Integrating truck scheduling and employee rostering in a cross-docking platform – an iterative approach

(presented at the 6^{th} IESM Conference, October 2015, Seville, Spain)

Anne-Laure Ladier
Univ. Grenoble Alpes, G-SCOP, F-38000 Grenoble, France
CNRS, G-SCOP, F-38000 Grenoble, France
Email: anne-laure.ladier@g-scop.fr

Gülgün Alpan
Univ. Grenoble Alpes, G-SCOP, F-38000 Grenoble, France
CNRS, G-SCOP, F-38000 Grenoble, France
Email: gulgun.alpan@g-scop.fr

***Abstract*—In a cross-docking platform, goods are unloaded, transferred and reloaded into trucks with little or no storage in between. The crossdock truck scheduling problem addresses the hard problem of coordinating the truck operations. However, crossdock operations are mostly done manually: it is therefore important to take staffing issues into account while building the truck schedule. This article shows how a truck scheduling model and an employee timetabling and rostering model can be combined to address both problems in an integrated manner. Three approaches are compared. The sequential approach consists in sequentially solving the two problems: from the truck schedule calculated first, a workload is deduced and used as input for the employee timetabling and rostering process. The iterative approach consists in solving both problems one after another in an iterative manner until a stable point is reached. Two iterative procedures are proposed, employees-first and trucks-first.**

I. Introduction

Cross-docking is a technique used in logistic platforms (also called *crossdocks*) to accelerate the flow of goods while minimizing storage. Each truck arriving at the platform (*inbound truck*) contains products aimed at different destinations or clients. The cross-docking process consists in unloading the products from the inbound truck, sorting them by clients, then transferring and loading them to *outbound trucks* – each outbound truck being related to a specific client.

To operate properly, a cross-docking platform requires a very good coordination between inbound and outbound trucks. The *crossdock truck scheduling problem* handles this problem and also makes decisions on the transfer of goods inside the platform.

Goods can be moved inside the cross-docking platform either manually, with an automated system (*e. g.* conveyor belts) or with a combination of both. Automation can also be used for storage (automated storage and retrieval systems) – see *e. g.* Baker and Halim [1] or Granlund [2]. Note that these systems support human's work but do not replace it. In general, automated systems represent heavy investments, but are feared to be not flexible enough to meet changing market requirements [1]. Therefore, automation is generally adopted by companies dealing with a limited range of product types, in a stable or growing market (*e. g.* postal and parcel services). For logistic service providers, whose survival depends on their flexibility, the operations stay mainly manual. Manpower is therefore the first cost center in logistics and especially for logistics providers (see Graham [3] and van den Berg [4]).

It is thus crucial to stick to the activity volume when dimensioning the task force; yet this activity volume depends directly on the truck schedule. Platform managers handle this issue by creating a truck schedule first, and then creating an employee timetable and an employee roster[1] in order to cover the resulting workload. The underlying idea is to organize first the operations involving external stakeholders (the transportation providers operating the trucks), and to organize the internal matters as a second step.

This sequential approach, however, might not be the best way to solve this problem, since truck scheduling and employee rostering are strongly intertwined. As noted by Van Belle et al. [6], "the scheduling of the trucks heavily influences the workload for the internal resources". Taking staffing issues into account when creating the truck schedule might therefore lead to better solutions. Indeed, as noted by Maravelias and Sung [7]:

> "To achieve globally optimal solutions, the interdependencies between the different planning functions should be taken into account, and planning decisions should be made simultaneously. In other words, planning problems should be integrated".

However, the crossdock truck scheduling problem and the employee rostering problem are both complex. In cross-docking literature, resource constraints are not often taken into account (Ladier [8]), let alone detailed timetabling issues. However, the crossdock truck scheduling problem, that consists of scheduling the trucks only, is covered by many authors listed in the state-of-the art established by Van Belle et al. [6] – see *e. g.* Fazel Zarandi et al. [9]. On the other hand, the employee timetabling problem is widely developed for different fields of application (nurse rostering, crew scheduling. . .); but in the logistics field, that requires specific constraints regarding the workload variation and the very diverse employee qualifications, it is only covered by Günther and Nissen [10, 11] and Ladier et al. [12].

Integrating the two problems including all realistic business-oriented constraints would result in a model way too complex to solve. We therefore propose to use an iterative approach instead, that runs the two different models iteratively until a stable point is reached.

[1]Rostering is "the placing, subject to constraints, of resources into slots in a pattern. One may seek to minimize some objective, or simply to obtain a feasible allocation. Often the resources will rotate through a roster". Wren [5]

The rest of the article is organized as follows. A review of the literature regarding the integration of truck scheduling and employee timetabling can be found in section II. The two models that are to be integrated, namely the crossdock truck scheduling model and the employee timetabling and rostering model, are presented in section III. A simple sequential approach that mimic the decision process usually used by platform managers is described in section IV in order to have a comparison reference when evaluating the iterative approaches described in section V; two different iterative strategies (employees-first and trucks-first) are detailed. Numerical experiments are conducted in section VI before giving concluding remarks in section VII.

II. Literature review

The crossdock truck scheduling problem and the employee rostering problem are solved separately in the literature (see details in section III) but the two aspects are rarely integrated. Only Ko et al. [13] integrate "fairness" when solving a truck-to-door assignment problem: their objective is to minimize both the number of workers engaged in loading operation and the imbalance ratio among the workers. They use a genetic algorithm approach with a line balancing heuristic. Li et al. [14] are the only ones who attempted a totally integrated approach: they propose an Excel tool (the exact functioning of which is not really provided) to conduct the operations planning, sequencing, real-time scheduling for container arrivals and pallet transfer, and real-time resource management. Although the detailed models are not given in the article, their approach seems to be based on greedy heuristics.

It is necessary to turn to different fields to find examples of combined operations planning and employee timetabling using exact methods: production planning on the one hand, and vehicle and crew scheduling on the other hand. Artigues et al. [15] give a review of articles dealing with the integration of task and employee scheduling in both application fields. Since the publication of his state-of-the-art in 2007, more recent work has been done on the topic. Artigues et al. [16] use a hybrid branch-and-bound to solve an integrated employee timetabling and job-shop scheduling problem. Working on two comparable problems, Guyon et al. [17, 18] propose to use a Benders decomposition, a specific decomposition with cut generation, and a hybridization of a cut generation process with a branch and bound strategy. In the transportation field, Mercier and Soumis [19] propose an integrated model for aircraft routing, crew scheduling and flight retiming, solved with a Benders decomposition method. Alternatively, Weide et al. [20] propose to solve the two models (aircraft routing and crew scheduling) in an iterative way. Traditionally, the routing problem is solved prior to the crew scheduling problem; but the authors note that this procedure might cause some crews to have a very short amount of time to transfer from one aircraft to another, which is likely to propagate delays. By solving both models in an integrated way, they aim at increasing the overall robustness of the operations.

> "We start with a minimal cost crew pairing solution without taking aircraft routings into account. Then, in each iteration we solve the individual aircraft routing problem first, taking into account the current crew pairing solution. Then, given the aircraft routing solution we resolve the crew pairing problem. We only use the objective functions in both problems to pass information from the problem solved previously to generate more and more robust solutions. [...] We stop the process when the level of robustness cannot be improved any further". Weide et al. [20]

In this article, we propose to apply a procedure comparable to the one used by Weide et al. [20] in order to connect a truck scheduling model and an employee timetabling and rostering model, both of which are detailed in the next section.

III. Presentation of the initial models

The objective of this article is to show how a truck scheduling model and an employee rostering model can be connected in order to create platform schedules of better quality than what the manager would obtain manually with a sequential process. Therefore, instead of writing new models for each sub-problem, we reuse models already present in the literature. This section introduces them both.

A. Crossdock truck scheduling

Ladier and Alpan [21] propose a crossdock truck scheduling model with time windows, in which the transportation providers express in advance their preferences regarding the time at which the trucks arrive at and leave from the platform. The objective is therefore to minimize the quantity of items going through storage instead of transferring directly, and also to minimize the dissatisfaction of the transportation providers regarding the time window each truck is allocated to. Figure 1 provides a quick list and description of the data and variables used in the model; Figure 2 shows the full integer program. For detailed explanations, we refer the reader to Ladier and Alpan [21].

The objective Π_0 is a weighted sum of objectives: Π_0^α and Π_0^β evaluate the dissatisfaction of the transportation providers for the inbound trucks and the outbound trucks, respectively; Π_0^γ is the total number of pallets transiting through storage. The number of trucks docked cannot exceed the number of doors (constraints 4-5); a pallet moves from/to a truck only when the truck is present (constraints 6-7). Constraint (8) models the content of an inbound truck, and constraint (9) the correct loading of an outbound truck. Constraint (11) limits the capacity of the platform (number of pallet transiting through at every time period). Only one time window is assigned per truck (constraint 12). Finally, constraints (13-14) calculate the number of pallets put in storage.

Because the problem is NP-hard in the strong sense and cannot be computed for instances of realistic sizes, Ladier and Alpan [21] propose three different heuristics, that perform differently depending on the instance size and the relative weights of the objective elements. In this article, we will use heuristic H2 as it performs well for any medium-sized instance. This heuristic uses a decomposition of the problem, solving separately the outbound side first before solving a version of the integer program restricted to the inbound side.

Thus in the remaining of the article, when referring to the "truck scheduling model", we mean the integer program

Definition sets:

- $\mathcal{H}$ — planning horizon;
- $\mathcal{I}$ — set of inbound trucks;
- $\mathcal{O}$ — set of outbound trucks;
- $\mathcal{C}$ — set of clients to whom the pallets should be delivered;
- $\mathcal{K}_i$ — set of possible presence slots of the truck $i \in \mathcal{I}$;
- $\mathcal{K}_o$ — set of possible presence slots of the truck $o \in \mathcal{O}$.

Decision variables:

- x_{hio} — amount of pallets transferred from inbound truck $i \in \mathcal{I}$ to outbound truck $o \in \mathcal{O}$ at time period $h \in \mathcal{H}$;
- w^I_{ik} — =1 if slot $k \in \mathcal{K}_i$ is chosen for truck $i \in \mathcal{I}$, 0 otherwise;
- w^O_{ok} — =1 if slot $k \in \mathcal{K}_o$ is chosen for truck $o \in \mathcal{O}$, 0 otherwise;
- s^I_{hic} — number of pallets for client $c \in \mathcal{C}$ going from truck $i \in \mathcal{I}$ to the storage location at time period $h \in \mathcal{H}$;
- s^O_{ho} — number of pallets going from the storage location to truck $o \in \mathcal{O}$ at time period $h \in \mathcal{H}$;
- s_{hc} — number of pallets for client c stored at time period h.

Data:

- Q_{ic} — number of pallets for client $c \in \mathcal{C}$ in truck $i \in \mathcal{I}$;
- Z_{co} — = 1 if truck $o \in \mathcal{O}$ is for client $c \in \mathcal{C}$, 0 otherwise;
- N^I — number of inbound doors;
- N^O — number of outbound doors;
- M — maximum number of pallets that can be moved during one time period inside the platform;
- F — number of pallets to fully load one outbound truck;
- W^I_{ikh} — = 1 if hour $h \in \mathcal{H}$ is in slot $k \in \mathcal{K}_i$ for truck $i \in \mathcal{I}$;
- W^O_{okh} — = 1 if hour $h \in \mathcal{H}$ is in slot $k \in \mathcal{K}_o$ for truck $o \in \mathcal{O}$.

Penalties:

- P^I_{ik} — penalty paid for using slot $k \in \mathcal{K}_i$ for truck $i \in \mathcal{I}$, if k is outside the wished time window expressed by the transportation provider;
- P^O_{ok} — penalty paid for using slot $k \in \mathcal{K}_o$ for truck $o \in \mathcal{O}$.

Source: Ladier and Alpan [21]

Fig. 1: Truck scheduling model data and variables

$$\min \Pi_0 = \alpha \Pi_0^\alpha + \beta \Pi_0^\beta + \gamma \Pi_0^\gamma$$

s.t.

$$\Pi_0^\alpha = \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} P^I_{ik} w^I_{ik} \tag{1}$$

$$\Pi_0^\beta = \sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}_o} P^O_{ok} w^O_{ok} \tag{2}$$

$$\Pi_0^\gamma = \sum_{h \in \mathcal{H}, i \in \mathcal{I}, c \in \mathcal{C}} s^I_{hic} \tag{3}$$

$$\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} W^I_{ikh} w^I_{ik} \le N^I \qquad \forall h \in \mathcal{H} \tag{4}$$

$$\sum_{o \in \mathcal{O}} \sum_{k \in \mathcal{K}_o} W^O_{okh} w^O_{ok} \le N^O \qquad \forall h \in \mathcal{H} \tag{5}$$

$$x_{hio} + \sum_{c \in \mathcal{C}} s^I_{hic} \le F \sum_{k \in \mathcal{K}_i} W^I_{ikh} w^I_{ik} \qquad \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O} \tag{6}$$

$$x_{hio} + s^O_{ho} \le F \sum_{k \in \mathcal{K}_o} W^O_{okh} w^O_{ok} \qquad \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O} \tag{7}$$

$$\sum_{h \in \mathcal{H}, o \in \mathcal{O}} Z_{co} x_{hio} + \sum_{h \in \mathcal{H}} s^I_{hic} = Q_{ic} \qquad \forall i \in \mathcal{I}, c \in \mathcal{C} \tag{8}$$

$$\sum_{i \in \mathcal{I}, h \in \mathcal{H}} x_{hio} + \sum_{h \in \mathcal{H}} s^O_{ho} = F \qquad \forall o \in \mathcal{O} \tag{9}$$

$$\sum_{o \in \mathcal{O}} x_{hio} + \sum_{c \in \mathcal{C}} s^I_{hic} \le M \qquad \forall i \in \mathcal{I}, h \in \mathcal{H} \tag{10}$$

$$\sum_{k \in \mathcal{K}_i} w^I_{ik} = 1 \qquad \forall i \in \mathcal{I} \tag{11}$$

$$\sum_{k \in \mathcal{K}_o} w^O_{ok} = 1 \qquad \forall o \in \mathcal{O} \tag{12}$$

$$s_{hc} = s_{(h-1)c} + \sum_{i \in \mathcal{I}} s^I_{hic} - \sum_{o \in \mathcal{O}} Z_{co} s^O_{ho} \qquad \forall c \in \mathcal{C}, h \in \mathcal{H} \setminus \{0\} \tag{13}$$

$$s_{0c} = \sum_{i \in \mathcal{I}} s^I_{0ic} - \sum_{o \in \mathcal{O}} Z_{co} s^O_{0o} \qquad \forall c \in \mathcal{C} \tag{14}$$

$$x_{hio}, s^I_{hic}, s^O_{ho}, s_{hc} \in \mathbb{N}^+ \qquad \forall h \in \mathcal{H}, i \in \mathcal{I}, o \in \mathcal{O}, c \in \mathcal{C}$$

$$w^I_{ik}, w^O_{ok} \in \{0,1\} \qquad \forall i \in \mathcal{I}, o \in \mathcal{O}, k \in K$$

Source: Ladier and Alpan [21]

Fig. 2: Truck scheduling model

$$\min \quad \Pi_3 = \alpha_3 \Pi_3^\alpha + \beta_3 \Pi_3^\beta$$

s.t.

$$\Pi_3^\alpha = \sum_{e \in \mathcal{E}'', s \in \mathcal{S}} \left| Y''_{es} - y''_{es} \right| \tag{15}$$

$$\Pi_3^\beta = \sum_{e \in \mathcal{E}'', t \in \mathcal{T}, q \in \mathcal{Q}} \left| X''_{etq} - x''_{etq} \right| \tag{16}$$

$$\sum_{e \in \mathcal{E}''} x''_{et_1q} = W^1_{t_1q} \qquad \forall t \in \mathcal{T}^1, q \in \mathcal{Q} \tag{17}$$

$$\sum_{e \in \mathcal{E}'', q \in \mathcal{Q}} S_{t_2q} x''_{et_2q} = W^2_{t_2} \qquad \forall t \in \mathcal{T}^2 \tag{18}$$

$$x''_{etq} \le X_{etq} \qquad \forall e \in \mathcal{E}'', t \in \mathcal{T}, q \in \mathcal{Q} \tag{19}$$

$$x''_{etq} = \sum_{s \in \mathcal{S}} Z_{sq} y''_{es} \qquad \forall e \in \mathcal{E}'', t \in \mathcal{T}, q \in \mathcal{Q} \tag{20}$$

$$\sum_{t \in \mathcal{T}} x''_{etq} \le 1 \qquad \forall e \in \mathcal{E}'', q \in \mathcal{Q} \tag{21}$$

$$\sum_{s \in \mathcal{S}} y''_{es} \le 1 \qquad \forall e \in \mathcal{E}'' \tag{22}$$

$$x''_{etq}, y''_{es} \in \{0,1\} \qquad \forall e \in \mathcal{E}'', t \in \mathcal{T}, q \in \mathcal{Q}$$

$$\Pi_3^\alpha, \Pi_3^\beta, \Pi_3^\gamma, \Pi_3^\delta \in \mathbb{R}^+$$

Source: Ladier et al. [12]

Fig. 3: Employee rostering model

proposed by Ladier and Alpan [21] for small instances, and heuristic H2 (also proposed by Ladier and Alpan [21]) for bigger ones.

B. Employee timetabling and rostering

Constraints for scheduling and rostering are numerous: logistic employees are multi-skilled employees and have flexible working hours or short-term contracts. Legal constraints and handling equipments' capacities should also be met. Ladier et al. [12] propose a model supporting the chain of decisions from weekly timetabling to daily rostering (detailed task allocation). The problem is divided into three sub-problems depending on the type of decision to be made: the *weekly timetabling* step consists in workforce dimensioning and task allocation for a week, and the second step is the detailed *daily rostering*. The three levels of decisions are made sequentially; each is modeled as a mixed integer linear program. Figure 4 gives an overview of the main inputs and outputs of the *weekly timetabling* and *daily rostering* models. Because they are solved sequentially, the outputs of the weekly timetabling step are reused as inputs to build the daily roster. The integer program modeling the daily rostering step is given in Figure 3, because the details of this steps are used in the remaining of this article. For more detailed explanations regarding the model[2], data and assumptions, we refer the reader to Ladier et al. [12].

Constraints (15) and (16) define the penalties given if a shift changes compared to the output of step 1 (Π_3^α), and if a task changes compared to the output of step 1 (Π_3^β). Constraint sets (17) and (18) match the workers to the workload, for the tasks from $\mathcal{T}^1$ defined per hour and for the tasks from $\mathcal{T}^2$ defined by slots, respectively. Constraint set (19) checks that the tasks are allocated to the employees only when it is possible. Finally, constraint sets (21) and (22) ensure that each employee has no more than one shift per day, and one task per interval.

Although each of the three subproblems is NP-hard in the strong sense (Ladier [8]), the numerical experiments carried out by Ladier et al. [12] demonstrate that the mixed integer linear programs for the weekly timetabling and daily rostering can be solved in a fast manner for real size problems.

The models detailed above are closely linked to each other: the platform capacity data M used in the truck scheduling model depends on the employee roster, while the workloads W, W^1, W^2 used in the weekly timetabling and rostering models clearly depend on the truck schedule. Because of the high number of variables and parameters, and the problems complexity, a mathematical model for the integrated problem is hardly usable in an industrial context. Instead, we propose to use a sequential and an iterative approach.

[2]especially the weekly timetabling model, not displayed here for the sake of brevity.

Definition sets:

$\mathcal{E}$ set of regular employees

$\mathcal{E}'$ set of employees including the temporary workers hired for the week

$\mathcal{E}''$ set of employees including the temporary workers hired for the day considered

$\mathcal{T}$ set of tasks, split into two subsets:
$\mathcal{T}^1$, tasks whose workload is defined hour per hour.
$\mathcal{T}^2$, tasks whose workload is defined for a whole slot or time window.

$\mathcal{P}$ set of available temporary workers profiles.

$\mathcal{D}$ set of working days considered for the weekly schedule.

$\mathcal{H}$ set of working hours in a day.

$\mathcal{S}$ set of possible shifts, for example "8AM–4PM".

$\mathcal{Q}$ set of intervals considered for the daily rostering, *e.g.* a quarter of an hour.

Weekly timetabling data:

X'_{edt} Data matrix indicating whether task t can be done by employee $e \in \mathcal{E}$ on day $d \in \mathcal{D}$.

Q_{et} Qualifications, or level of experience of employee $e \in \mathcal{E}$ for task $t \in \mathcal{T}$, defined on $\{0..\zeta\}$ where $\zeta \in \mathbb{N}^*$.

P_{pt} $= 1$ if a worker with profile $p \in \mathcal{P}$ is qualified for task t, 0 otherwise.

C_p Cost of hiring a worker with profile $p \in \mathcal{P}$.

W_{td} Workload for task $t \in \mathcal{T}$ and day $d \in \mathcal{D}$.

W^1_{tdh} Workload for task $t \in \mathcal{T}^1$ defined on a precise time window, hour $h \in \mathcal{H}$ and day $d \in \mathcal{D}$.

W^2_{td} Workload for task $t \in \mathcal{T}^2$ defined on a slot, for day $d \in \mathcal{D}$.

S_{thd} Slot description: $S_{thd} = 1$ if task $t \in \mathcal{T}^2$ can be done on hour $h \in \mathcal{H}$ and day $d \in \mathcal{D}$, 0 otherwise.

Z_{sh} $= 1$ if hour $h \in \mathcal{H}$ is in shift $s \in \mathcal{S}$, 0 otherwise.

D_s length of shift $s \in \mathcal{S}$, in hours.

Weekly timetabling outputs:

x'_{etdh} Percentage of time spent on task $t \in \mathcal{T}$ by employee $e \in \mathcal{E}'$ on day $d \in \mathcal{D}$ and hour $h \in \mathcal{H}$.

y'_{eds} Shift allocation: $y'_{eds} = 1$ if employee $e \in \mathcal{E}'$ is allocated to shift $s \in \mathcal{S}$ on day $d \in \mathcal{D}$.

Daily rostering input data:

X''_{etq} Task allocation as calculated by step 1. $X''_{etq} = 1$ if employee $e \in \mathcal{E}''$ is present on time interval $q \in \mathcal{Q}$, able to do task $t \in \mathcal{T}$, and if task t can be done on interval q; 0 otherwise.

Y''_{es} Shift allocation as calculated by step 1.

W^1_{tq} Workload for task $t \in \mathcal{T}^1$ defined per interval, for time interval $q \in \mathcal{Q}$.

W^2_t Workload of the considered day for the task $t \in \mathcal{T}^2$ defined per slot.

S_{tq} Slot description: $S_{tq} = 1$ if task $t \in \mathcal{T}^2$ can be done during interval $q \in \mathcal{Q}$, 0 otherwise.

Daily rostering outputs:

y''_{es} Shift allocation: $y''_{es} = 1$ if employee $e \in \mathcal{E}''$ is allocated on shift $s \in \mathcal{S}$, 0 otherwise.

x''_{etq} Task allocation: $x''_{etq} = 1$ if employee $e \in \mathcal{E}''$ works on task $t \in \mathcal{T}$ during interval $q \in \mathcal{Q}$, 0 otherwise.

Source: Ladier et al. [12]

Fig. 4: Weekly timetabling/rostering data and variables

IV. Sequential approach

The sequential approach is the "intuitive" one, which could be used by a manager who has at his disposal both a truck scheduling tool and a weekly timetabling and daily rostering tool based on the models described in section III. It is therefore the approach more commonly used in industry. The general idea is to deal first with the external stakeholders (transportation providers) by creating the truck schedule, and then to sort out the internal issues by creating the employee timetable and roster.

The employee timetabling and rostering models need a workload as input, workload which is directly linked to the truck schedule. Yet the truck schedule is difficult to obtain in a cross-docking platform; hence it would be natural to firstly run the truck scheduling model for each day of the week. The workload for the week (W) can then be deduced from the truck schedules (see the detailed procedure below) and used as input to run the weekly step of the timetabling process, using the model proposed by Ladier et al. [12]. The daily roster is created every morning, using the workload deduced from the truck schedule of the day, and the timetable already created for the week. The employee rostering model thus creates a schedule that matches the workload and does not differ too much from the weekly schedule (X'', Y''). The process is summarized in Figure 5.

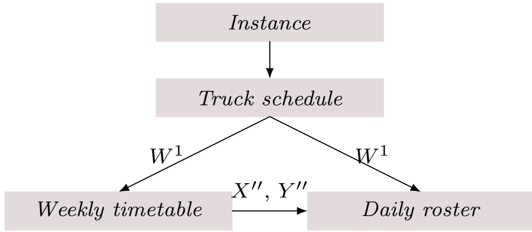


Fig. 5: Principle of the sequential approach

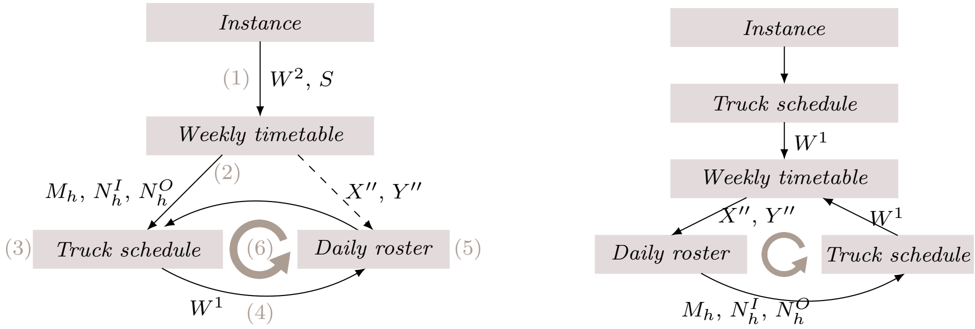
a) Deducing workload W^1 from the truck schedule: Among the outputs of the truck scheduling model are x_{hio}, which gives the number of pallets moving directly from truck i to truck o within time unit h; s^I_{hic} which denotes the moves from truck i to storage at time h (for each client c) and s^O_{ho} which gives the number of pallets transferred from storage to truck o at time h. Using these three outputs, the workload can be expressed precisely, hour by hour: all tasks therefore belong to set $\mathcal{T}^1$. The workload is defined as follows for all $h \in \mathcal{H}$:

$$\begin{array}{lrcl}
\text{Unloading} & W^1_{0dh} & = & ST^0\Big(\sum_{i\in\mathcal{I},o\in\mathcal{O}} x_{hio} + \sum_{i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic}\Big) \\
\text{Control and scan} & W^1_{1dh} & = & ST^1\Big(\sum_{i\in\mathcal{I},o\in\mathcal{O}} x_{hio} + \sum_{i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic}\Big) \\
\text{Direct transfer} & W^1_{2dh} & = & ST^2 \sum_{i\in\mathcal{I},o\in\mathcal{O}} x_{hio} \\
\text{Transfer to stock} & W^1_{3dh} & = & ST^3 \sum_{i\in\mathcal{I},c\in\mathcal{C}} s^I_{hic} \\
\text{Transfer from stock} & W^1_{4dh} & = & ST^4 \sum_{o\in\mathcal{O}} s^O_{ho} \\
\text{Loading} & W^1_{5dh} & = & ST^5\Big(\sum_{o\in\mathcal{O}} s^O_{ho} + \sum_{i\in\mathcal{I},o\in\mathcal{O}} x_{hio}\Big)
\end{array}$$

b) Deriving the daily roster from the weekly timetable: This step is done exactly as described in section III. The decisions taken at the weekly timetabling steps are used as input data (X'', Y'') in the rostering step.

V. Iterative approaches

The sequential approach described in the previous section does not guarantee global optimality. Although the employee timetable and roster match the previously calculated truck schedule, maybe a better solution could be reached if the truck schedule was calculated taking staffing issues into account. We therefore apply an approach similar to the one described by Weide et al. [20] to our problem. The truck schedule and the employee roster are run iteratively until a stable point is reached. Two different cases are studied: starting with the calculation of the employee timetable and roster (*employees-first*) and starting with the truck schedule (*trucks-first*). Both principles are described in Figure 6 and further detailed in the following sections.



(a) Employees first

(b) Trucks first

Fig. 6: Principle of the iterative approaches

A. Employees-first

This solution considers the timescale of the different decisions to be made and therefore calculates first the employees weekly timetabling; the working days and the starting and ending time of each employee on each day is communicated to the employees one week in advance. In the following we detail the steps to follow in the employee-first procedure. The numbers in parenthesis help relating the different steps to Figure 6.

a) (1) Deduce workload W^2 from an instance: A difficulty of this approach is that the employee timetable has to be calculated before the actual truck schedule is known, since the truck scheduling model has not yet been run at this stage. Hence, the workload has to be estimated. The proposed solution is to define all tasks as defined by slots, *i. e.* all tasks belong to set $\mathcal{T}^2$. The slots are defined based on the wishes of the transportation providers. The timetabling and rostering models take the decision about when to carry out the different tasks, within the predefined slots. In order to quantify the workload regarding storage, an estimation τ^{stock} is given as the proportion of pallets which usually go to stock – for instance, based on historical data. More precisely, workload W^2 and slots S are defined for day d as follows – we denote by $\mathcal{H}^I$ (resp. $\mathcal{H}^O$) the set of hours that are in the preferred time window of at least one inbound (resp. outbound) truck:

Unloading	$W^2_{0d} = \; |\mathcal{I}| \times ST^0$
	$S_{0hd} = \; 1$ if $h \in \mathcal{H}^I$, 0 otherwise;
Control and scan	$W^2_{1d} = \; |\mathcal{I}| \times ST^1$
	$S_{1hd} = \; 1$ if $h \in \mathcal{H}^I$, 0 otherwise;
Direct transfer	$W^2_{2d} = \; (1-\tau^{\text{stock}}) \times ST^2$
	$S_{2hd} = \; 1$ if $h \in \mathcal{H}^I$, 0 otherwise;
Transfer to stock	$W^2_{3d} = \; \tau^{\text{stock}} \times ST^3$
	$S_{3hd} = \; 1$ if $h \in \mathcal{H}^I$, 0 otherwise;
Transfer from stock	$W^2_{4d} = \; \tau^{\text{stock}} \times ST^4$
	$S_{1hd} = \; 1$ if $h_i n \mathcal{H}^O$, 0 otherwise;
Loading	$W^2_{5d} = \; |\mathcal{I}| \times ST^5$
	$S_{5hd} = \; 1$ if $h \in \mathcal{H}^O$, 0 otherwise.

b) (2) Deduce new data M_h, N_h^I, N_h^O from the weekly timetable: The staffing decisions made in the weekly schedule create some constraints for the platform operations, in terms of the number of persons available to carry out the different tasks. Three new data elements are thus calculated from the weekly timetable:

- M_h — maximum number of pallets that can be transferred at time unit $h \in \mathcal{H}$, according to the weekly employee timetable;
- N_h^I — maximum number of pallets that can be unloaded at time unit $h \in \mathcal{H}$, according to the employees' weekly timetable;
- N_h^O — maximum number of pallets that can be loaded at time $h \in \mathcal{H}$, according to the employees' weekly timetable.

The values of M_h, N_h^I and N_h^O are deduced from the allocation of employees to the transfer, unloading and loading tasks ($t = 0$, $t = 2$, $t = 5$). For a given day d, they are calculated from the weekly timetabling output x'_{etdh} as follows:

$$M_h = \sum_{e \in \mathcal{E}} \frac{x'_{e2dh}}{ST^2} \qquad \forall h \in \mathcal{H} \qquad \text{(Equation 1)}$$

$$N_h^I = \sum_{e \in \mathcal{E}} \frac{x'_{e0dh}}{ST^0} \qquad \forall h \in \mathcal{H} \qquad \text{(Equation 2)}$$

$$N_h^O = \sum_{e \in \mathcal{E}} \frac{x'_{e5dh}}{ST^5} \qquad \forall h \in \mathcal{H} \qquad \text{(Equation 3)}$$

M_h is obtained from the allocation of employees to task 2 (direct transfer). Variable x'_{etdh}, which gives a number of persons, is divided by the standard time of the operations (in hour/pallet) to obtain a number of pallet for each hour. Similarly, N^I and N^O are calculated from the allocation to tasks 0 (unloading) and 5 (loading), respectively.

c) (3) Include new data M_h, N_h^I, N_h^O in the truck scheduling model: The truck daily schedule is calculated every day; in order to take into account the new staffing-related information as soft constraints, three new constraints are added to the truck scheduling model given in Figure 2:

$$\sum_{o \in O, i \in \mathcal{I}} x_{hio} + \sum_{i \in \mathcal{I}, c \in \mathcal{C}} s^I_{hid} \leq N^I_h + \delta^I_h \qquad \forall h \in \mathcal{H} \quad (10.1)$$

$$\sum_{o \in O, i \in \mathcal{I}} x_{hio} + \sum_{o \in \mathcal{O}} s^O_{ho} \leq N^O_h + \delta^O_h \qquad \forall h \in \mathcal{H} \quad (10.2)$$

$$\sum_{o \in O, i \in \mathcal{I}} x_{hio} \leq M_h + \varepsilon_h \qquad \forall h \in \mathcal{H} \quad (10.3)$$

$$\Pi^\delta_0 = \sum_{h \in \mathcal{H}} \delta^I_h + \delta^O_h \qquad (23)$$

$$\Pi^\varepsilon_0 = \textstyle\sum_{h \in \mathcal{H}} \varepsilon_h \qquad (24)$$

Constraint sets (10.1), (10.2) and (10.3) give a penalty point each time the soft constraint is violated, *i. e.* each time the number of persons necessary to unload, load or transfer the pallets is different from the workload determined by the daily rostering model. The sums of these penalty points, defined by constraints (23) and (24), are then added to the objective function of the model given in Figure 2, thus the new objective is to minimize $\alpha_0\Pi_0^\alpha + \beta_0\Pi_0^\beta + \gamma_0\Pi_0^\gamma + \delta_0\Pi_0^\delta + \varepsilon_0\Pi_0^\varepsilon$.

d) (4) Deduce W from the truck schedule: After the truck scheduling model is solved with the new constraints and new objective function, the output is used to calculate workload W as detailed in section IV. The workload is used as an input in the daily rostering model, together with the values of X and Y fixed in the weekly timetable.

e) (5) Add interval flexibility in the daily rostering model: For the daily truck schedule and employee roster to be able to influence each other until a stable point is reached, it is important to leave some flexibility to the daily rostering model regarding the intervals in which the work can be done. Therefore, constraint set (17) of the daily rostering model in Figure 3

$$\sum_{e\in\mathcal{E}''} x''_{et_1q} = W^1_{t_1q} \qquad \forall t \in \mathcal{T}^1, q \in \mathcal{Q} \quad (17)$$

is replaced by constraint sets (17.1), (17.2) and (17.3) as follows:

$$\sum_{e\in\mathcal{E}''} x''_{et_1q} = W^1_{t_1q} + \varepsilon^+_{t_1q} - \varepsilon^-_{t_1q} \qquad \forall t \in \mathcal{T}^1, q \in \mathcal{Q} \quad (17.1)$$

$$\sum_{e\in\mathcal{E}'', q\in\mathcal{Q}} x''_{et_1q} = \sum_{q\in\mathcal{Q}} W^1_{t_1q} \qquad \forall t \in \mathcal{T}^1 \quad (17.2)$$

$$\Pi_3^\varepsilon = \sum_{t\in\mathcal{T}^1, q\in\mathcal{Q}} \varepsilon^+_{t_1q} + \varepsilon^-_{t_1q} \quad (17.3)$$

Constraint set (17.1) replaces constraint set (17) and changes it into a set of soft constraints. Constraint (17.2) ensures that, despite the flexibility provided to replace the work in different time slots, the total amount of hours worked still matches the workload. The objective function is changed in order to add Π_3^ε, defined in constraint (17.3), to the objective function of the daily rostering model defined in Figure 3.

f) (6) Iterate until reaching a stable point: Using the daily roster, the values of M_h, N_h^I and N_h^O can be updated and used to run the truck scheduling model again. The new versions of the truck scheduling and employee rostering models are run iteratively until a stable point is reached. The stable point is considered reached when the values of the different penalties that measure adjustments, *i. e.* Π_0^δ, Π_0^ε, Π_3^α, Π_3^β and Π_3^ε, do not change any more. Table I recapitulates the different penalties used in the models described in section III as well as the penalties added to connect the two models. In some cases, the iteration does not converge to a single stable point but to a set of two, three or more solutions (oscillator): in this case the loop is stopped and the solution with the smallest objective function Π_3 is chosen.

B. Trucks-first

Calculating the employees timetable first can favor the employees, but leaving the employee-related model to decide when the trucks should be docked could lead to strongly sub-optimal truck schedules. In order to prevent that problem, the trucks-first approach starts as the sequential approach: a truck schedule is first calculated from the instance. The workload W is calculated from the truck schedule (see section IV for details) and used as input to generate the weekly schedule, followed by the daily roster. While the sequential approach stops there, the iterative approach questions this daily roster to adapt it to the truck schedule constraints.

Truck scheduling penalties		*Employee rostering penalties*	
Π_0^α	inbound truck time window penalty	Π_3^α	shift changes compared to the weekly timetable
Π_0^β	outbound truck time window penalty	Π_3^β	task changes compared to the weekly timetable
Π_0^γ	number of pallets in storage		
Π_0^δ	transfer capacity violations		
Π_0^ε	loading/unloading capacity violations	Π_3^ε	interval changes for tasks in $\mathcal{T}^1$

TABLE I: Penalties description

From the output x'' of the daily roster, one can calculate the values of M_q, N_q^I and N_q^O, which are capacity constraints at time interval $q \in \mathcal{Q}$ for the transfer, unloading and loading operations, respectively. The values of these data elements are calculated as detailed in section V-A. The truck schedule is then obtained with the new version of the truck scheduling model described in section V-A, with constraints sets (10.1), (10.2) (10.3), (23) and (24). Based on this truck schedule, a new workload W is calculated and used as input for the daily roster as well as the outputs of the weekly timetabling model X'' an Y''. The version of the employee rostering model used also replaces constraint set (24) by constraints sets (24.1), (24.2) and (24.3) as detailed in section V-A, in order to add flexibility regarding the possible intervals to execute each task.

Similar to the employees-first approach, the truck scheduling and employee daily rostering models are run iteratively until a stable point or an oscillator is reached – for the latter, the solution with the smallest objective function Π_0 is chosen.

VI. Numerical results

In this section, exploratory numerical experiments are carried out: the aim is to demonstrate that the method detailed in section V is a valid way to combine the truck scheduling model with the employee scheduling model. After a presentation of the instance generation process in section VI-A, section VI-B uses an example to show how the iterative approach outperforms the sequential approach. In section VI-C, the performances of both iterative approaches (truck-first and employees-first) are compared and discussed using a bigger instance set.

A. Instance generation

The truck-related parts of the instances correspond to the instance `set3+3` described in Ladier and Alpan [21], which is composed of 11 instances and available online[3]. The employee-related parts of the instances are generated randomly, using the generator proposed by Ladier et al. [12] (available online along with the instances), with the number

[3]http://www.g-scop.fr/ gaujalg/ XDockInstances

of employees set to 10 for the instances where $M = 17$, and set to 15 for the instances where $M = 34$. The time horizon (number of hours $|\mathcal{H}|$) on the employees side is set equal to the value of $|\mathcal{H}|$ on the trucks side.

In order to keep the weekly and daily stages easily comparable, the time unit considered when creating the daily roster (interval) has a length of one hour, thus $\mathcal{H} = \mathcal{Q}$.

The value of τ^{stock}, estimation of the percentage of pallets that go through storage, is set to 3%.

B. Comparison sequential / iterative approaches

When introducing the iterative approach, we pointed out the fact that reaching a local optimum for both models separately does not necessarily mean reaching a good solution when both are combined. This point is illustrated in this section by applying the sequential procedure and an iterative one (here trucks-first) to instance `17_1`. Instance `17_1` has a time horizon $|\mathcal{H}| = 10$, 5 inbound and 5 outbound trucks ($|\mathcal{I}| = |\mathcal{O}| = 5$) serving 3 different clients.

a) Sequential approach: Instance `17_1` represents a rather small platform, therefore the linear program presented in section III can be used to find a truck schedule for the crossdock in a reasonable time. The solution obtained, with an objective value of 0 ($\Pi_0^\alpha = 0$, $\Pi_0^\beta = 0$, $\Pi_0^\gamma = 0$) is displayed in Figure 7.

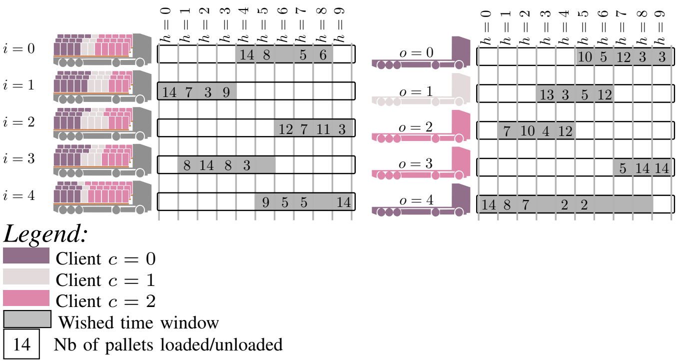


Fig. 7: Sequential approach on `17_1`: truck schedule

As a first approximation, let us assume that this truck schedule will apply to each of the five days of the week. The workload W corresponding to this truck schedule, as well as the qualification matrix Q used in this instance, are as follows:

$$W_{td} = \begin{bmatrix} 8 & 3 & 10 & 0 & 0 & 8 \\ 8 & 3 & 10 & 0 & 0 & 8 \\ 8 & 3 & 10 & 0 & 0 & 8 \\ 8 & 3 & 10 & 0 & 0 & 8 \\ 8 & 3 & 10 & 0 & 0 & 8 \end{bmatrix} \quad Q_{et} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Note that two tasks related to storage have a null workload, since no pallet is put in storage in this solution. Using this workload as an input, the weekly timetabling models give the timetable shown in Figure 8.

Employees 0, 2 and 5 are not put to work in this timetable and are absent all week. Running the employee rostering model for day $d = 0$ (Monday) gives a daily roster exactly equal to the one displayed in Figure 8 for Monday, thus the objective value for the daily rostering step is 0 ($\Pi_3^\alpha = 0$, $\Pi_3^\beta = 0$, $\Pi_3^\gamma = 0$ and $\Pi_3^\delta = 0$).

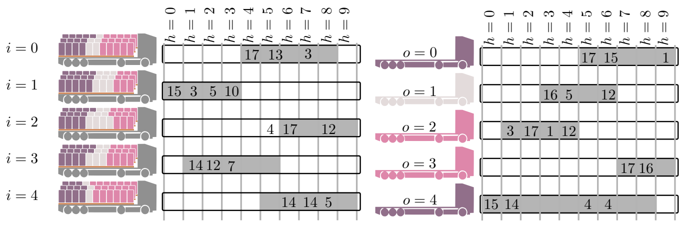
When looking at the objective functions only, this approach seems very good since each model, taken independently, is solved to optimum with no soft constraint violated. But can these two results (truck timetable and employee daily roster) be combined easily? Looking at the number of employees allocated to each task at the different time units on Monday (Figure 8), and using equations 1 to 3, we can calculate the employee capacities available at every time unit $h \in \mathcal{H}$:

$$\begin{array}{lcl} M & = & [\ 17\ \ 17\ \ 17\ \ 17\ \ 17\ \ 17\ \ 34\ \ 17\ \ 17\ \ 0\ ] \\ N^I & = & [\ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 0\ \ 0\ \ 20\ \ 20\ \ 0\ ] \\ N^O & = & [\ 0\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 0\ ] \end{array}$$

For example, Figure 8 shows that two employees are allocated to direct transfer at time $h = 6$, therefore $M_6 = 34$.

Looking at the truck schedule used by the sequential approach (presented in Figure 7), we can see that those capacity constraints are violated many times. The loading/unloading capacities N^I and N^O are violated for 51 pallets in total (all the pallets loaded or unloaded when the capacity is 0 for those tasks), and the transfer capacity M for 17 pallets (all the pallets transferred at time $h = 9$). That would be equivalent to objective values $\Pi_0^\delta = 51$ and $\Pi_0^\varepsilon = 17$. Is it possible to do better with the trucks-first approach?

b) Trucks-first approach: The trucks-first approach starts exactly like the sequential approach, but the values of M, N^I and N^O are now integrated to the truck scheduling model as soft constraints. The result, displayed in Figure **??**, yields to the objective function $\Pi_0 = 58$ where $\Pi_0^\alpha = 1$, $\Pi_0^\beta = 0$, $\Pi_0^\gamma = 0$, $\Pi_0^\delta = 57$, $\Pi_0^\varepsilon = 0$.



From this truck schedule, the detailed workload for each interval $q \in \mathcal{Q}$ can be expressed as shown in Figure 9. Then, using this workload as input, the daily rostering model is run again to give the result shown in Figure 9. The corresponding penalties are $\Pi_3^\alpha = 2$, $\Pi_3^\beta = 6$, $\Pi_3^\gamma = 0$, $\Pi_3^\delta = 0$, $\Pi_3^\varepsilon = 8$. It means that there is a 2-hour change in the allocated shifts compared to the weekly timetable (for employee 9) and 6 hours of task changes (for the tasks of employee 8).

New capacity constraints can be derived from this employee roster:

$$\begin{array}{lcl} M & = & [\ 17\ \ 17\ \ 17\ \ 17\ \ 17\ \ 17\ \ 34\ \ 17\ \ 17\ \ 0\ ] \\ N^I & = & [\ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 40\ \ 20\ \ 20\ \ 20\ \ 0\ ] \\ N^O & = & [\ 0\ \ 20\ \ 40\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 20\ \ 0\ ] \end{array}$$

and used in the truck scheduling model to obtain the truck

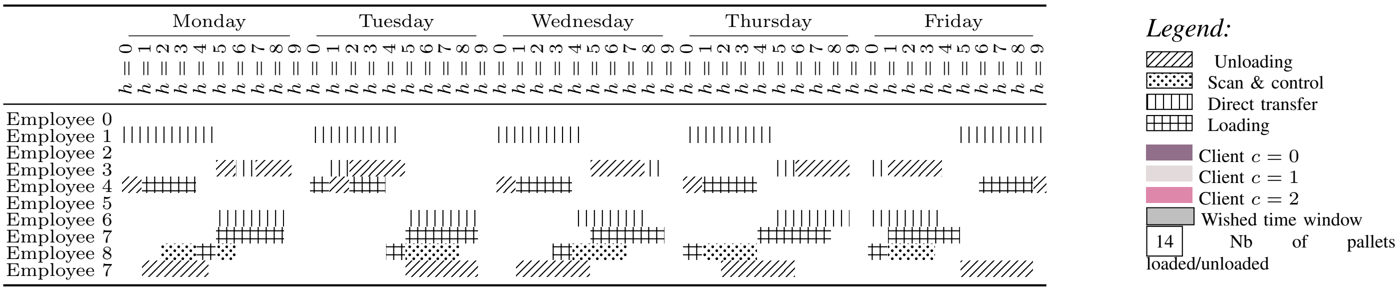


Fig. 8: Sequential approach on `17_1`: weekly timetable

$$W_{t_1 q} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 2 & 1 & 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

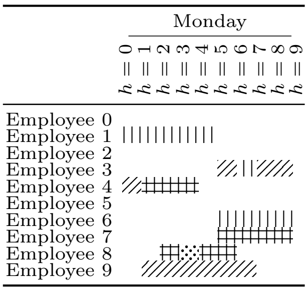


Fig. 9: Iteration 2 on `17_1`: employee roster

schedule in Figure 10, with penalties $\Pi_0^\alpha = 1$ (because of the hour added at $h = 5$ for inbound truck $i = 2$), $\Pi_0^\beta = 0$, $\Pi_0^\gamma = 0$, $\Pi_0^\delta = 37$, $\Pi_0^\varepsilon = 0$.

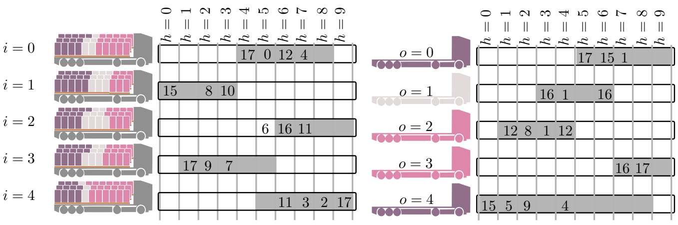


Fig. 10: Iteration 2 on `17_1`: truck schedule

The next iteration yields exactly the same solution – therefore the procedure stops after three iterations in total. The comparison between the sequential and the truck first approach, in terms of value of the objective function, is done in Table II. The trucks-first approach reduces the values of Π_0^δ and Π_0^ε, *i. e.* reduces the violations of the staff-related capacity constraints. It also increases the value of Π_0^α (one inbound truck is assigned to a time windows slightly different from its wish) and the difference between the weekly timetable and the daily roster (Π_3^α, Π_3^β), but it is a price to pay to make the truck schedule and the employee roster more compatible.

C. Comparison employees-first / trucks-first

Intuitively, one could think that the employees-first procedure favors the employees' wishes, while the truck-first procedure favors the transportation providers' wishes instead. The results obtained on instance `set3+3`, displayed in Table III, confirm this idea. The first line correspond to instance `17_1`

	Truck timetable					Employee roster		
	Π_0^α	Π_0^β	Π_0^γ	Π_0^δ	Π_0^ε	Π_3^α	Π_3^β	Π_3^ε
Sequential	0	0	0	51	17	0	0	0
Trucks-first	1	0	0	37	0	2	6	8

TABLE II: Sequential/trucks-first results for `17_1`

already investigated in the previous section; 10 other instances are tested besides. Most of the time, the values of Π_3^α and Π_3^β are smaller for the employees-first approach. All the other penalties, however, are bigger for the employee approach. The penalties regarding truck time windows assignments (Π_0^α and Π_0^β), especially, are significantly bigger for the employees-first approach compared to the trucks-first approach. On this set of small instances, the trucks-first approach therefore dominates the employees-first approach.

VII. Conclusion

This article demonstrates how an iterative procedure can be used to combine a truck scheduling model and an employee weekly timetabling and daily rostering model to plan the operations of a cross-docking platform in an integrated manner. Numerical experiments on small instances show that the best results are obtained when the truck scheduling model is run first. Further work is needed to check whether this result scales-up for bigger instances, and to analyze the behavior of the system when the different parameters change. Quantifying the Pareto optimum would also be interesting.

The limits of this approach reside in the fact that no fully integrated model is available, therefore the quality of the solutions given by the iterative process cannot be compared to the optimal value. A model integrating all the industrial constraints of the truck scheduling and the employee rostering would probably be too hard to be solved. However, the different decomposition processes proposed by Guyon et al. [17, 18] might be applicable to our case (or a simplified version of it). They are exact methods yielding to optimal solutions. Specifically, the cut generation process presented by Guyon et al. [17] splits the model into a master problem, which assigns a work pattern to each operator, and a sub-problem which checks the feasibility of the assignment – this sub-problem is actually a maximum flow problem on a directed transportation network. Because the crossdock truck scheduling problem also contains a maximum flow problem as a sub-problem (see Ladier and Alpan [21]), applying the method

	Employees-first									Trucks-first								
	Ite.	Π_0^α	Π_0^β	Π_0^γ	Π_0^δ	Π_0^ε	Π_3^α	Π_3^β	Π_3^ε	Ite.	Π_0^α	Π_0^β	Π_0^γ	Π_0^δ	Π_0^ε	Π_3^α	Π_3^β	Π_3^ε
17_1	3	1	1	0	135	1	2	3	14	3	1	0	0	37	0	2	6	8
17_2	4	8	12	0	195	0	2	3	14	3	0	1	0	31	0	3	6	6
17_3	5	16	16	0	208	0	2	2	24	3	0	2	0	31	0	3	11	6
17_4	2	15	15	0	208	0	0	0	24	3	0	0	0	51	1	3	9	4
17_5	3	14	12	0	198	6	2	3	22	3	0	3	0	51	0	3	9	4
34_1	3	0	0	0	52	0	4	3	6	3	0	0	0	38	0	4	4	4
34_2	3	0	0	0	72	8	4	4	14	4	0	1	0	42	0	4	8	4
34_3	2	0	1	0	115	14	2	1	14	3	0	1	0	44	0	0	0	4
34_4	3	12	12	0	210	0	2	4	22	5	1	2	0	51	0	2	8	6
34_5	4	18	20	0	210	12	2	4	24	5	0	2	0	51	1	2	9	6
34_6	5	24	24	0	198	0	4	4	24	5	0	3	0	48	0	6	8	6

TABLE III: Results for both iterative approaches

proposed by Guyon et al. to the cross-docking environment seems a promising idea. Applying Benders decomposition is also a possible perspective in order to get an exact solution to the integrated problem.

ACKNOWLEDGMENT

G-SCOP is partner of the LabEx PERSYVAL-Lab (ANR–11-LABX-0025).

REFERENCES

[1] P. Baker and Z. Halim, "An exploration of warehouse automation implementations: cost, service and flexibility issues," *Supply Chain Management: An International Journal*, vol. 12, pp. 129–138, 2007.

[2] A. Granlund, "Competitive internal logistics systems through automation," Ph.D. dissertation, Mälardalen University, School of Innovation, Design and Engineering, 2011.

[3] D. D. Graham, "Warehouse of the future," *Frontline Solutions*, vol. 4, pp. 20–26, 2003.

[4] J. P. van den Berg, *Integral warehouse management: the next generation in transparency, collaboration and warehouse management systems*. Utrecht, The Netherlands: Management Outlook, 2007.

[5] A. Wren, "Scheduling, timetabling and rostering – A special relationship?" in *International Conference on the Practice and Theory of Automated Timetabling*, ser. Lecture Notes in Computer Science, Edinburgh, UK, 1996, vol. 1153, pp. 46–75.

[6] J. Van Belle, P. Valckenaers, and D. Cattrysse, "Cross-docking: State of the art," *Omega*, vol. 40, no. 6, pp. 827–846, 2012.

[7] C. T. Maravelias and C. Sung, "Integration of production planning and scheduling: Overview, challenges and opportunities," *Computers & Chemical Engineering*, vol. 33, no. 12, pp. 1919–1930, 2009.

[8] A.-L. Ladier, "Planification des opérations de cross-docking – Prise en compte des incertitudes opérationnelles et de la capacité des ressources internes," Ph.D. dissertation, Université de Grenoble, 2014.

[9] M. H. Fazel Zarandi, H. Khorshidian, and M. Akbarpour Shirazi, "A constraint programming model for the scheduling of JIT cross-docking systems with preemption," *Journal of Intelligent Manufacturing*, pp. 1–17, Jan. 2014.

[10] M. Günther and V. Nissen, "A comparison of three heuristics on a practical case of sub-daily staff scheduling," in *Annals of Operations Research*, vol. 218, no. 1, 2014, pp. 201–219.

[11] ——, "Sub-daily staff scheduling for a logistics service provider," *Künstliche Intelligenz*, vol. 24, no. 2, pp. 105–113, 2010.

[12] A.-L. Ladier, G. Alpan, and B. Penz, "Joint employee weekly timetabling and daily rostering: A decision-support tool for a logistics platform," *European Journal of Operational Research*, vol. 234, no. 1, pp. 278–291, 2014.

[13] C. S. Ko, H. K. Lee, E. J. Choi, and T. Kim, "A genetic algorithm approach to dock door assignment in automated cross-docking terminal with restricted layout," in *International Conference on Genetic and Evolutionary Methods*, Las Vegas, Nevada, USA, 2008.

[14] Z. Li, C. H. Sim, W. He, and C. C. Chen, "A solution for cross-docking operations planning, scheduling and coordination," *Journal of Service Science and Management*, vol. 05, no. 02, pp. 111–117, 2012.

[15] C. Artigues, M. Gendreau, and L.-M. Rousseau, "A flexible model and a hybrid exact method for integrated employee timetabling and production scheduling," *Lecture Notes in Computer Science*, vol. 3867, no. Practice and Theory of Automated Timetabling VI, pp. 67–84, 2007.

[16] C. Artigues, M. Gendreau, L.-M. Rousseau, and A. Vergnaud, "Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound," *Computers & Operations Research*, vol. 36, no. 8, pp. 2330–2340, 2009.

[17] O. Guyon, P. Lemaire, E. Pinson, and D. Rivreau, "Cut generation for an integrated employee timetabling and production scheduling problem," *European Journal of Operational Research*, vol. 201, no. 2, pp. 557–567, 2010.

[18] ——, "Solving an integrated job-shop problem with human resource constraints," *Annals of Operations Research*, vol. 213, no. 1, pp. 147–171, 2014.

[19] A. Mercier and F. Soumis, "An integrated aircraft routing, crew scheduling and flight retiming model," *Computers & Operations Research*, vol. 34, no. 8, pp. 2251–2265,

2007.
[20] O. Weide, D. Ryan, and M. Ehrgott, “An iterative approach to robust and integrated aircraft routing and crew scheduling,” *Computers & Operations Research*, vol. 37, no. 5, pp. 833–844, 2010.
[21] A.-L. Ladier and G. Alpan, “Crossdock truck scheduling with time windows – Earliness, tardiness and storage policies,” *Journal of Intelligent Manufacturing*, 2014.