Crossdock scheduling under uncertainty using groups of permutable trucks

Q. Fabry[1,2], L. Berghman[2] and C. Briand[1]

[1] LAAS-CNRS, Université de Toulouse, UPS, Toulouse, France - {quentin.fabry;cyril.briand}@laas.fr

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

Mots-clés : *Crossdocking, Robust scheduling, flexibility, groups of permutable operations.*

1 Introduction

A crossdock is a logistic platform where trucks from different suppliers can unload their products, which are reorganized before being reloaded on different trucks, depending on their destination. This organisation gives rise to numerous problems, which are classified by (Boysen and Fliedner 2010). This paper focuses on the particular scheduling problem arising on the crossdock, which aims at determining both on which door each truck is loaded / unloaded and the starting times of those operations. The objective is to minimize the sum of the sojourn time of the pallets inside the crossdock, which favors a good turnover. We are more specifically concerned by the design of robust schedules in order to face the uncertainties related to the real arrivals of trucks that may not exactly match their expected values.

A common way to enforce schedule robustness consists in adding artificial idle time at specific moments so that late truck arrivals can be absorbed, see e.g. (Herroelen and Leus 2004). In that way, the solution can remain feasible as the starting times can be modified within the time buffer. However this simple method can excessively deteriorate the value of the objective function.

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 such that 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 abstract first states the schedule problem considered. Then, assuming the partition of the trucks into the predetermined groups, we focus on the evaluation of the robust schedule both from the lateness and sojourn time viewpoint. 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

We consider 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 $i \in I$ ($o \in O$) have a processing time p_i (p_o) (assumed proportional to the number of pallets to unload / load), a release date (i.e. a planned arrival time) r_i (r_o), a deadline (i.e. a latest departure time) $\overline{d_i}$ ($\overline{d_o}$). 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 has to be loaded after i begins to be unloaded (start-start precedence constraint). As trucks might arrive late, interval $\Omega_u = [\underline{r_u}, \overline{r_u}]$ defines the uncertainty domain of the release date of each truck u (the probability being assumed uniformly distributed on the interval). 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 (**?**, Artigues et al. 2016). We define a *group* g as a subset of trucks, assigned to the same door, that can be sequenced inside the group in any order. We refer to $g(u)$ as the group containing truck u. If $w_{io} > 0$, we assume that $g(i) \neq g(o)$ in order to avoid infeasible schedules.The groups are totally ordered on each door k and form a *group sequence* G_k. The group sequences of the different doors interact with each other due to the start-start precedence constraints between inbound and outbound trucks. We refer to $\mathcal{G} = \{G_1, \ldots, G_n\}$ as a complete flexible schedule, which is composed of n group sequences, one for each door. In addition, $\theta_{\mathcal{G}}$ refers to n particular total orderings of the trucks assigned to the same door, which is an extension of $\mathcal{G}$ (i.e. $\theta_{\mathcal{G}}$ is a particular order of the trucks, 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 : $\psi(r, \theta_{\mathcal{G}}) = \sum w_{io}(s_o - s_i) = \sum p_o s_o - \sum p_i s_i$ where (s_o, s_i) are feasible start times under scenario r and given a total order $\theta_{\mathcal{G}}$. 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. Also note that a combination of r and $\theta_{\mathcal{G}}$ may be infeasible with respect to the truck departure times. This issue is addressed below.

3 Evaluation of $\mathcal{G}$

As some total orders $\theta_{\mathcal{G}}$ can be infeasible with respect to the truck departure times, it is better to considered them as due dates instead of deadlines. In that way, the evaluation of the flexible schedule $\mathcal{G}$ can be made according to the maximal lateness and the total sojourn time. Let us focus first on the lateness. Over all scenarios r and any total order $\theta_{\mathcal{G}}$, one can compute an upper bound on the maximal lateness as $\overline{L_{\max}(\mathcal{G})} = \max_{r \in \Omega} \max_{\theta_{\mathcal{G}}} L_{\max}(r, \theta_{\mathcal{G}})$. It is obvious that $\overline{L_{\max}(\mathcal{G})} = \max_{\theta_{\mathcal{G}}} L_{\max}(\overline{r}, \theta_{\mathcal{G}})$ as the lateness can only grow with r. Moreover, considering one particular truck, it is easy to determine the sequence that gives its worst lateness $\theta_{\mathcal{G}}$ applying rules given in (Artigues et al. 2016).

In the following, we assume that truck deadlines are extended so that any scenario $\theta_{\mathcal{G}}$ is time-feasible (i.e., $d_u \leftarrow \max(d_u, d_u + \overline{L_u})$, where $\overline{L_u}$ is the highest maximal lateness of truck u over all r and all $\theta_{\mathcal{G}}$. Over all scenarios r and any total order $\theta_{\mathcal{G}}$, one can compute the worst total sojourn time as $\overline{\psi(\mathcal{G})} = \max_{r \in \Omega} \max_{\theta_{\mathcal{G}}} \psi^*(r, \theta_{\mathcal{G}}) = \max_{\theta_{\mathcal{G}}} \psi^*(\overline{r}, \theta_{\mathcal{G}})$. Here again, $\overline{\psi(\mathcal{G})} = \max_{\theta_{\mathcal{G}}} \psi^*(\overline{r}, \theta_{\mathcal{G}})$, as any relaxation such that $r \leq \overline{r}$ will give a better ψ^*. Unfortunately, finding a $\theta_{\mathcal{G}}$ that maximizes ψ^* is NP-Hard. We will show during the conference how good-quality upper and lower bound of $\overline{\psi(\mathcal{G})}$ can be computed in polynomial time.

The previous worst-case evaluation is known to be 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 total sojourn time. A Monte-Carlo method can be used for this purpose and will be discussed during the conference.

Références

C. Artigues, J.C. Billaut, A. Cheref, N. Mebarki and Z. Yahouni (2016). Robust Machine Scheduling Based on Group of Permutable Jobs. *In : Robustness Analysis in Decision Aiding, Optimization and Analytics*, Springer, Cham, 191-220.

N. Boysen and M. Fliedner (2010). Cross dock scheduling : Classification, literature review and research agenda, *Omega*, 413-422.

W. Herroelen and R. Leus (2005). On the construction of stable project baseline schedules, *European Journal of Operational Research*, 156(3), 550–565.