



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Using flexible crossdock schedules to face truck arrivals uncertainty

Quentin Fabry^{1,2}, Lotte Berghman², and Cyril Briand¹

¹ LAAS-CNRS, Université de Toulouse, UPS, Toulouse, France
{q.fabry,briand}@laas.fr

² TBS Business School, 1 Place Alfonse Jourdain, 31000 Toulouse, France
l.berghman@tbs-education.fr

Abstract. This paper discusses the use of robust schedules for facing uncertainty arising in crossdock truck scheduling problems. Groups of permutable operations bring sequential flexibility in the schedule, allowing to deal with late or early truck arrivals as the total order and the starting times can be determined in a reactive manner. We focus on the evaluation of a robust schedule considering both the worst maximal lateness of the trucks and the worst total sojourn time of the pallets. We also show how a robust schedule can be evaluated online, taking into account the actual arrival times of the trucks.

Keywords: Crossdocking · Scheduling · Robustness · Groups of permutable operations

1 Introduction

A crossdock is a logistic platform with a set of doors where trucks from different suppliers can unload their products, which are reorganized inside the crossdock before being reloaded on different trucks, depending on their destination. A truck is assumed to be present on the platform at a specific time-window that specifies its planned arrival and latest departure time. This organisation gives rise to numerous problems that have been classified in [3].

This paper focuses on the particular scheduling problem which aims at determining both the door on which each truck is (un)loaded and the starting times of these activities while respecting the time-windows. We assume the general case where each door can be used for both loading and unloading operations. Pallets of inbound trucks can either be temporarily stored inside the crossdock, or can be immediately transferred to outbound trucks that are already docked. We assume that the assignment of the pallets to the trucks is given and that the workforce and handling capacity allow the correct management of pallets inside the crossdock, avoiding any congestion or delay.

Several objective functions have been considered in literature. Many authors focus on time-related objectives such as makespan, lateness or tardiness minimization (see e.g. [4]). Others try to keep the workload (amount of pallets in stock) as low as possible. In this work, we minimize the maximal lateness and

we combine both types of objective functions by minimizing the sum of the sojourn times of the pallets. Indeed, a low sojourn time tends to guarantee a high turnover, which is a common objective for crossdock managers [2].

In practice, a good baseline schedule is no guarantee to be efficient because one has to face many perturbations, which can highly influence the quality and the realized turnover and even make the solution infeasible (with respect to the latest departure times of the trucks). There are numerous uncertainties that can occur in a crossdock, for example because of dock incidents, missing workers and last but not least late arrivals of trucks.

There exist different methods providing robustness to a schedule (see e.g. [6]). A classical one is based on the insertion of idle times able to absorb lateness (see e.g. [5]). By allowing to change the starting times inside the time buffer, the modified solution can stay feasible and efficient. In this work, instead of using temporal flexibility, we choose to use *sequential* flexibility, using *groups* to construct robust schedules. For each door of the crossdock, a sequence of groups of permutable operations (i.e. trucks) is defined but the order of the operations inside a group is undetermined a priori. Avoiding to fix the order inside a group allows to adapt the schedule in a reactive manner, according to the real truck arrival times, while controlling the schedule performance.

This paper first states the considered scheduling problem. Then, assuming a predetermined partition of the trucks into groups, it focuses on the evaluation of a robust schedule. We also explain how, during the execution of the planning, the groups can be advantageously used in a reactive manner depending on the actual arrival times of the trucks.

2 Problem statement

This paper focuses on the so-called Crossdock Truck Scheduling Problem (CTSP). A set I of inbound trucks and a set O of outbound trucks has to be scheduled on a crossdock having n doors. Each inbound (resp. outbound) truck $i \in I$ ($o \in O$) has a processing time p_i (p_o) assumed proportional to the number of pallets to (un)load, a release date or planned arrival time r_i (r_o) and a due date or agreed latest departure time d_i (d_o). For each truck pair (i, o) , w_{io} defines the number of pallets going from truck i to truck o . If $w_{io} > 0$, o cannot be loaded before i begins to be unloaded (start-start precedence constraint). In this case, we say that trucks i and o are *connected*. As trucks might arrive late, interval $\Omega_u = [r_u, \bar{r}_u]$ defines the uncertainty domain of the release date of truck u (the probability being assumed uniformly distributed on the interval). We further refer to r as a particular arrival time scenario, i.e. a vector which specifies for each truck u its realized arrival time $r_u \in \Omega_u$. Decision variables s_i and s_o model the starting time of the unloading and loading activities, respectively.

In order to face truck arrival uncertainties, we aim at providing a schedule offering sequential flexibility using the well-known concept of groups of permutable operations [1]. We define a *group* g as a subset of trucks, assigned to the same door, that can be sequenced in any order. We refer to $g(u)$ as the group con-

taining truck u . Groups can be formed from only inbound trucks, only outbound trucks, or a mix of both as far as they are not connected. Indeed, for any pair $\{i, o\}$ of trucks in $I \times O$, if $w_{io} > 0$, i.e. truck i has pallets that need to be loaded on truck o , we assume that $g(i) \neq g(o)$ as these trucks are not permutable (truck i can not be scheduled after truck o as there is a precedence constraint from i to o).

On each door k , the groups are sequenced and form a *group sequence* G_k . G_k is an ordered set of groups, assigned to a same door k . The group sequences of the different doors interact with each other due to the start-start precedence constraints between connected inbound and outbound trucks. We refer to $\mathcal{G} = \{G_1, \dots, G_n\}$ as a complete flexible schedule, which is composed of n group sequences, one for each door. In addition, $\theta_{\mathcal{G}}$ refers to a total order of the trucks, which is an extension of \mathcal{G} (i.e. $\theta_{\mathcal{G}}$ contains n total orderings of the trucks assigned to the same door, respecting the group sequences of \mathcal{G}).

Given a scenario of the arrival times r and a total order $\theta_{\mathcal{G}}$, one can compute the sojourn time ψ as follows:

$$\psi(r, \theta_{\mathcal{G}}) = \sum w_{io}(s_o - s_i) \quad (1)$$

where s_i and s_o are feasible start times under scenario r and given a total order $\theta_{\mathcal{G}}$. Assuming that the length of a time unit corresponds to the time required to (un)load a single pallet and that $\sum p_i = \sum p_o$, ψ also equals $\sum p_o s_o - \sum p_i s_i$. When r and $\theta_{\mathcal{G}}$ are known, note that the minimum sojourn time ψ^* can be computed in polynomial time using for instance the simplex algorithm. We will detail in Section 3 how we combine the maximum lateness and the total sojourn time in a multi-criteria approach.

Let's consider an example with 2 doors and 5 trucks: $I = \{1, 2\}$ and $O = \{3, 4, 5\}$. The arrival times windows are $\Omega_1 = [\underline{r}_1, \overline{r}_1] = [0, 5]$, $\Omega_2 = [6, 11]$, $\Omega_3 = [0, 5]$, $\Omega_4 = [6, 11]$, and $\Omega_5 = [2, 7]$. The due dates are $d_1 = 15$, $d_2 = 17$, $d_3 = 20$, $d_4 = 15$, and $d_5 = 18$. Trucks 1 and 2 carry 5 and 4 pallets, respectively. Trucks 3, 4, and 5, receive 2, 2, and 5 pallets, respectively, assuming that: $w_{13} = 2$, $w_{14} = 2$, $w_{15} = 1$, $w_{23} = 0$, $w_{24} = 0$, and $w_{25} = 4$ (truck 2 only deserves truck 5). On this example with 2 doors, a flexible schedule $\mathcal{G} = \{G_1, G_2\}$ is considered. $G_1 = \{1\} \prec \{2, 3\}$ (i.e., the group sequence on the first door has two groups: a first group containing only truck 1, sequenced before a second group composed of trucks 2 and 3). We assume $G_2 = \{4, 5\}$ (i.e. there is a single group containing trucks 4 and 5 on the second door). The schedule represented in the Gantt diagram of Figure 1 minimizes the maximum lateness L_{\max} , assuming scenario \underline{r} and total order $\theta_{\mathcal{G}}^1$ (with the additional precedence constraints $2 \prec 3$ and $4 \prec 5$). Blue (red) tasks correspond to inbound (outbound) trucks. The three groups are materialized with green boxes. Truck time-windows are represented with colored bars up and below the tasks. $L_{\max} = -5$ is the smallest maximal lateness that can be obtained. The total sojourn time is $\psi = 48$, which can be improved if we choose other additional precedence constraints.

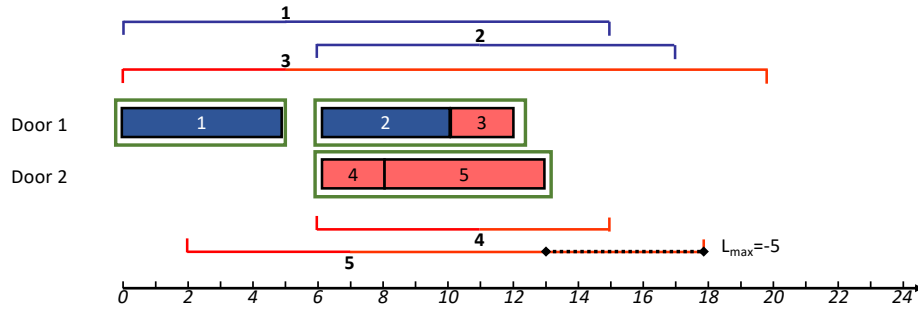


Fig. 1. A schedule minimizing L_{\max} under scenario \underline{r} and total order $\theta_{\mathcal{G}}^1$.

3 Evaluation of the worst lateness

Let us focus first on the lateness. Over all scenarios r and all total orders $\theta_{\mathcal{G}}$, one can compute the worst value of the maximal lateness as

$$\overline{L_{\max}(\mathcal{G})} = \max_{r \in \Omega} \max_{\theta_{\mathcal{G}}} L_{\max}(r, \theta_{\mathcal{G}}). \quad (2)$$

It is obvious that

$$\overline{L_{\max}(\mathcal{G})} = \max_{\theta_{\mathcal{G}}} L_{\max}(\bar{r}, \theta_{\mathcal{G}}) \quad (3)$$

as the lateness can only grow with increasing r . Moreover, considering one particular truck u , the rules defined in [1] allow to determine the particular sequence $\theta_{\mathcal{G}}$ that gives the worst lateness \bar{L}_u .

For the above mentioned example, Figure 2 displays a worst lateness schedule under scenario \bar{r} , which is obtained for the total order $\theta_{\mathcal{G}}^2$ imposing $3 \prec 2$ and $5 \prec 4$. The worst lateness $\overline{L_{\max}(\mathcal{G})} = 4$.

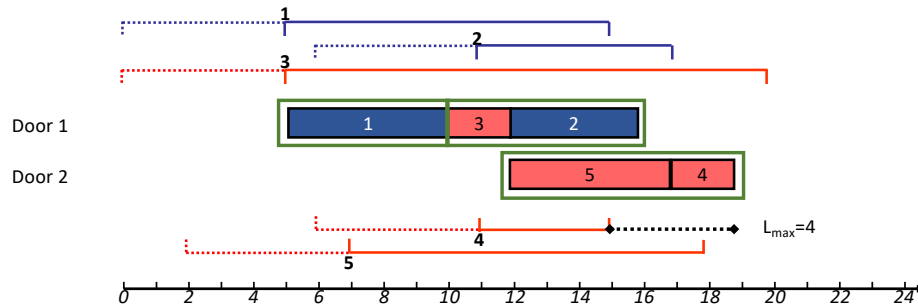


Fig. 2. A schedule giving the worst L_{\max} under scenario \bar{r} and total order $\theta_{\mathcal{G}}^2$.

4 Evaluation of the worst total sojourn time

In the following, we assume that the due dates are extended so that any scenario $\theta_{\mathcal{G}}$ is time-feasible (i.e., $\tilde{d}_u \leftarrow \max\{d_u; d_u + \overline{L_{\max}}\}$). Over all scenarios r and all total orders $\theta_{\mathcal{G}}$, one can now compute the worst total sojourn time as

$$\overline{\psi(\mathcal{G})} = \max_{r \in \Omega} \max_{\theta_{\mathcal{G}}} \psi(r, \theta_{\mathcal{G}}) = \max_{\theta_{\mathcal{G}}} \psi(\bar{r}, \theta_{\mathcal{G}}). \quad (4)$$

Here again,

$$\overline{\psi(\mathcal{G})} = \max_{\theta_{\mathcal{G}}} \psi(\bar{r}, \theta_{\mathcal{G}}), \quad (5)$$

as any relaxation such that $r \leq \bar{r}$ can only give a better objective value ψ . For our small example, $\overline{L_{\max}(\mathcal{G})} = 4$ so the new due dates are $\tilde{d}_1 = 15 + 4 = 19$, $\tilde{d}_2 = 17 + 4 = 21$, $\tilde{d}_3 = 20 + 4 = 24$, $\tilde{d}_4 = 15 + 4 = 19$ and $\tilde{d}_5 = 18 + 4 = 22$.

If we come back to our crossdock instance, the schedule depicted in Figure 2 is also accidentally a worst sojourn time schedule, where $\overline{\psi(\mathcal{G})} = \sum p_o s_o - \sum p_i s_i = (2 * 10) + (2 * 17) + (5 * 12) - (5 * 5) - (4 * 12) = \overline{\psi(\mathcal{G})} = 41$. This worst-case schedule can be compared with the schedule presented in Figure 3, obtained for scenario \underline{r} with total order $\theta_{\mathcal{G}}^3$ defined by $3 \prec 2$ and $4 \prec 5$ and $\psi(\mathcal{G}) = \sum p_o s_o - \sum p_i s_i = (2 * 11) + (2 * 6) + (5 * 13) - (5 * 6) - (4 * 13) = \underline{\psi(\mathcal{G})} = 17$, which is a best sojourn time schedule.

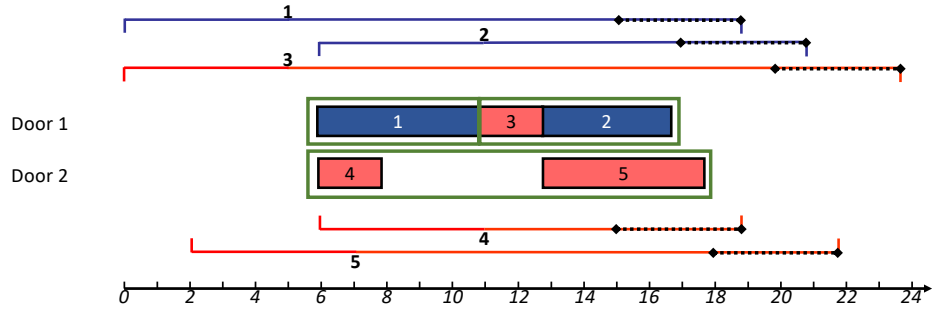


Fig. 3. A schedule giving the best total sojourn time under scenario \underline{r} and total order $\theta_{\mathcal{G}}^3$.

Theorem 1. *Given a general crossdock problem with a flexible schedule \mathcal{G} , finding a total order $\theta_{\mathcal{G}}$ that minimizes $\psi(\mathcal{G})$ is NP-Hard.*

Proof. We reduce the single machine scheduling problem $1|r_j|\sum C_j$, which is known to be NP-Hard, to our problem.

We consider a single machine scheduling problem where the operations are indexed $1, \dots, n$ and we define p_j^s as the processing time and r_j^s as the release date of the operation indexed j . Let us construct a crossdock problem instance

with 2 doors, n inbound trucks indexed i such that $p_i = p_i^s - 1$ and $r_i = r_i^s$, for $i = 1, \dots, n$ and n outbound trucks indexed o such that $p_o = p_o^s$ and $r_o = r_o^s$, for $o = 1, \dots, n$. We assume that the $n = \sum p_o - \sum p_i$ pallets are the initial stock that is available in the crossdock. We assume that inbound truck indexed i delivers its p_i pallets to outbound truck indexed $o = i$ (i.e., $w_{io} = p_i = p_o - 1$ if $o = i$ else $w_{io} = 0$). Moreover, each outbound truck indexed o receives exactly 1 pallet from the initial stock. The flexible schedule $\mathcal{G} = \{G_1, G_2\}$ is such that G_1 contains one group in which we can find all inbound trucks, while G_2 contains one group in which we can find all outbound trucks. The sojourn time can be expressed as

$$\psi(\mathcal{G}) = \sum_o (s_o - t_0) + \sum_i \sum_o w_{io} (s_o - s_i). \quad (6)$$

The former term corresponds to the cost of the pallets transferred directly from the stock to the outbound trucks where t_0 defines the beginning of the schedule. The latter is the cost of the pallets moved from inbound to outbound trucks. Let us observe that, for any feasible schedule of the outbound trucks on the second door, it is always possible to build up a similar schedule on the first door for the inbound trucks by defining $s_i = s_o$ if $i = o$ (because $p_i < p_o$ and $r_i = r_o$). Therefore, $\psi(\mathcal{G})$ reduces to $\sum_o (s_o - t_0)$, which equals $\sum_o C_o - K$ where $K = \sum_o (p_o + t_0)$. Hence, minimizing ψ is equivalent to solving a $1|r_j|\sum C_j$ problem on the second door. \square

From Theorem 1, it results that finding the total order that gives the best or the worst sojourn time is also NP-Hard (as the sub-problem is itself NP-hard). For this reason, we will focus now on upper bound of $\psi(\mathcal{G})$ that can be computed in polynomial time.

To obtain a trivial upper bound on the sojourn time, all inbound trucks i can be scheduled at $s_i = \bar{r}_i$ and all outbound trucks o at $s_o = \tilde{d}_o - p_o$. Such a schedule is not necessary feasible because several trucks may be (un)loaded at the same time on the same door and/or the group sequences might not be respected. Nevertheless, since one can ensure that $s_i \geq \bar{r}_i$ and $s_o \leq \tilde{d}_o - p_o$ in any schedule, an upper bound on the sojourn time is obtained. For the example that was presented earlier, the value would be $\sum p_o s_o - \sum p_i s_i = \sum p_o (\tilde{d}_o - p_o) - \sum p_i \bar{r}_i = (2 * (24 - 2)) + (2 * (19 - 2)) + (5 * (22 - 5)) - (5 * 5) - (4 * 11) = 94$. A tighter upper bound can be calculated by imposing sequences inside each group. The incoming trucks are sequenced by non-decreasing \bar{r}_i and scheduled as early as possible. The outgoing trucks are sequenced by non-decreasing latest departure time \tilde{d}_o and scheduled as late as possible. Note that the resulting schedule might still be infeasible as there can be capacity violations between adjacent groups.

Figure 4 illustrates that sequencing the trucks inside the groups and scheduling them according to the previous rule enforces the total order θ_G^k (already used for the best lateness). This schedule shows an upper bound of $\sum p_o s_o - \sum p_i s_i = (2 * 22) + (2 * 15) + (5 * 17) - (5 * 5) - (4 * 11) = 90$ for $\psi(\mathcal{G})$, which is better than the trivial bound of 94.

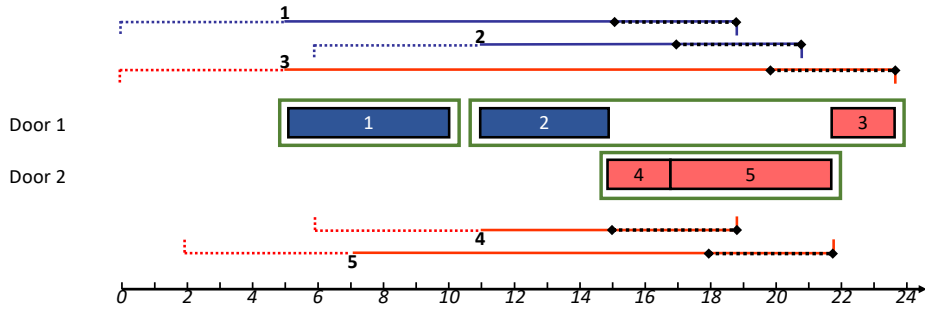


Fig. 4. A schedule giving an upper-bound total sojourn time under scenario r and total order θ_G^1 .

5 Online evaluation of Θ_G

The previous worst-case evaluations of the lateness and the sojourn time are very pessimistic. Assuming a particular online sequencing rule \mathcal{R} for sequencing the trucks in a group under scenario r , one can instead try to evaluate the expected value of the maximal lateness and the total sojourn time. A Monte-Carlo method can be used for this purpose. Figure 5 illustrates this idea for scenario $r = \{1, 9, 0, 11, 4\}$ and the online scheduling rule FIFO (inside each group, the trucks are scheduled as soon as they become available). We obtain $L_{\max}(\mathcal{G}) = 1$ and $\psi(\mathcal{G}) = (2 * 6) + (2 * 14) + (5 * 9) - (5 * 1) - (4 * 9) = 44$.

Note that the previous defined bounds assume an optimal schedule for a given sequence of the trucks inside the group. But this might not be respected by a particular online scheduling rule \mathcal{R} . Here we find for example $\psi(\mathcal{G}) = 44 \geq \bar{\psi}(\mathcal{G}) = 41$.

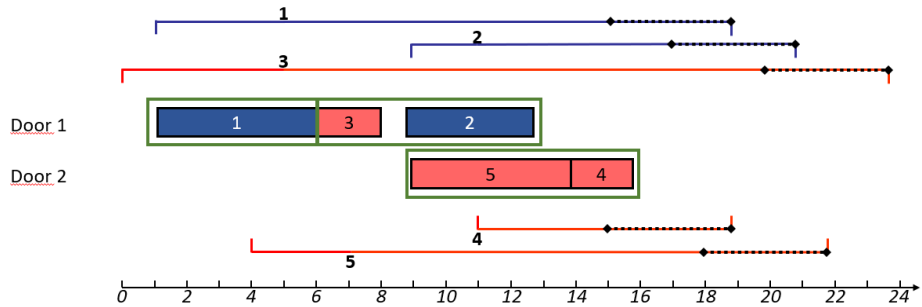


Fig. 5. A online schedule obtained with scenario $r = \{1, 9, 0, 11, 4\}$ and rule FIFO.

6 Conclusion

In this article, we presented some first performance measures to quantify the quality of a flexible schedule on a crossdock defined by a set of group sequences, one for each door. We are able to compute some bounds on the objective value for both the worst maximal lateness and the worst total sojourn time. Future research will concentrate on tightening these bounds, for example by taking into account the start-start precedence constraints that exist between inbound and outbound trucks. Assessing the quality of a flexible schedule is essential to be able to develop algorithms that construct good quality group sequences.

References

1. Artigues, C., Billaut, J.C., Esswein, C.: Maximization of solution flexibility for robust shop scheduling. In: *European Journal of Operational Research*. vol. 165, pp. 314–328. North-Holland (2005)
2. Berghman, L., Briand, C., Leus, R., Lopez, P.: The truck scheduling problem at crossdocking terminals: Exclusive versus mixed mode. In: *ICORES 2015*. pp. 247–253 (2015)
3. Boysen, N., Fliedner, M.: Cross dock scheduling: Classification, literature review and research agenda. *Omega* **38**, 413–422 (2010)
4. Ladier, A.L., Alpan, G.: Crossdock truck scheduling with time windows: earliness, tardiness and storage policies. *Journal of Intelligent Manufacturing* **29**, 569–583 (2018)
5. Lambrechts, O., Demeulemeester, E., Herroelen, W.: Time slack-based techniques for robust project scheduling subject to resource uncertainty. *Annals of Operations Research* **186**, 443–464 (2011)
6. Xi, X., Changchun, L., Yuan, W., Loo Hay, L.: Two-stage conflict robust optimization models for cross-dock truck scheduling problem under uncertainty. *Transportation Research Part E: Logistics and Transportation Review* **144**, 102123 (2020)