dock. The authors introduce a notion of "buffer" and they come with modifications on the M formulation so as to allow temporary storage of pallets.

4.1.2. Bi-directional transfer

There is a link between z_{ijkl} and z_{jilk}. Indeed, there is a bi-directional transfer between the two trucks i and j when they share a common time window while they are docked, i.e. $z_{ijkl} = z_{jilk}$ holds. This means that there is an equivalent solution where truck j is assigned to dock l and truck i is assigned to dock k, in which $z_{ijkl} = z_{jilk}$ also holds.

To address this issue, the definition of these variables has been revised so that $z_{ijkl} = 1$ if and only if there is an actual transfer from i to j. Furthermore, using a temporary storage makes the constraint (6) obsolete. It comes from the fact that (i) contrary to the model M, a same dock k can now be used to do a transfer from truck i to truck j and (ii) j does not need to share a common docking time with i and can arrive at any time after the departure of i. Thus, the constraint (6) is revised so that a transfer is able to use the buffer to make deliveries on a same dock k.

4.1.3. Extra transportation costs

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

We saw that a bidirectional transfer raises a problem if, for i, j such that $i \neq j$, $f_{ij} = 0$ and $f_{ji} > 0$ (case where truck j leaves before truck i arrives). In such occurences, j will drop its cargo in cross-dock buffer for i can pick it up later. However, since $f_{ij} = 0$, constraint (8) is satisfied without any restriction on z_{ijkl}, if for some k, l $z_{jilk} = 1$, 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.

To further correct the model, constraint (5) is removed and replaced by new constraints that allow for non-overlaping time windows in the trucks' docking time. Then, together with constraint (6), a transfert between two distinct trucks on a same dock can be performed using the temporary storage. Also, the constraint (8) has been rewritten so that $z_{ijkl} = 0$ when $f_{ij} = 0$ and when the time window is too short to make a delivery.

All these modifications led to a revised formulation proposed by the authors and discussed in Section 5.

4.2. Discussion

4.2.1. Concerning the temporary storage

When assuming that a truck i can unload and store its pallets in the crossdock while waiting the arrival of the truck j to pick them up later, the authors assume the existence of a buffer area to temporarily store the pallets. However, this assumption is related to the operational situation to consider. Indeed, a crossdock may not have such buffer zones to store cargo, or cargo may not be stored, even for a short period of time, e.g. when the cargo are subject to cold chain constraint.

This option is not considered in (Miao et al., 2009), and it changes the nature of the problem considered. Thus it cannot be viewed as a weakness of the formulation M. In other words, formulations M and G consider two different variants of the TDAP.

4.2.2. Inaccuracies in documents

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

First, the authors use in these documents an example to demonstrate that the M formulation does not compute all the feasible solutions. The example is built so that M find no solution and forbid all transfers while the G formulation should find solution.

- With the 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 transferts to consider, a penalty value of 1 for each unfullfiled transfert, and 2 pallets per transfert, the penalty term gives:

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

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

  Here 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, the formulation G published by the authors does not allow to reproduce the results reported in the papers. Indeed, the description of the formulation given in both documents is the same but it contains several errors and missing information to implement it properly, see Section 5 for details. Some corrections of these deficiencies can be inferred with certainty, others are subject to interpretation. After correcting the errors identified, the fixed formulation was implemented in JuMP (see Appendix B.2) and solved to optimality using GUROBI as MIP solver and using the datasets (see Section 7.1.1) provided online by these authors. Unfortunately none of the numerical experiments has led us to the optimal value announced in papers for the given instances. Because the authors do not mention any open access to their codes, it is not possible to determine the origin of divergence between both implementations. This fixed formulation will be the basis for the formulation 2R proposed and presented in Section 6.

5. Formulation G

The IP model corresponding to the formulation G uses the same definition of the decision variables of the formulation M. The formulation is reproduced exactly as described in (Gelareh et al., 2015, 2016; Gelareh, 2021) in Appendix A.1, with errors and missing information. Indeed, a number of inaccuracies and incompleteness were identified in formulation G, making this formulation non-operational. They are listed and a proposed correction is reported in Appendix A.2.

Taken individually each of these points can be corrected quite easily once identified. However, some are not immediately identifiable, which could only be found after a meticulous analysis of the flat formulation, which necessitated testing instances by hand. In addition, when the indices are missing or in excess, or when their values are not stated, this leads to confusion and ambiguity of expression, exposing us to the possibility of deviating from the authors' original intentions.

We have therefore made the corrections, which have enabled us to obtain a formulation that provides consistent solutions. Unfortunately, we were unable to reproduce the results announced by the authors. The implementation in Julia with JuMP after having fixed issues of this formulation is given in Appendix B.2 and on a GitHub repository available online at `https://github.com/xgandibleux/TDAP`.

5.1. Discussion

5.1.1. Storage capacity

The question of the overall capacity constraint expressed by

$$\sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i\in\{i:a_i\leq\tau_r\}}\sum_{j=1}^{n} f_{ij}\, z_{ijkl} - \sum_{k=1}^{m}\sum_{l=1}^{m}\sum_{i=1}^{n}\sum_{j\in\{j:d_j\leq\tau_r\}} f_{ij}\, z_{ijkl} \leq C \qquad \forall r \in \{1,2,...,2n\} \quad (7)$$

which does not discriminate the allowed storage in the formulation G. This point has already been discussed in the analysis of formulation M. This concern is more relevant in formulation G, in particular when the time windows of trucks i and j have no intersection. In this situation, 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) .

5.1.2. The case of pallets not transfered.

With formulations M and G, and numerical instances used in papers Gelareh et al. (2016); Miao et al. (2009), in an optimal solution, when a pallet cannot be transferred, it remains in the truck. However a capacity problem may occurs here. Indeed, if this truck is initially planned to leave the crossdock fully loaded with pallets issued from other trucks, an infeasible situation occurs (truck loaded over its capacity) but is not managed by the model. A way to deal with these cases is to consider planning with incoming and outcoming trucks, where incoming trucks do not receive pallets. Thus, a pallet not transferred is returned to the furnisher. The new numerical instances that we produced (see Subsection 7.1.2) are inspired from existing instances for the CDAP, and then are compliant with this policy.

6. Formulation 2R

The formulation 2R (R stand for Revised, 2 for two objectives) is now presented. Founded on the formulation G where all the corrections that we proposed are integrated, this formulation is basically a rigorous description of the formulation G revisited. Indeed, it comes with a redefinition of one constraint (referred by expression (8) hereafter), plus several changes underlined along the discussions in the preceding sections.

The decision variables are identical to the formulations M and G. The expression (1) giving the objective function in of M and G is spitted in two parts, giving the two objectives functions without the artifacts c_{kl} and p_{ij}: (1.1) minimize the transfer time of the pallets in the cross-dock and (1.2) minimize the number of undelivered pallets. The IP model is formulated as follow:

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

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} f_{ij}\left(1 - \sum_{k=1}^{m}\sum_{l=1}^{m} z_{ijkl}\right) \tag{1.2}$$

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

$$z_{ijkl} \leq y_{ik} \qquad \forall i \in N,\ j \in N,\ k \in M,\ l \in M \tag{3}$$

$$z_{ijkl} \leq y_{jl} \qquad \forall i \in N,\ j \in N,\ k \in M,\ l \in M \tag{4}$$

$$y_{ik} + y_{jk} \leq 1 + \hat{x}_{ij} + \hat{x}_{ji} \qquad \forall i \in N,\ j \in N,\ k \in M,\ i \neq j \tag{5}$$

$$z_{ijkk} \leq \hat{x}_{ij} \qquad \forall i \in N,\ j \in N,\ k \in M,\ i \neq j \tag{6}$$

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

$$z_{ijkl} = 0 \qquad \forall i \in N,\ j \in N,\ k \in M,\ l \in M,\ i \neq j,\ (d_j - a_i - f_{ij}\, t_{kl} \leq 0) \tag{8}$$

$$y_{ik} \in \{0, 1\} \qquad \forall i \in N,\ k \in M \tag{9}$$

$$z_{ijkl} \in \{0, 1\} \qquad \forall i \in N,\ j \in N,\ k \in M,\ l \in M \tag{10}$$

Compared to the formulation G corrected, the constraint (8) is adapted in the formulation 2R. Indeed, in the formulation G, the parameter t_{kl} is considered as a global value. This value is initially established independently of the quantity of pallets really convoyed from k to l. In formulation 2R, t_{kl} is the unitary time required by a forklift to do a round trip when it moves a pallet from k to l. With this adaptation, the formulation takes into consideration the real quantity f_{ij} of pallets convoyed into account in the computation of the time required for transferring all the pallets from k to l.

The implementation of this formulation in Julia with JuMP is given in Appendix B.3 and on a GitHub repository available online at `https://github.com/xgandibleux/TDAP`.

6.1. Comments on the formulation 2R

A crucial aspect to grasp for an overall understanding of the formulation 2R is the role that represents the matrices t_{kl} and f_{ij}. By making these two unitary values, we can represent the transfers as a "flow" of pallets ($t_{kl} \times f_{ij}$) which was not possible with formulations M and G, t_{kl} and f_{ij} being independent values in those formulations. Another aspect of these two matrices is that their unit is not always specified in papers on M and G. For the formulation 2R and the associated instances, we consider t_{kl} as fractional hours and f_{ij} as a number of pallets.

7. Numerical experiments

7.1. Numerical Instances

7.1.1. Instances "full mesh"

The experiments reported by Miao et al. (2009) have been conducted with a collection of instances that represent a crossdock 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 minutes at minimum and 74 minutes at maximum;
- the value f_{ij} giving the number of pallets to be shipped is randomly chosen between 6 and 60.
- the duration of the transfert 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".

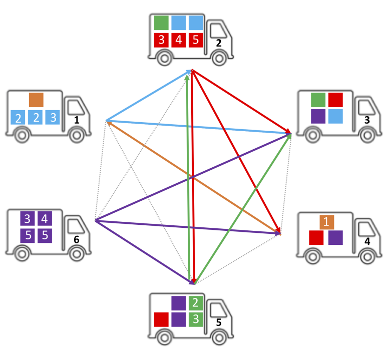


Figure 4: Example of a full-mesh transfer of pallets. Trucks arrives with pallets from furnishers, and leave with pallets to customers. For example, truck numbered 2 arrives with three (red) pallets destined to trucks 3/4/5, and leaves with pallets provided by truck 1 (blue) and truck 5 (green).

On base of their size $n \times m$, 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, datafiles are not provided by the authors.

For the paper (Gelareh et al., 2015), authors have generated instances following generation procedure described below. For the t_{kl}, no proportional conversion is considered, only the Manhattan distance is kept. The datafiles generated are available online[1] 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. This led us to develop a new set of instances following the generation procedure presented in the next section.

7.1.2. Instances "bi-partite"

Based on data collected and reported in Appendix B.4, new instances aiming to be realistic are proposed for the TDAP.

Main differences with instances presented in Section 7.1.1 concerns the trucks. First, a distinction is made between "incoming trucks", those who carry pallets issued from furnishers (factories, other warehouses, etc.), and "outcoming trucks", those who carry pallets to customers (shops, other warehouses, etc.). Next, the fleet of trucks is heterogeneous in the sense of the capacity of incoming trucks is larger than the outcoming ones. Finally, incoming trucks do not receive pallets from any other trucks. Only outcoming trucks receive pallets. Thus the transfer of pallets can be seen following a bi-partite graph (see Figure 5), giving the name of this new collection.

The instances have been generated according this set of rules:

- A crossdock has a layout of "I" and presents m docks, with $4 \leq m \leq 14$.
- The distance $dist(k, l)$ between two docks k and l is based on the Manhattan distance. Measures are deduced from the physical configuration of the warehouse.
- The crossdock has no fix equipment inside. It is viewed as an empty building where the pallets are transfered by forklifts.

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

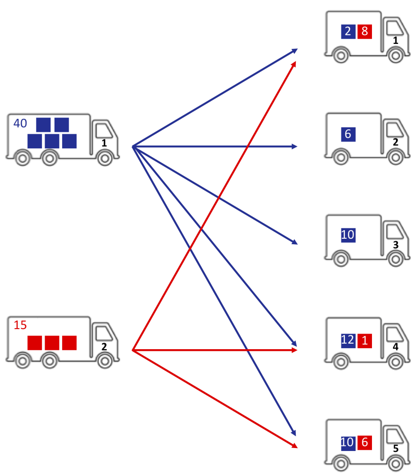


Figure 5: Example of a bipartite transfer of pallets. Incoming (and biggest) trucks (on the left) carry pallets from furnishers, outcoming (and smallest) trucks (on the right) carry pallets to customers.

- The transfer time between two docks k and l is computed with the formula $dist(k, l) \times speed/10000$, $speed$ a constant that have been fixed to 5 for 5km/h the mean speed of forklifts in a warehouse.
- The incoming trucks are 12 wheels; there are at most 8 of them and each one can transport a maximum of 40 pallets.
- The outcoming trucks are 6 wheels; there are at most 20 of then and each one can transport a maximum of 16 pallets.
- The time windows of trucks' availability is generated to follow a Normal distribution and to be between 8 a.m and 7 p.m. The incoming trucks arrives at 11 a.m more or less 3 hours. The outcomming trucks arrives at 1 p.m more or less 3 hours. Figure 6 illustrates a previsonnal planning.
- If any pallets have to be temporary stored in the crossdock, they remain located on the area near the dock where the incoming truck has delivered these pallets.
- The Cross-docks' capacity is set up to 100.

A series of 10 "bi-partite" instances have been generated.

Note that in an operational solution, an incoming truck may leave the warehouse with pallets, if they cannot be transhipped through the crossdock

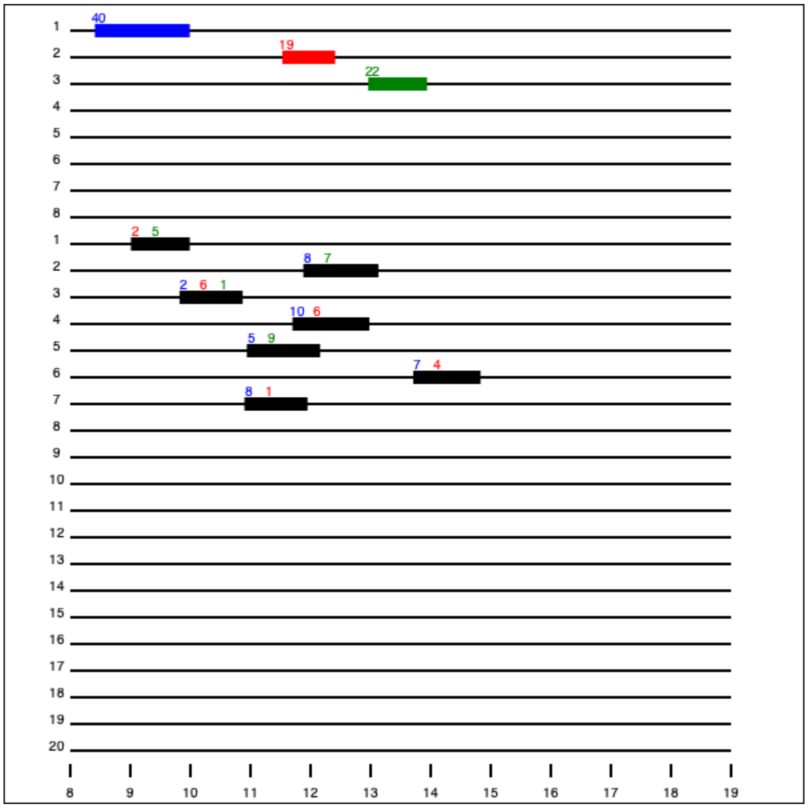


Figure 6: Example of a planning. Timewindows for incoming trucks are on the top, where a color represents a given furnished, and the number indicates the number of pallets to deliver. Timewindows for outcoming trucks are on the bottom colored in black, where the colored number represents the number of pallets to collect from a given furnisher.

to the corresponding outcoming trucks. The capacity problem discussed in Section 5.1.2 does not exist in this configuration.

7.2. Experiments

The goal of the experiments is to observe and understand the optimal solutions outputted by formulations.

7.2.1. Experimental environment

All the algorithms are implemented in Julia (version 1.10) and uses JuMP (version 1.21) as algebraic modeling language. The codes are available online at `https://github.com/xgandibleux/TDAP`. 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.

Gurobi Optimizer (version 10.0.3 build v10.0.3rc0 (mac64[x86])) is used as MIP solver[2].

7.2.2. Experiment 1: detailed numerical analysis of formulations M and G on a didactic instance

This experiment examines in detail the optimal solutions obtained by the M and G formulations on a didactic instance "full mesh". This instance is composed of 5 trucks and 3 docks. The maximum capacity of the terminal is 813 pallets. The previsional arrival and departure time of trucks into the format (hh.mm), transformed in minutes in [00:00;23:59] is given by Table 1a.

The previsional number of pallets f_{ij} to transfert from truck i to truck j, the penalities p_{ij} (€) when a transfert from i to j is not achieved, the operational times (minutes) t_{kl} from k to l, and the transportation cost (€) c_{kl} from k to l are respectively given by Table 1b, 1c, 1d, and 1e. The Gantt chart depicted by Figure 7 illustrates the previsionnal planning of trucks and transferts of pallets.

The information deduced from the data are (1) δ_{ij}, the time conflicts between trucks to be assigned on a same dock (1 iff $[a_i, d_i] \cap [a_j, d_j]$ is empty, 0 otherwise), and (2) the time markers τ. They are reported in Tables 2 and 3 respectively.

An optimal solution obtained respectively with formulation M and formulation G is given in Table 4, where Table 4a given the assigments of trucks

[2]The codes provided online are ready to use GLPK as MIP solver.

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
i	1	.	.	.	.	33
	2	.	.	36	.	.
	3	.	.	.	8	50
	4	.	.	8	.	52
	5	.	.	24	.	.

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

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

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

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

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

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

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

Table 1: Data describing the didactic instance.

to docks and Table 4b reports the transfert 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 transferts awaited (7 transferts) are fulfilled (ratio of 100.0%), thus the penalty cost is equal to 0. With the 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 transferts on the 7 awaited are fulfilled (ratio of 85.71%). For the two optimal solutions collected, Figures 8a and 8b 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

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

Table 2: δ_{ij}, time conflicts between trucks i and j to be assigned on a same dock (1 if no conflict, 0 otherwise).

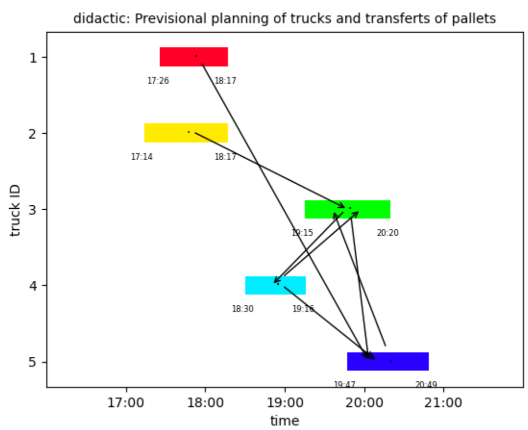
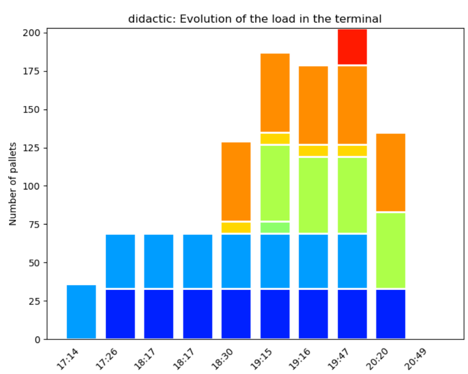


Figure 7: Previsionnal planning of trucks and transferts of pallets.

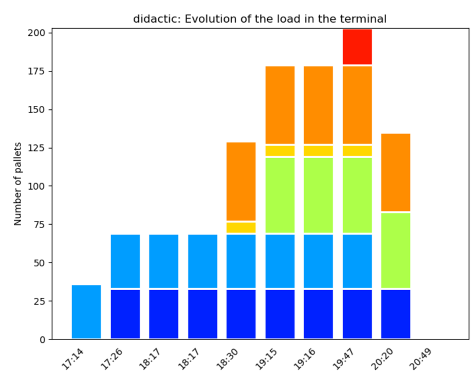
didactic instance shows that an optimal solution obtained with formulation G may be dominated than an optimal solution obtained with M.

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]

Table 3: Arrivals and departures of trucks at a time marker. For example, at the 2nd ($r = 2$) time marker (τ_2 = 17:26), trucks 1 and 2 are arrived (`atr[2]=[1,2]`), none departure of truck (`dtr[2]=[]`).



(a) Result for formulation M.



(b) Result for formulation G.

Figure 8: Evolution of the load in the terminal in term of pallets.

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) Assigment trucks to docks.

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

(b) Transfert of pallets.

Table 4: Optimal solution collected with each formulation.

7.2.3. Experiment 2: comparison between formulations M and G

This experiment focusses on the optimal solutions obtained by the M and G formulations on small and medium size instances “full mesh”. In particular, we want to observe the pallets transferred between trucks. For this experiment, a computation time limit of 600 seconds is given to Gurobi[3]. This value was chosen because it has shown to be sufficient to observe the behaviour of the two formulations given our computer configuration.

65 instances of increasing size were selected from the available dataset with the aim of being optimally solved by Gurobi. For both formulations and for each instance, the results collected, i.e. (1) the optimal objective value (aggregated cost, operational cost, penalty cost), (2) the CPUtime in seconds, (3) the transferts of pallets between docks (number and percentage) achieved, and (4) the number of trucks assigned to a dock are gathered together in Tables B.6 and B.7, and Figures 9 and 10 synthesise graphically these results.

Hormis pour deux instances de petites tailles, la figure 9a montre que la formulation M requière plus de temps de resolution au solveur MIP que la formulation G. L’écart entre les deux augmente significativement lorsque les tailles d’instance augmentent, notamment lorsque le nombre de docks à considérer augmente. Le temps de resolution d’une instance sous la formulation M avec Gurobi atteint la limite de temps de 600 secondes bien avant la formulation G. Sous l’angle des temps de resolution, la formulation G apparait donc meilleure que la formulation M. Here, It is worth to remind that results are presented in (Gelareh et al., 2015, 2016) to tighten the formulation G, letting a potential advantage to the latter formulation on this criterion.

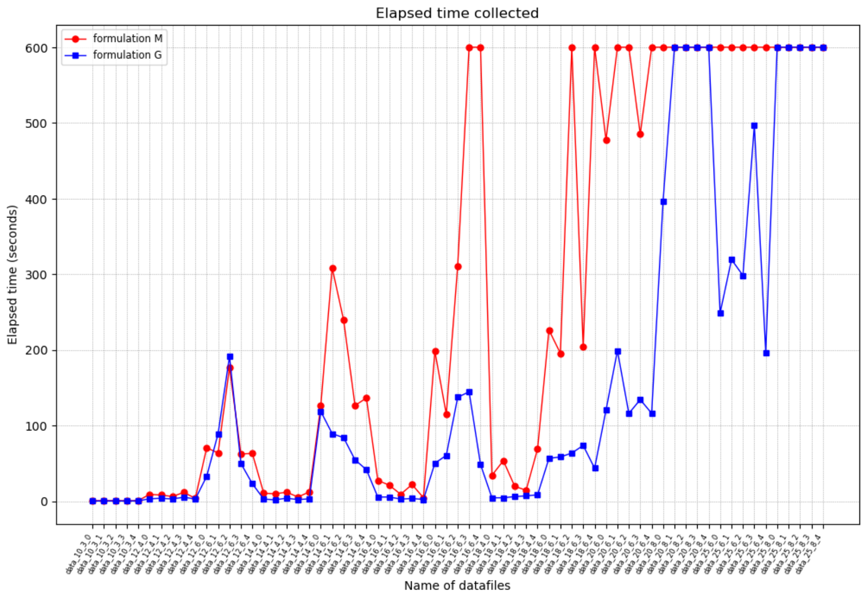
Comme le rapporte la figure 9b, la valeur optimale de la fonction économique obtenue avec la formulation G est inférieure a la valeur obtenue avec la formulation M, hormis pour six instances. La faiblesse pointée par Gelareh et al. (2015) dans la formulation M et la correction proposée avec la formulation G apparait justifiée numériquement mais sans toutefois dominer sur toutes les instances. En regardant de plus près les valeurs dans les tables B.6 et B.7, c’est la partie cout opérationnel qui est plus élevée dans les solutions optimales obtenues avec la formulation M. Par contre le cout de pénalité dans les solutions optimales apparait equivalent dans les deux formulations, avec une

[3]Miao et al. (2009) have reported the difficulty for finding optimal solutions for medium-sized instances within a set time limit of 7200 seconds using CPLEX.

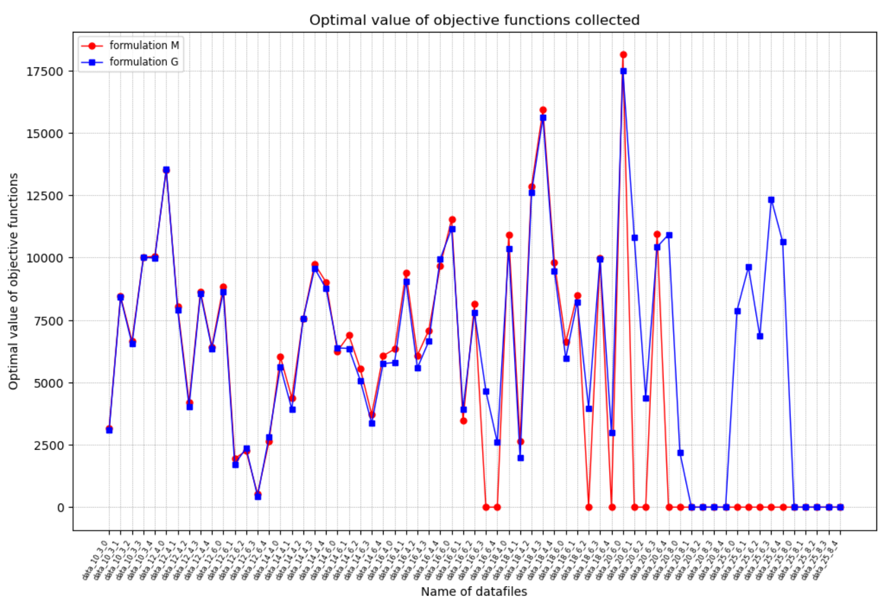
valeur légèrement à l'avantage de la formulation M pour 16 instances. La formulation M élabore donc des solutions optimales avec un meilleur nombre de transfert, ce que illustrent les figures 10a et 10b qui rapportent le nombre de transferts de palettes réalisés dans les solutions optimales pour les deux formulations. Ceci invalide numériquement l'affirmation qui avance que la formulation G permet d'effectuer davantage de transferts de palettes que la formulation M. En effet, tres souvent les deux formulations donnent le même nombre de transfert comme le montre la figure 10b; et quand ce n'est pas le cas, la formulation M rapporte un nombre de transfert tres légèrement supérieur.

Vu d'un exploitant de terminal de cross-docking, cette nuance peut s'avérer sensible. En effet, il apparait raisonnable de souhaiter optimiser en première priorité le nombre de palettes transférées, et optimiser en seconde priorité l'utilisation du terminal en réduisant le temps passé par les palettes dans le terminal. Une utilisation optimisée du terminal offre la possibilité d'anticiper le départ des camions des docks.

Ces observations faites sur les résultats numériques issus des formulations M et G justifient la transition à l'elaboration de solutions optimales à l'aide de la formulation bi-objectif 2R proposée qui permet notamment de rapporter des solutions lexicographiquement optimales.

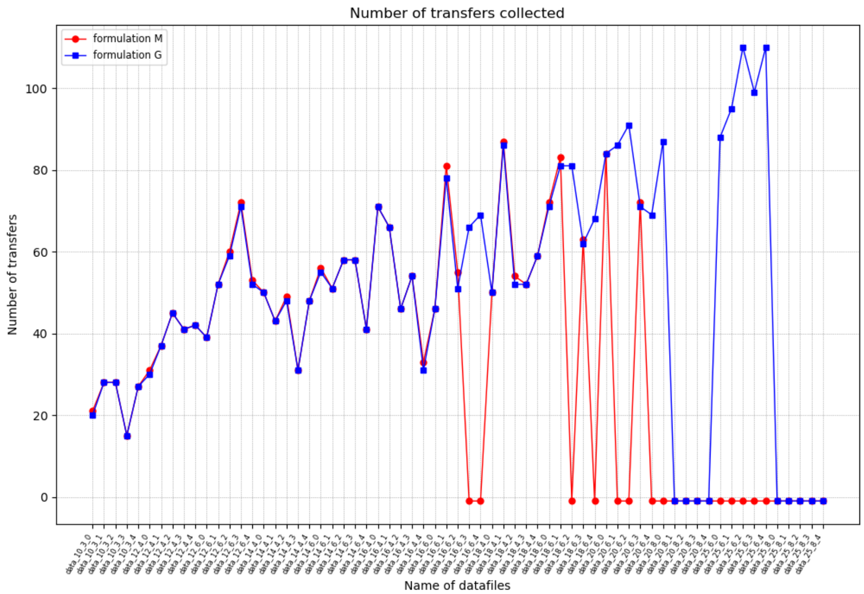


(a) Comparison elapsed time collected for each instance. When a y value is equal to 600, the corresponding optimal solution is not available.

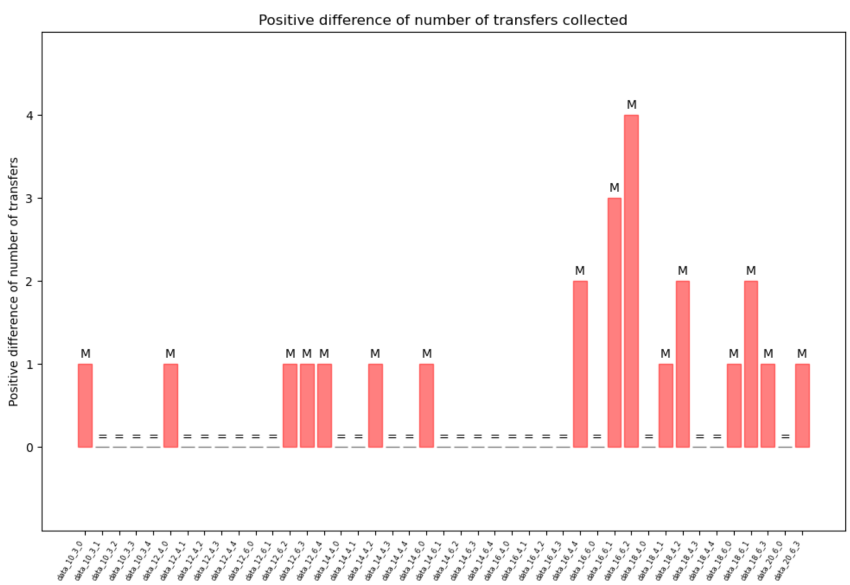


(b) Comparison of objective values collected for each instance. When a y value is equal to 0, the corresponding optimal solution is not available.

Figure 9: 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 10: Comparison between formulations M and G on the number of transferts.

7.2.4. Experiment 3: results obtained with formulation 2R

Formulations M and G from one side and 2R from the other side handle different hypotheses (macroscopic view of pallet's flow plus aggregated single objective for M and G vs microscopic view of pallet's flow plus multiple objective without artifact costs for 2R).

Expe a refaire, écriture à poursuivre

8. Conclusion and perspectives

Ce papier se présente avant tout au lecteur comme un état de l'art synthétisant les différents formulations qui sont rencontrées dans la littérature sur le TDAP depuis 2009. Les formulations, ici nommées M et G, sont décrites avec un cadre de notations unifiées, analysées et discutées. Notre revue de la littérature mettra en évidence un certain nombre de points critiques/écueils relevés dans différentes papiers, engendrant notamment des publications de papiers ayant pour objectif d'apporter une mise au point. Alors que les auteurs concernés présentent un argumentaire et un fond technique solides, nous avons relevé différentes erreurs dans ces documents scientifiques publiés ou diffusés, pour lesquelles nous apportons présentement des corrections. Ainsi notre premier volet de contribution à pour objectif de clarifier de manière rigoureuse les formulations des modèles IP décrivant le TDAP.

Notre second volet de contribution se décline en trois actes. D'abord, les discussions menée ont permis de mettre en lumière un certain nombre de points soit limitant, soit particularisant les formulations M et G. Elles nous nous ont conduit à proposer une formulation 2R immédiatement dérivée de celles-ci, laquelle intègre une partie de ces points mis en lumière. Ainsi cette formulation considère notamment deux fonctions objectif indépendamment et est plus fine dans la prise en compte des données au niveau des contraintes proposées. Ensuite, les instances numériques utilisées dans les papiers relatifs aux formulations M et G questionnent également. En réponse à cela, une nouvelle famille d'instance numérique a été proposé. Les fichiers de données et le générateur permettant de produire ces données sont disponibles en ligne en Open source à `https://github.com/xgandibleux/TDAP`. Enfin, des expérimentations numériques sont rapportées. Toutes les solutions produites par les trois modèles ont été obtenues avec un certificat d'optimalité. Les méthodes et techniques de résolution utilisées pour calculer les solutions sont génériques. Implémentées en Julia et utilisant le langage de modélisation algébrique JuMP, la résolution des instances numériques présentée en entrée de l'une des trois formulations est obtenue avec le solveur GUROBI. Le package MOA qui propose une collection d'algorithmes multi-objectif est utilisé pour produire les solutions pour la formulation 2R.

Les expérimentations numériques ....

*** perspectives ***

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

Appendix A.1. The original formulation

The decision variables are identical to the formulation M (see Section 3). The IP model is thus formulated as (reproduced exactly as presented in papers, with errors and missing information):

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

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

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

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

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

$$z_{ijkk} \le \hat{x}_{ij} \qquad \forall i, j \neq i, k, l \tag{6}$$

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

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

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

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

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

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

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

where:

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

(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 intersect (i.e. i leaves no later than j arrives);

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

and the others parts of the formulation (i.e. (1-4), (7), and (13)) share the same definition with the formulation M.

Appendix A.2. The corrections proposed

A number of inaccuracies and incompleteness were identified in formulation G, making this formulation non-operational. Because of this we were unable to reproduce the results announced by the authors. 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:

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

– in (6): wrong definition of index:

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

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

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

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

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

Taken individually each of these points can be corrected quite easily once identified. However, some are not immediately identifiable, as in the case of the point relating to constraint (5), which could only be identified after a meticulous analysis of the flat formulation, which necessitated testing instances by hand.

Appendix B. Formulations implemented in Julia with JuMP

Appendix 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
         )

# --- (1): 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( (sum( p[i,j] * f[i,j] * ( 1 - sum( z[i,j,k,l] for k=1:m, l=1:m) ) for j=1:n) ) for i=1:n)
          )

# --- (2-8): constraints --------------------------------------------------
@constraint( mod,
             cst2_[i=1:n],
             sum(y[i,k] for k=1:m) <= 1
           )
@constraint( mod,
             cst3_[i=1:n, j=1:n, k=1:m, l=1:m],
             z[i,j,k,l] <= y[i,k]
           )
@constraint( mod,
             cst4_[i=1:n, j=1:n, k=1:m, l=1:m],
             z[i,j,k,l] <= y[j,l]
           )
@constraint( mod,
             cst5_[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,
             cst6_[i=1:n, j=1:n, k=1:m; i!=j],
             x[i,j] + x[j,i] >= z[i,j,k,k]
           )
@constraint( mod,
             cst7_[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,
             cst8_[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
           )
```

Appendix 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
         )

# --- (1): 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( (sum( p[i,j] * f[i,j] * ( 1 - sum( z[i,j,k,l] for k=1:m, l=1:m) ) for j=1:n) ) for i=1:n)
          )

# --- (2-8): constraints ------------------------------------------------------
@constraint( mod,
             cst2_[i=1:n],
             sum(y[i,k] for k=1:m) <= 1
           )
@constraint( mod,
             cst3_[i=1:n, j=1:n, k=1:m, l=1:m],
             z[i,j,k,l] <= y[i,k]
           )
@constraint( mod,
             cst4_[i=1:n, j=1:n, k=1:m, l=1:m],
             z[i,j,k,l] <= y[j,l]
           )
@constraint( mod,
             cst5_[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,
             cst6_[i=1:n, j=1:n, k=1:m; i!=j],
             z[i,j,k,k] <= x[i,j]
           )
@constraint( mod,
             cst7_[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,
             cst8_[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
           )
```

Appendix B.3. Formulation 2R

Code

• *The crossdock.*
A real warehouse has been choosen as reference. Built on a surface of 42×20 meters, the crossdock has a layout of "I" and presents 14 docks.

The distances between docks is based on the Manhattan distance between two docks. Measures are deduced from the physical configuration of the warehouse.

The crossdock has no fix equipment inside. It is viewed as an empty building where the pallets are transfered by forklifts.

If any pallets have to be temporary stored, they remain located on the area near the dock where the incoming truck has delivered these pallets.

• *Maximal number of pallets in a truck.*
A standard pallet has a surface of $120cm \times 80cm$ in general. The maximum number of pallets transported by a truck depends of the type of truck. According the specialised website[4], the number of transported pallets is given in Table B.5.

Number of wheels of the truck	Number of standard pallets
6 wheels (three-axle truck)	14 to 16 standard pallets
10 wheels (five-axle truck)	24 to 30 standard pallets
12-wheel truck (six-axle truck)	36 to 40 standard pallets
16 wheels	48 to 56 standard pallets

Table B.5: Number of pallets transported by a truck.

• *Loading/unloading a truck.*
In average, 1 minute is required to loading/unloading a pallet in a truck with a forklift. Also 15 minutes are legaly required in France for completing various operations (coupling the truck at the dock, fulfill the administrative procedure, and the control of cargo).

• *The forklifts.*
Inside a warehouse or a factory, the French reglementation limits the forklift speed to maximum 10km/h[5]. In practice, the speed is much more low for

[4] `https://www.transports64.fr/combien-de-palette-dans-un-camion`

[5] `https://www.boplan.com/fr/les-reglementations-circulation-des-chariots-elevateurs`

security reasons and to protect the cargo during the transfer of pallets.

Knowing now the distance between docks and the speed of a forklift, a time t_{kl} required for transfering one pallet between two docks is computed.

• *Incoming/outcoming trucks.*
We call "incoming trucks", those who carry pallets issued from furnishers (factories, other warehouses, etc.), and outcoming trucks, those who carry pallets to customers (shops, other warehouses, etc.).

We consider the capacity of incoming trucks larger than the outcoming ones. Also incoming trucks do not receive pallets from any other trucks. Only outcoming trucks receive pallets. Thus the transfer of pallets can be seen following a bi-partite graph (see Figure 5), giving the name of this new collection.

Note that an incoming truck may leave the warehouse with pallets, if they cannot be transhipped through the crossdock to the corresponding outcoming trucks. The capacity problem discussed in Section 5.1.2 does not exist in this configuration.

Appendix B.5. Results for formulations M and G

fname	tElapsed sec	zOpt	zOptCost	zOptPenalty	nTransfertDone #	nTruckAssigned #	pTransfertDone %
data_10_3_0	0.53	3163	160	3003	21	9	67.74
data_10_3_1	0.575	8454	156	8298	28	7	46.67
data_10_3_2	0.691	6659	234	6425	28	8	58.33
data_10_3_3	0.403	10021	60	9961	15	6	31.25
data_10_3_4	0.723	10047	192	9855	27	7	45.76
data_12_4_0	8.994	13508	196	13312	31	8	41.89
data_12_4_1	8.036	8035	220	7815	37	9	56.06
data_12_4_2	6.188	4215	344	3871	45	10	76.27
data_12_4_3	11.606	8642	208	8434	41	9	59.42
data_12_4_4	3.697	6429	198	6231	42	9	66.67
data_12_6_0	70.511	8845	406	8439	39	9	57.35
data_12_6_1	63.788	1954	408	1546	52	11	89.66
data_12_6_2	176.614	2251	532	1719	60	11	89.55
data_12_6_3	62.408	528	528	0	72	12	100.0
data_12_6_4	63.126	2625	422	2203	53	11	88.33
data_14_4_0	10.426	6023	684	5339	50	11	75.76
data_14_4_1	9.734	4376	600	3776	43	12	78.18
data_14_4_2	11.841	7568	472	7096	49	11	62.82
data_14_4_3	5.555	9744	260	9484	31	10	50.0
data_14_4_4	12.22	9015	420	8595	48	11	60.0
data_14_6_0	125.873	6254	760	5494	56	12	73.68
data_14_6_1	308.347	6889	778	6111	51	12	72.86
data_14_6_2	239.314	5540	796	4744	58	12	77.33
data_14_6_3	126.419	3713	568	3145	58	12	81.69
data_14_6_4	136.334	6067	468	5599	41	11	66.13
data_16_4_0	27.428	6334	800	5534	71	14	80.68
data_16_4_1	21.113	9400	560	8840	66	13	70.21
data_16_4_2	9.089	6083	680	5403	46	13	71.88
data_16_4_3	21.952	7090	606	6484	54	13	66.67
data_16_4_4	4.513	9664	286	9378	33	10	44.59
data_16_6_0	198.037	11542	556	10986	46	12	54.76
data_16_6_1	115.055	3485	806	2679	81	14	90.0
data_16_6_2	310.591	8146	528	7618	55	12	69.62
data_16_6_3	600.0	.	.	.	.	.	.
data_16_6_4	600.0	.	.	.	.	.	.
data_18_4_0	34.08	10910	656	10254	50	14	56.82
data_18_4_1	53.474	2653	1352	1301	87	17	94.57
data_18_4_2	19.571	12854	368	12486	54	13	56.25
data_18_4_3	14.447	15922	522	15400	52	12	50.0
data_18_4_4	68.957	9799	486	9313	59	14	60.2
data_18_6_0	226.138	6615	1148	5467	72	15	76.6
data_18_6_1	195.451	8506	842	7664	83	15	75.45
data_18_6_2	600.0	.	.	.	.	.	.
data_18_6_3	204.135	9988	662	9326	63	14	64.95
data_18_6_4	600.0	.	.	.	.	.	.
data_20_6_0	477.613	18159	976	17183	84	16	61.31
data_20_6_1	600.0	.	.	.	.	.	.
data_20_6_2	600.0	.	.	.	.	.	.
data_20_6_3	484.95	10963	942	10021	72	16	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 B.6: Numerical results for formulation M

fname	tElapsed sec	zOpt	zOptCost	zOptPenalty	nTransfertDone #	nTruckAssigned #	pTransfertDone %
data_10_3_0	0.198	3105	38	3067	20	9	64.52
data_10_3_1	0.347	8410	112	8298	28	7	46.67
data_10_3_2	0.263	6545	120	6425	28	8	58.33
data_10_3_3	0.245	10005	44	9961	15	6	31.25
data_10_3_4	0.381	9985	130	9855	27	7	45.76
data_12_4_0	3.1	13554	92	13462	30	8	40.54
data_12_4_1	3.804	7911	96	7815	37	9	56.06
data_12_4_2	2.874	4032	161	3871	45	10	76.27
data_12_4_3	5.227	8556	122	8434	41	9	59.42
data_12_4_4	2.764	6353	122	6231	42	9	66.67
data_12_6_0	32.727	8630	191	8439	39	9	57.35
data_12_6_1	89.21	1722	176	1546	52	11	89.66
data_12_6_2	191.381	2366	260	2106	59	11	88.06
data_12_6_3	49.773	440	251	189	71	12	98.61
data_12_6_4	23.037	2809	186	2623	52	11	86.67
data_14_4_0	2.688	5627	288	5339	50	11	75.76
data_14_4_1	1.693	3932	156	3776	43	12	78.18
data_14_4_2	3.977	7560	192	7368	48	11	61.54
data_14_4_3	1.954	9568	84	9484	31	10	50.0
data_14_4_4	3.066	8762	167	8595	48	11	60.0
data_14_6_0	118.641	6380	292	6088	55	12	72.37
data_14_6_1	89.199	6363	252	6111	51	12	72.86
data_14_6_2	83.93	5076	332	4744	58	12	77.33
data_14_6_3	54.634	3361	216	3145	58	12	81.69
data_14_6_4	41.577	5756	157	5599	41	11	66.13
data_16_4_0	5.413	5793	259	5534	71	14	80.68
data_16_4_1	5.449	9050	210	8840	66	13	70.21
data_16_4_2	2.752	5579	176	5403	46	13	71.88
data_16_4_3	3.484	6649	165	6484	54	13	66.67
data_16_4_4	2.247	9951	87	9864	31	10	41.89
data_16_6_0	50.119	11162	176	10986	46	12	54.76
data_16_6_1	60.471	3915	293	3622	78	14	86.67
data_16_6_2	137.281	7799	174	7625	51	12	64.56
data_16_6_3	144.512	4654	307	4347	66	14	79.52
data_16_6_4	48.524	2597	324	2273	69	15	88.46
data_18_4_0	4.448	10376	122	10254	50	14	56.82
data_18_4_1	4.462	1987	358	1629	86	17	93.48
data_18_4_2	6.004	12626	111	12515	52	13	54.17
data_18_4_3	7.01	15611	211	15400	52	12	50.0
data_18_4_4	8.389	9467	154	9313	59	14	60.2
data_18_6_0	56.549	5972	321	5651	71	15	75.53
data_18_6_1	58.384	8231	335	7896	81	15	73.64
data_18_6_2	63.577	3965	316	3649	81	16	86.17
data_18_6_3	73.592	9956	220	9736	62	14	63.92
data_18_6_4	43.497	2979	238	2741	68	16	87.18
data_20_6_0	120.286	17504	321	17183	84	16	61.31
data_20_6_1	198.974	10825	349	10476	86	16	69.92
data_20_6_2	115.921	4365	304	4061	91	17	88.35
data_20_6_3	134.17	10432	235	10197	71	16	67.62
data_20_6_4	116.109	10924	271	10653	69	16	65.09
data_20_8_0	396.362	2176	353	1823	87	19	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	248.723	7881	301	7580	88	21	75.21
data_25_6_1	319.343	9627	267	9360	95	22	76.0
data_25_6_2	298.268	6853	477	6376	110	22	82.09
data_25_6_3	496.524	12345	456	11889	99	20	68.75
data_25_6_4	196.427	10633	510	10123	110	20	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 B.7: Numerical results for formulation G

References

Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B., 2017. Julia: A fresh approach to numerical computing. SIAM Review 59, 65–98. URL: https://doi.org/10.1137/141000671, doi:10.1137/141000671, arXiv:https://doi.org/10.1137/141000671.

Daquin, C., Allaoui, H., Goncalves, G., Hsu, T., 2021. Variable neighborhood search based algorithms for crossdock truck assignment. RAIRO-Oper. Res. 55, S2291–S2323. URL: https://doi.org/10.1051/ro/2020087, doi:10.1051/ro/2020087.

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

Gelareh, S., Glover, F., Guemri, O., Hanafi, S., Nduwayo, P., Todosijević, R., 2020. A comparative study of formulations for a cross-dock door assignment problem. Omega 91, 102015. URL: https://www.sciencedirect.com/science/article/pii/S0305048317311908, doi:https://doi.org/10.1016/j.omega.2018.12.004.

Gelareh, S., Goncalves, G., Monemi, R.N., 2015. On truck dock assignment problem with operational time constraint within cross docks. Optimization Online. 18 pages. https://optimization-online.org/2014/11/4626/.

Gelareh, S., Monemi, R.N., Semet, F., Goncalves, G., 2016. A branch-and-cut algorithm for the truck dock assignment problem with operational time constraints. European Journal of Operational Research 249, 1144–1152. URL: https://www.sciencedirect.com/science/article/pii/S0377221715008917, doi:https://doi.org/10.1016/j.ejor.2015.09.049.

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

Kucukoglu, I., Ozturk, N., 2017. A mathematical model for truck-door assignment and product placement problem in cross-docking center. Global

Journal of Business, Economics and Management: Current Issues 7, 135–142.

Ladier, A.L., 2014. Planification des opérations de cross-docking : prise en compte des incertitudes opérationnelles et de la capacité des ressources internes. Ph.D. thesis. Grenoble-INP (France).

Lim, A., Ma, H., Miao, Z., 2006. Truck dock assignment problem with time windows and capacity constraint in transshipment network through cross-docks, in: Gavrilova, M., Gervasi, O., Kumar, V., Tan, C.J.K., Taniar, D., Laganá, A., Mun, Y., Choo, H. (Eds.), Computational Science and Its Applications - ICCSA 2006, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 688–697.

Lim, A., Miao, Z., Rodrigues, B., Xu, Z., 2005. Transshipment through crossdocks with inventory and time windows. Naval Research Logistics (NRL) 52, 724–733. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.20113, doi:https://doi.org/10.1002/nav.20113, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nav.20113.

Lubin, M., Dowson, O., Garcia, J.D., Huchette, J., Legat, B., Vielma, J.P., 2023. Jump 1.0: recent improvements to a modeling language for mathematical optimization. Mathematical Programming Computation 15, 581–589. URL: https://doi.org/10.1007/s12532-023-00239-3, doi:10.1007/s12532-023-00239-3.

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

Miao, Z., Lim, A., Ma, H., 2009. Truck dock assignment problem with operational time constraint within crossdocks. European Journal of Operational Research 192, 105–115. URL: https://www.sciencedirect.com/science/article/pii/S0377221707009393, doi:https://doi.org/10.1016/j.ejor.2007.09.031.

Nassief, W., 2017. Cross-dock door assignments: models, algorithms & extensions. Ph.D. thesis. Concordia University (Canada).

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

Tsui, L.Y., Chang, C.H., 1992. An optimal solution to a dock door assignment problem. Computers & Industrial Engineering 23, 283–286. URL: https://www.sciencedirect.com/science/article/pii/0360835292901173, doi:https://doi.org/10.1016/0360-8352(92)90117-3.

Van Belle, J., Valckenaers, P., Cattrysse, D., 2012. Cross-docking: State of the art. Omega 40, 827–846. URL: https://www.sciencedirect.com/science/article/pii/S0305048312000060, doi:https://doi.org/10.1016/j.omega.2012.01.005. special Issue on Forecasting in Management Science.

Zhang, L., 2016. Optimization and simulation of a cross-docking terminal. Ph.D. thesis. Centrale Lille (France).

Zhu, Y.R., 2007. Recent Advances and Challenges in Quadratic Assignment and Related Problems. Ph.D. thesis. University of Pennsylvania (USA).

Zhu, Y.R., Hahn, P.M., Liu, Y., Guignard-Spielberg, M., 2009. New approach for the cross-dock door assignment problem, in: XLI SBPO 2009 - Pesquisa Operacional na Gestão do Conhecimento, pp. 1226–1236.