Scheduling truck arrivals and departures in a crossdock: earliness, tardiness and storage policies

(presented at the 5^{th} IESM Conference, October 2013, Rabat, Morocco)

Anne-Laure Ladier
Grenoble-INP / UJF-Grenoble 1 / CNRS
G-SCOP UMR5272 Grenoble, F-38031
46 avenue Félix Viallet, 38000 Grenoble, France
Email: anne-laure.ladier@g-scop.grenoble-inp.fr

Gülgün Alpan
Grenoble-INP / UJF-Grenoble 1 / CNRS
G-SCOP UMR5272 Grenoble, F-38031
46 avenue Félix Viallet, 38000 Grenoble, France
Email: gulgun.alpan@g-scop.grenoble-inp.fr

***Abstract*—This article proposes a model to simultaneously plan truck arrivals, truck departures and internal pallet handling in a crossdocking platform. The objective is to minimize both the total number of pallets put in storage on the planning horizon, and the dissatisfaction of the transportation providers, by creating a truck schedule as close as possible to the wished schedule they communicate in advance. The problem is modeled with an integer program, which is tested on generated instances to assess its performances, especially regarding the computation time. Since the execution takes too long to be used by platform managers on a daily basis, two heuristics are also proposed and tested. We show in which conditions each heuristic performs best, which can help in choosing a solution method when confronted to a real-life problem.**

I. Introduction

In a context of economic tensions and global competition, where consumers become highly impatient and volatile, the industrial competitive advantage can lie in more intensive and faster goods flow. The cross-docking technique is meant to bring rapidity and reactivity in the supply chain. A cross docking facility (also called crossdock) is a logistics platform where the goods are unloaded from inbound trucks, sorted, dispatched, and directly reloaded in outbound trucks. Inventory is not necessarily prohibited but reduced to a minimum, therefore most of the goods stay less than 24 hours in the cross dock (Li et al. [1]). Interested readers can refer to Van Belle et al. [2] for a review of the state-of-the-art in cross-docking.

The advantages of such a just-in-time organization are known: speeding the goods flow, increasing the reactivity, cutting the inventory costs (Apte et al. [3]). Nevertheless, the synchronization of the trucks is a key element for a fluent functioning of the platform: a flawless planning system is thus required for a successful cross dock implementation. Warehouse operations planning problems are numerous and an extensive review can be found in Gu et al. [4]. However, only a few research papers deal with the scheduling of the daily operations (truck arrivals, departures, storage) inside a cross dock.

This article proposes a model to plan the internal transfers in a cross-dock and schedule the arrivals and departures of the inbound and outbound trucks. The problem consisting in scheduling trucks on a given number of doors, over a given time horizon, is called the *truck scheduling problem*. For a review of the papers dealing with this problem, see Boysen and Fliedner [5]. Several variants of this problem have been identified: the first one seeks to schedule the trucks on specific doors, in order to minimize the total distance of the shipments inside the platform. A second approach seeks to minimize the costs; for instance, Alpan et al. [6] propose a model to minimize the total inventory and preemption costs. The third approach considers the time dimension as an optimisation objective; see for instance Boysen et al. [7] who give insights on a basic inbound and outbound truck scheduling problem with one inbound and one outbound door, minimizing the makespan.

The notion of delay or tardiness of the shipments in truck scheduling problem is considered in the following articles: Boysen et al. [5] deal with a cross docking problem with fixed outbound schedules, where the objective function is to minimize the weighted number of delayed shipments. Boysen [8] proposes a model for a frozen food platform in which the storage is forbidden. In this model, the objective is to minimize the flow time, processing time and tardiness of the outbound trucks. In both papers the focus is on the tardiness of the outbound operations.

In a logistics platform, the punctuality of the trucks is of crucial importance for the managers, not only for the outbound but also for the inbound trucks. Early truck arrivals may disturb the internal operations as much as delays (e.g. unexpected congestion inside the platform or in the parking area, need for a reorganization of internal resources, etc.). Therefore, unlike previous work found in the literature, this article considers both the earliness and tardiness of the trucks, for both inbound and outbound operations.

The earliness or tardiness of a given truck is only meaningful when compared to a reference, a wished arrival or departure time. In order to bring more flexibility, we define this reference as a time window rather than a single time. The notion of time windows for the loading or the unloading operation is first used by Lim et al. [9], who consider general transshipment networks, with multiple platforms, where both loading and unloading are done within specified time windows at different locations. Regarding crossdock internal operations scheduling, time windows are introduced by Lim et al. [10]. They describe a *truck dock assignment problem* in which the trucks are loaded or unloaded during fixed time windows. Their approach differs from ours because Lim et al. take the layout of the cross dock into account and minimize the goods travel distance, while our focus point is on earliness and tardiness of inbound and outbound trucks.

To the best of our knowledge, there is only one paper dealing with a crossdock scheduling problem with time windows and tardiness penalties. Miao et al. [11] search for an inbound and outbound truck assignment which minimizes the transfer costs and lateness penalties, subject to time windows for truck arrival and departures. The layout is taken into account, since the transfer cost used depends on the distance to be covered by the handling devices. Miao et al. [11] assume that a late truck is lost, and the lateness penalty used is the total number of unfulfilled shipments. In reality, this assumption barely holds: according to our industrial experience, a truck is loaded and goods are delivered despite the tardiness. Furthermore, the magnitude of lateness is very important since small delays may be caught up during the transportation, from the platform to the client. We, therefore, use the time windows as soft constraints, and consider penalties on the total tardiness. Other differences are related to internal operations. In our case, the transfer of goods inside the platform is done in hidden time, hence the distance traveled is not of importance. And finally, we consider temporary storage which is not explicitly modeled in Miao et al. [11]. In crossdocks, existence of a temporary storage adds a certain flexibility in operations management, but also generates costs. We express the related costs with penalties on the utilization of the stock.

To sum up, this paper proposes a model to plan inbound and outbound truck arrivals and departures, as well as pallet moves through the cross dock. The objective is to stay as close as possible to a wished trucks planning given as input, while minimizing the inventory level.

The rest of the paper is organized as follows. In Section II we present and formulate the problem with an integer program. Section III presents the computation limits of such a formulation. Two different heuristics are then presented in Section IV, to build feasible solutions in reasonable time for real size problems. Numerical tests assessing the performance of the heuristics are also presented in this section. Finally, concluding remarks are given in Section V.

II. Problem definition and modeling

We wish to build a schedule of the inbound and outbound trucks, maximizing the transportation providers' satisfaction and minimizing storage. The goal of the model is to provide the logistics manager with a decision-aid tool for scheduling its operations and storage plan.

For the inbound and outbound trucks, the manager knows the preferences of the transportation provider regarding arrival and departure times. The employees timetabling being crucial for the platform overall performance, the manager needs precise information about the workload inside the platform, both for moves from trucks to trucks and for moves to stock (which require different forklift licenses). Tracking pallet moves is also needed to ensure the synchronization of the inbound flows with the outbound flows.

A. Assumptions

We assume that the doors have an *exclusive* mode of service (Boysen and Fliedner [5]), which means that each door is either inbound or outbound, but cannot switch from one role to the other during the day. This assumption is common in academic work (see for example Boysen and Fliedner [5]), and it is also a common practice in industry – even though it may lower the efficiency of the dock utilization. The reason is that having fixed inbound and outbound doors eases the operations management inside a platform. The number of inbound and outbound doors is thus known and used as input data.

We assume that transfer, loading and unloading operations are all done within the same time period; consequently we define the time period to be long enough (e.g. at least half an hour) to ensure the product transfers in masked time. We do not consider the distance of the transfer, and hence the location of the door in the platform is not taken into account.

We make the hypothesis that the exact content of the inbound trucks (number of pallets for each destination) is known. When the trucks arrive, we assume they are entirely unloaded on the dock floor (then presumably checked and scanned for the reception in the WMS). It means that the pallets can then be picked from the floor in any order. The outbound trucks have a fixed capacity C, and cannot leave before they are fully loaded.

When the matching truck is not available to load a given pallet, the pallet is placed in storage – there are no stock capacity constraints. Note that our model does not follow a FIFO policy to empty the stock. We consider that the pallets will not be stored for a very long time, therefore the costly operation is the placing in storage.

B. Input data

From the assumptions detailed above, we can make a list of the input data we consider in our model:

H	set of time periods (e.g. half an hour) in the planning horizon considered
I	set of inbound trucks
O	set of outbound trucks
D	set of destinations
Q_{id}	number of pallets for destination $d \in D$ in truck $i \in I$
Z_{do}	= 1 if truck $o \in O$ is for destination $d \in D$, 0 otherwise
N^I	number of inbound doors
N^O	number of outbound doors
M	maximum number of units that can be moved within one time period from one truck to another
C	outbound trucks capacity

The data listed above (and especially the truck capacity, the number of inbound and outbound doors and the bound on the number of units moved) all correspond to physical constraints in the cross dock, that cannot be violated. They will therefore become hard constraints in our model.

For each inbound (resp. outbound) truck, we know its earliest possible arrival time and latest possible departure time.

$$
\begin{array}{lll}
\min & \alpha \sum_{i\in I}\sum_{k\in K_i}\Pi^I_{ik}w^I_{ik} + \beta\sum_{o\in O}\sum_{k\in K_o}\Pi^O_{ok}w^O_{ok} + \gamma\sum_{h\in H, i\in I, d\in D} s^I_{hid} & \\
\text{s.t.} & \sum_{i\in I}\sum_{k\in K_i} W^I_{ikh}w^I_{ik} \le N^I & \forall h\in H \qquad (1)\\
& \sum_{o\in O}\sum_{k\in K_o} W^O_{okh}w^O_{ok} \le N^O & \forall h\in H \qquad (2)\\
& x_{hio} + s^I_{hid} \le M\sum_{k\in K_i} W^I_{ikh}w^I_{ik} & \forall h\in H, i\in I, o\in O \qquad (3)\\
& x_{hio} + s^O_{ho} \le M\sum_{k\in K_o} W^I_{okh}w^O_{ok} & \forall h\in H, i\in I, o\in O \qquad (4)\\
& \sum_{h\in H, o\in O} Z_{do}x_{hio} + \sum_{h\in H} s^I_{hid} = Q_{id} & \forall i\in I, d\in D \qquad (5)\\
& \sum_{i\in I, h\in H} x_{hio} + \sum_{h\in H} s^O_{ho} = C & \forall o\in O \qquad (6)\\
& \sum_{k\in K_i} w_{ik} = 1 & \forall i\in I \qquad (7)\\
& \sum_{k\in K_o} w_{ok} = 1 & \forall o\in O \qquad (8)\\
& s_{hd} = s_{(h-1)d} + \sum_{i\in I} s^I_{hid} - \sum_{o\in O} Z_{do}s^O_{ho} & \forall d\in D, h\in H\setminus\{0\} \qquad (9)\\
& s_{0d} = \sum_{i\in I} s^I_{0id} - \sum_{o\in O} Z_{do}s^O_{0o} & \forall d\in D \qquad (10)\\
& x_{hio}, s^I_{hid}, s^O_{ho}, s_{hd} \in \mathbb{N}^+ & \forall h\in H, i\in I, o\in O, d\in D\\
& w^I_{ik}, w^O_{ok} \in \{0,1\} & \forall i\in I, o\in O, k\in K
\end{array}
$$

(IP*)

Therefore, we can enumerate the possible slots in which the truck could be present. We note K_i (resp. K_o) as the set of possible presence slots of the truck $i \in I$ (resp. $o \in O$). These possible presence slots are described by matrices W^I and W^O.

- W^I_{ikh} = 1 if hour $h \in H$ is in slot $k \in K_i$ for inbound truck $i \in I$
- W^O_{okh} = 1 if hour $h \in H$ is in slot $k \in K_o$ for outbound truck $o \in O$

We assume that we also know the wishes of the transport provider regarding the arrival and presence time of trucks. We seek at satisfying them as much as possible, but time slots can be changed if necessary for the ongoing operations. Those wishes are therefore seen as soft constraints: if trucks are scheduled outside their wished slots, penalties are paid. Those penalties Π^I and Π^O are therefore defined as follows:

- Π^I_{ik} penalty paid for using slot $k \in K_i$ for truck $i \in I$, if k is outside the wished time window expressed by the transport provider;
- Π^O_{ok} penalty paid for using slot $k \in K_o$ for truck $o \in O$.

C. Decision variables

The model aims at defining the truck schedule, with the objective of being as close as possible to the wishes of the transportation providers, and at the same time minimizing the storage. Monitoring the pallet moves is necessary to ensure the synchronization of the inbound and outbound flows. The model therefore uses the following decisions variables:

- x_{hio} amount of units going from truck i to truck o at time period h;
- w^I_{ik} =1 if slot $k \in K_i$ is chosen for truck i, 0 otherwise;
- w^O_{ok} =1 if slot $k \in K_o$ is chosen for truck o, 0 otherwise;
- s^I_{hid} amount of products with destination d going from truck i to the storage location at time period h;
- s^O_{ho} amount of products going from the storage location to truck o at time period h;
- s_{hd} amount of products for destination d stored at time period h.

D. Integer program

The planning problem can now be formulated as an integer program (IP*): see above.

The objective is to minimize the time windows penalties for the inbound and the outbound trucks, as well as the number of pallets placed in the storage area. α, β and γ are the coefficients that weight those often conflicting objectives.

Constraint (1) (resp. (2)) checks that the number of inbound (resp. outbound) trucks present during a given time period does not exceed the number of inbound (resp. outbound) doors. Constraint (3) (resp. (4)) ensures that the pallet moves from the inbound trucks (resp. to the outbound trucks) occur only when the concerned truck is present. Constraint (5) makes sure that all the pallets of a given inbound truck are unloaded and dispatched to the right destination. Constraint (6) indicates the capacity of the outbound trucks, and makes sure that they are fully loaded. Constraints (7) and (8) make sure that each inbound (resp. outbound) truck is assigned to a single presence time window. Constraints (9) and (10) give the stock conservation rule for all $h \in H \setminus \{0\}$ and for $h = 0$, respectively.

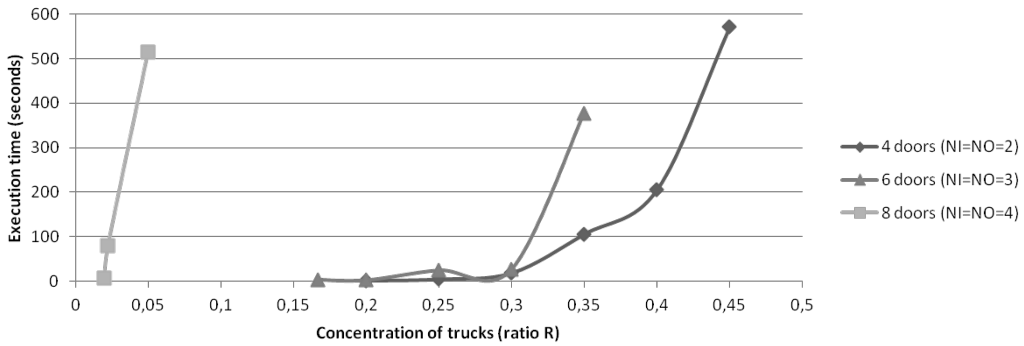


Figure 1: Execution time of (IP*) as a function of the concentration of trucks

III. Numerical tests and limits of the integer program

In this section, we first present the instance generation protocol, then present some numerical results based on the generated instances.

A. Instances generation

We consider that an instance is characterized by the parameters $|H|$, $|I|$, $|O|$, $|D|$, N^I, N^O, M, and C. Note that the total number of destinations should be picked such that $|D| \leq |O|$, to make sure that there is at least one outbound truck per destination.

For each inbound (resp. outbound) truck, we set 0 as earliest arrival time and $|H|$ as latest departure time – this is chosen to make sure that we do not restrict the solution space when testing. Two random integers are picked within this range to get the wished arrival and departure time; then W^I and W^O are generated as explained in section II-B.

The penalties Π^I and Π^O are directly calculated from W^I and W^O, as the number of hours outside the "wished" range in slot k.

To generate Z_{do}, the first $|D|$ columns of the matrix are filled with "1" along the diagonal, to make sure that each destination is served by at least one truck. The remaining columns are filled picking a random destination number for each truck left.

Q_{id} describes the content of the incoming trucks. Its generation process must ensure that the inbound quantity for each destination is equivalent to the capacity of the outbound trucks for this destination, and that the number of pallets per inbound truck is not too different from one truck to another. With these elements in mind, we calculate for each destination, the total number of inbound pallets $C\sum_{o\in O} Z_{do}$. This quantity is spread amongst the inbound trucks by picking a random truck for each pallet, and allocating this pallet to the truck if it is not full – the maximum quantity per truck being $1 + \frac{C|O|}{|I|}$.

Finally, the values for M, N^I, N^o and $|H|$ are chosen carefully to avoid obviously infeasible instance sets.

B. Numerical results

All linear programs are run with IBM ILOG CPLEX Optimizers 12.2, on a personal computer with a 2.40GHz processor and a 4.00GB RAM.

The tests are carried with the following input parameters: $|H| = 10$, $|D| = 4$, $M = 4$, $C = 4$. The number of doors is fixed to $N^I = N^O = 2$, $N^I = N^O = 3$ or $N^I = N^O = 4$. Note that there are only four to eight doors in total, thus the instances tested can only represent a very small platform. The coefficients α, β and γ are assumed equally important and are all set to 0.33. We test the performance of (IP*) with different number of doors by increasing the number of trucks $|I|$ and $|O|$, setting $|I| = |O|$. For the sake of comparison, we present in Figure 1 the results as a function of the concentration of trucks. The concentration of trucks (in trucks per door per hour) is defined by the ratio $R = \frac{|I|+|O|}{(N^I+N^O)|H|}$.

Even with very small platforms (8 doors or less) and low concentration rates, we can see that the execution times increases very quickly. If we consider that 10 seconds is the limit for a logistics manager to use this program as a daily decision-aid tool, then we cannot deal with more than 10 trucks on a platform with two inbound and two outbound doors. Obviously, the performances of (IP*) in terms of computation time are not good enough to use it on a daily basis within crossdocks.

IV. Heuristics

Since the integer program presented in Section II-D takes too long to compute instances of real-life size, we propose in this section two heuristics. The principle is to relax a part of (IP*), in order to simplify the number of decisions taken during the execution of (IP*). In the first heuristic, we fix the inbound trucks schedule, while the outbound schedule is considered fixed in the second heuristic. In both heuristics, the schedule of the trucks considered as "fixed" is first determined with a dedicated integer program. The principle is described in Figure 2.

A. Heuristic 1

The first step of heuristic (IP1) uses the wished presence time windows of the outbound trucks as data in order to

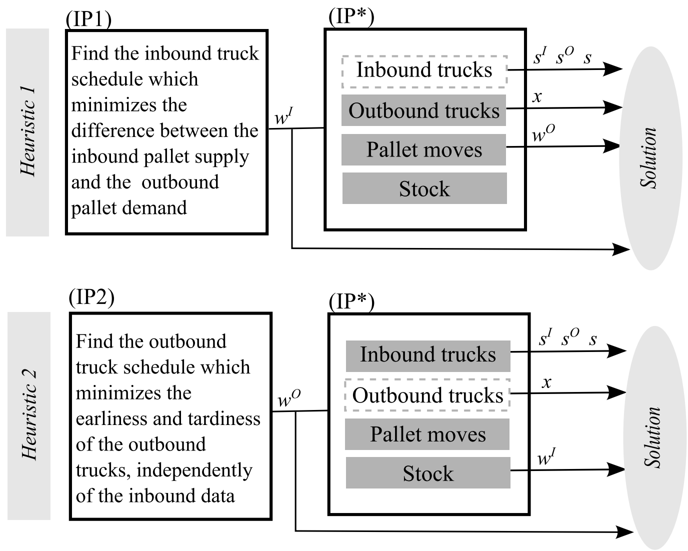


Figure 2: Principle of heuristics H1 and H2

determine a good schedule for the inbound trucks. Then the second step (IP*) uses this inbound truck schedule as data, in order to compute the final schedule of the outbound trucks.

Let us assume, just for this first part of the heuristic, that the wished departure and arrival times of the outbound trucks are all satisfied. Using the matrix Z which indicates the destination of each outbound truck, we can easily calculate X^O, a binary matrix defined as follows:

X^O_{dh} $= 1$ if there is an outbound truck for the destination d present at time period h, 0 otherwise.

Integer program (IP1) uses w^I_{ik} as a decision variable, as well as two new variables that measure the difference between the inbound and the outbound plans:

δ^+_{dh} for time period $h \in H$, positive difference between the number of pallets for destination $d \in D$ available to be unloaded, and the number of pallets that can be loaded in the trucks for d present at the outbound doors.

δ^-_{dh} for time period $h \in H$, negative difference between the number of pallets for destination $d \in D$ available to be unloaded, and the number of pallets that can be loaded in the trucks for d present at the outbound doors.

(IP1) is thus formulated as shown here opposite.

The objective is to minimize the total difference between the inbound pallet supply and the outbound pallet demand. Constraint (11) defines δ^+ and δ^- as described above. Constraint (12) ensures that the number of inbound doors is enforced, while constraint (13) makes sure that only one time window is assigned to each inbound truck.

In the second step, the outcome of (IP1) w^I_{ik} is used as a data to run (IP*). The formulation of (IP*) does not change,

$$
\begin{array}{lll}
\min & \sum_{d\in D,h\in H}(\delta^+_{dh}+\delta^-_{dh}) & \\
\text{s.t.} & \sum_{i\in I}\sum_{k\in K_i} Q_{id}W^I_{ikh}w^I_{ik} & \\
& \quad = MX^O_{dh}+\delta^+_{dh}-\delta^-_{dh} & \forall d\in D, h\in H \quad (11)\\
& \sum_{i\in I}\sum_{k\in K_i} W^I_{ikh}w^I_{ik} \le N^I & \forall h\in H \quad (12)\\
& \sum_{k\in K_i} w_{ik} = 1 & \forall i\in I \quad (13)\\
& \delta^+_{dh},\delta^-_{dh}\in\mathbb{N}^+ & \forall d\in D, h\in H\\
& w^I_{ik}\in\{0,1\} & \forall i\in I, k\in K
\end{array}
$$

(IP1)

except for the fact that w^I_{ik} is no longer a decision variable but rather an input data. Constraints (1) and (7) are therefore discarded. Note that the term of the objective function which includes w^I_{ik} is not removed, although it is now a constant, so that the objective value stays comparable to the results of (IP*).

B. Heuristic 2

The first step of heuristic (IP2) builds a feasible outbound truck schedule which minimizes the earliness and tardiness of the outbound trucks, independently of the inbound data. Then, considering the outbound data fixed, (IP*) is used to generate the inbound truck schedule.

Integer program (IP2) uses w^O_{ok} as the only decision variable. It is formulated as shown below.

The objective is to minimize the outbound transport providers' dissatisfaction. Constraint (14) ensures that the

$$
\begin{array}{lll}
\min & \sum_{o \in O} \sum_{k \in K_o} \Pi^O_{ok} w^O_{ok} & \\
\text{s.t.} & \sum_{o \in O} \sum_{k \in K_o} W^O_{okh} w^O_{ok} \leq N^O & \forall h \in H \quad (14) \\
& \sum_{k \in K_o} w_{ok} = 1 & \forall o \in O \quad (15) \\
& w^O_{ok} \in \{0, 1\} & \forall o \in O, k \in K
\end{array}
$$

(IP2)

number of trucks in use for any time period does not exceed the number of outbound doors, while constraint (15) makes sure that only one time window is assigned to each outbound truck.

In the second step, the outcome of (IP2) w^O_{ok} is used as a data to run (IP*). Similarly to what was done in section IV-A, the formulation of (IP*) does not change, except for the fact that w^O_{ok} is no longer a decision variable but rather an input data. Constraints (2) and (8) are now discarded. The term of the objective function which includes w^O_{ok} is not removed, for the sake of comparison.

C. Numerical tests

In this section, we test the heuristics described above, in order to assess their performances regarding computation time, compare their results to the optimal solution, and see in which situation each heuristic provides good results. Similar to Section III-B, the data parameters are set such that $|H| = 10$, $|D| = 4$, $M = 4$ and $C = 4$.

Keeping the concentration of trucks fixed and equal to 0.4 truck/door/hour, we monitor in the first set of tests the total execution time of the heuristic when the number of doors increases. The coefficients α, β and γ are all equal and set to 0.33. Each value in the figure is the average of the execution times obtained for 10 different instances, generated randomly from the parameters, as explained in section III-A. The result is displayed in Figure 3.

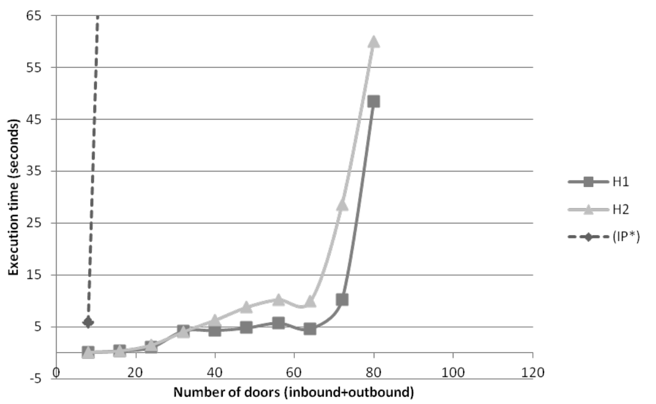


Figure 3: Execution time when the number of doors increases (concentration ratio $R = 0.4$)

First of all, we note that the heuristics are 75 times faster than (IP*) on the average for 2 doors and 8 trucks in total. The execution of (IP*) for 4 doors and 16 trucks takes 205 seconds on average. The solution of (IP*) for this instance is not represented in the figure to avoid stretching the scale too much. The heuristics are 570 times faster than (IP*) in this case.

H1 can be computed in less than 10 seconds with up to 72 doors in the platform, whereas H2 can only handle 64 doors in 10 seconds. Within one minute, we can get a result for 80 doors. We note that the execution time increases exponentially beyond 80 doors. Since cross-docking platforms can be larger than 80 doors, different strategies should be applied to deal with the largest platforms. Metaheuristics using neighbourhood search is an option to be investigated in that case.

In the second set of tests, we assess the performance of the heuristics compared to (IP*). We therefore fix the number of doors ($N^I = N^O = 4$) and the number of trucks ($|I| = |O| = 8$). The instance is deliberately small (concentration 0.2 trucks/door/hour) so that the computation time of (IP*) stays reasonable. For each dot on figure 4, 10 different instances are generated from the data parameters, and we calculate the difference between the objective values of the heuristics and the optimal value given by (IP*). We vary the coefficients α, β and γ, keeping $\alpha + \beta + \gamma = 1$.

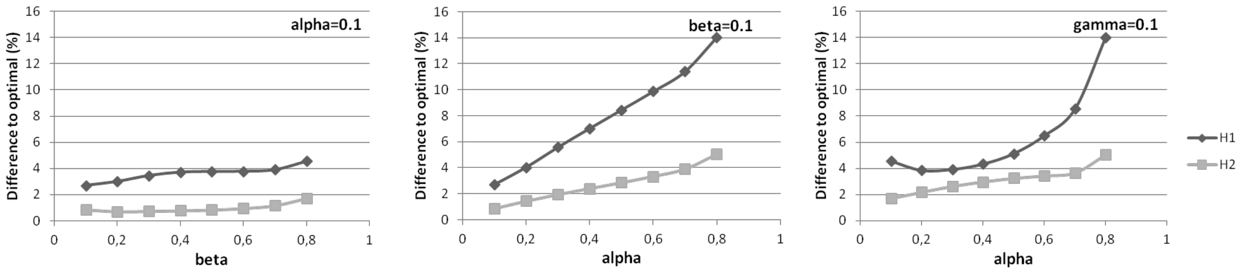
We observe that H2 always performs better than H1. However, when α is small, the results are very close to the optimum for both heuristics (less than 5% of deviation for H1 and 2% for H2). It means that both heuristics perform well when the inbound truck schedule penalties do not weight much in the objective function.

The results are insensitive to β, the parameter weighting the outbound truck schedule penalties. It is the consequence of the fact that both heuristics focus primarily on the performance of the outbound truck schedule: H2 fixes the outbound schedule, while H1 fixes the inbound schedule subject to the synchronization of the inbound and outbound plans.

H2 is less sensitive than H1 to the changes in parameters α, β and γ, and performs quite well compared to (IP*): the figure 4 shows less than 6% of deviation for any combination of α, β and γ. Therefore, H2 can be used to solve any instance of reasonable size. However, for the big instances with small α, H1 may be more interesting to use since its execution time is shorter, and the results are not much deteriorated.

V. Conclusion

This article considers a planning problem for the internal operations (namely truck presence and pallet moves) of a crossdocking platform. The objective is to minimize both the utilization of the storage and the dissatisfaction of the transportation providers. The problem is modeled with an integer program. The tests carried out with generated instances show that the computation time is too long to be used by platform managers on a daily basis, thus two heuristics are also proposed and tested. The observations of the conditions in which the different heuristics perform best can help refining the decision-aid tool.



(a) (b) (c)

Figure 4: Difference to optimal of the heuristics H1 and H2, when varying α, β and γ

The perspectives are numerous regarding possible extensions of this work. For the time being we consider infinite storage capacity; one extension is to limit the storage capacity. In order to help the manager to create feasible employee schedules, the model should also aim at smoothing the workload balance through the day. It would be interesting to have a flexible door allocation, where doors can change their "inbound" and "outbound" roles during the day, depending on the activity level. The distance or congestion could also be taken into account in the doors assignment process; for instance the model proposed by Yu et al. [12] or Bozer and Carlo [13] could be connected to our model.

Last, but not least, the proposed model assumes that the input data is fully known. To be closer to the actual situations encountered daily in cross dock management, it should be possible to take decisions under uncertainties, especially regarding the contents of the trucks. The proposed schedule should also be as robust as possible to unexpected changes in truck arrival times.

Acknowledgment

This work has been supported by the Rhône-Alpes Region.

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

References

[1] Y. Li, A. Lim, and B. Rodrigues, "Crossdocking – JIT scheduling with time windows," *Journal of the Operational Research Society*, vol. 55, no. 12, pp. 1342–1351, 2004.

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

[3] U. Apte and S. Viswanathan, "Effective cross docking for improving distribution efficiencies," *International Journal of Logistics Research and Applications*, vol. 3, no. 3, pp. 291–302, 2000.

[4] J. Gu, M. Goetschalckx, and L. McGinnis, "Research on warehouse operation: A comprehensive review," *European Journal of Operational Research*, vol. 177, no. 1, pp. 1–21, 2007.

[5] N. Boysen and M. Fliedner, "Cross dock scheduling: Classification, literature review and research agenda," *Omega*, vol. 38, no. 6, pp. 413–422, 2010.

[6] G. Alpan, R. Larbi, and B. Penz, "A bounded dynamic programming approach to schedule operations in a cross docking platform," *Computers & Industrial Engineering*, vol. 60, no. 3, pp. 385–396, 2011.

[7] N. Boysen, M. Fliedner, and A. Scholl, "Scheduling inbound and outbound trucks at cross docking terminals," *OR Spectrum*, vol. 32, pp. 135–161, 2010.

[8] N. Boysen, "Truck scheduling at zero-inventory cross docking terminals," *Computers & Operations Research*, vol. 37, no. 1, pp. 32–41, 2010.

[9] A. Lim, Z. Miao, B. Rodrigues, and Z. Xu, "Transshipment through crossdocks with inventory and time windows," *Naval Research Logistics*, vol. 52, no. 8, pp. 724–733, 2005.

[10] A. Lim, H. Ma, and Z. Miao, "Truck dock assignment problem with operational time constraint within crossdocks," in *Advances in Applied Artificial Intelligence*, ser. Lecture Notes in Computer Science, 2006, vol. 4031, pp. 262–271.

[11] Z. Miao, A. Lim, and H. Ma, "Truck dock assignment problem with operational time constraint within crossdocks," *European Journal of Operational Research*, vol. 192, no. 1, pp. 105–115, 2009.

[12] V. Yu, D. Sharma, and K. Murty, "Door allocations to origins and destinations at less-than-truckload trucking terminals," *Journal of Industrial and Systems Engineering*, vol. 2, no. 1, pp. 1–15, 2008.

[13] Y. A. Bozer and H. Carlo, "Optimizing inbound and outbound door assignments in less-than-truckload crossdocks," *IIE Transactions*, vol. 40, no. 11, pp. 1007–1018, 2008.