Container transshipment at rail yards: A two-way bounded dynamic programming approach

Alena Otto · Xiyu Li · Erwin Pesch

1 Introduction

The share of the freight rail transportation is decreasing because of its lack of competitive advantage. Long processing and waiting times at rail yards delay the delivery of customer goods. The actual average speed of freight trains is estimated to be 10-18 km/h [6, 3] and almost half of freight trains reach their destination with more than a 30 minutes delay [2]. Substantial financial savings and less train delays are likely to be achieved with more efficient yard operations. In this paper, we consider the assignment and scheduling of container moves to cranes at container transshipment yards, we setup a model and design an exact solution procedure.

Figure 1 provides a schematic illustration of a transshipment yard consisting of multiple tracks, a parking and a driving lanes for trucks and a storage. Observe that trucks can always park at the most convenient positions to receive or to deliver containers. We refer to an operation of fixing the cranes grip hooks at a container, picking it up, bringing it to its final position and dropping it as a *job*. We call the time required by a crane between execution

Alena Otto
University of Siegen, Department of Management Information Science, Kohlbettstraße 15, D-57068 Siegen, Germany
Tel.: +49 271 740 2986
E-mail: alena.otto@uni-siegen.de

Xiyu Li
University of Siegen, Department of Management Information Science, Kohlbettstraße 15, D-57068 Siegen, Germany
Tel.: +49 271 740 4291
E-mail: xiyu.li@uni-siegen.de

Erwin Pesch
University of Siegen, Department of Management Information Science, Kohlbettstraße 15, D-57068 Siegen, Germany
Tel.: +49 271 740 2420
E-mail: erwin.pesch@uni-siegen.de

Proceedings of the 11$^{\text{th}}$ International Confenference on Practice and Theory of Automated Timetabling (PATAT-2016) – Udine, Italy, August 23–26, 2016

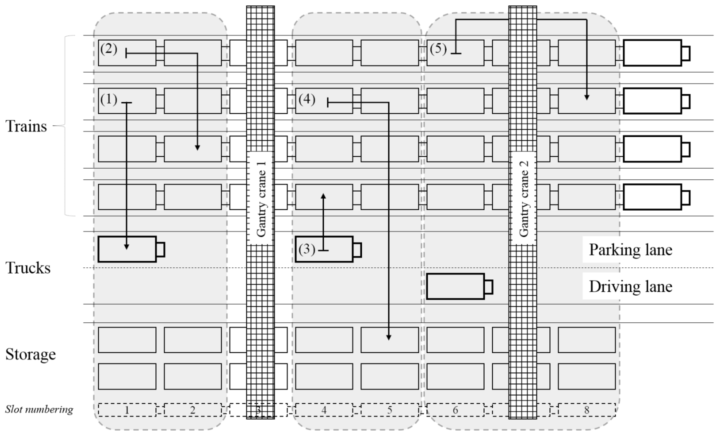

Fig. 1: Illustration of a mixed container transshipment yard

of two consecutive jobs i, j as the *setup time*. As a servicing promise to the customers' trucks, we can specify the *earliest finishing time* and the *deadline* for each job.

We assume that each crane operates in its fixed zone of operation, called *crane zone*. We also assume that each container move is performed by a single crane within its crane zone. The latter assumption presupposes that the picking and dropping positions of containers are relatively close to each other, which is relevant for gateway yards [4].

In order to describe crane zones, we partition the transshipment yard into *blocks* of slots $K = \{1, \dots, |K|\}$, each block has to be assigned to a single crane in order to avoid container transshipments between crane zones. For example, there are three blocks $K = \{1, 2, 3\}$ in Figure 1 which contain slots 1 and 2, 4 and 5 as well as 6 to 8, respectively. We number blocks and cranes in the increasing order from left to right according to their position.

The Static Crane Scheduling Problem (SCSP) considers yard partitioning and job scheduling problems simultaneously and can be described as follows:

Input parameters:
- a set of jobs $J = \{1, 2, \dots, n\}$,
- processing times p_j, earliest finishing times τ_j^e and deadlines τ_j^d, $j \in J$,
- setup times σ_{ij} between jobs $i, j \in J$,
- a set of precedence relations $E = \{(i, j) | i$ has to precede $j, i, j \in J\}$,
- a set of blocks $K = \{1, 2, \dots, |K|\}$,
- binary parameters $a_{kj} : a_{kj} = 1$ if job j takes place in block $k, a_{kj} =$ 0 otherwise,
- a set of cranes $V = \{1, 2, \dots, R\}$.

Yard partitioning subproblem: Determine yard partition into crane zones, i.e. a partition of blocks K into R disjoint subsets K^r in which crane r operates so that

- for each crane $r \in V$ and for each $k < k' < k''$, if $k, k'' \in K^r$, then $k' \in K^r$ (there is no overlap between zones),
- for each crane $r \in V$, we denote the set of jobs performed by this crane as $J^r = \{j \in J | \sum_{k \in K^r} a_{kj} \geq 1\}$.

Job scheduling subproblems for each crane $r \in V$: Determine finishing times $\tau_j^s, j \in J^r$ so that
- $\tau_j^e \leq \tau_j^s \leq \tau_j^d$ (time windows are respected),
- if $\tau_i^s \leq \tau_j^s$, then $\tau_i^s + \sigma_{ij} + p_j \leq \tau_j^s$ (setup and processing times are respected),
- if $(i, j) \in E, i, j \in J^r$, then $\tau_i^s + \sigma_{ij} + p_j \leq \tau_j^s$ (precedence relations are satisfied).

Objective function: Minimize the makespan $C^{max} = \max_{j \in J}\{\tau_j^s\}$.

Observe that each of R job scheduling subproblems is a generalization of the asymmetric traveling salesman problem with the makespan objective function, which is sometimes referred to as the *minimum completion time problem* [cf. 5], and is NP-hard in the strong sense. We denote a job scheduling subproblem of a single crane operating in crane zone $K^r = \{k, \dots, k'\}$ as $\Pi(k, k')$.

We refer to [1] for a literature review on operational planning at transshipment yards and to [4] for the relation of SCSP to other crane scheduling problems.

2 Two-way bounded dynamic programming approach

To solve SCSP, we have designed an exact solution procedure *the two-way bounded dynamic programming approach (TBDP)*. We implicitly generate all possible yard partitions by formulating a state graph. We examine upper and lower bounds of the partial solutions to calculate a *global upper bound (GUB)* and a *global lower bound (GLB)* and to determine *bottleneck* job scheduling subproblems $\Pi(k, k')$ that are pivotal for the current values of the GUB and the GLB. After an improved calculation of upper and lower bounds of partial solutions that include the bottleneck job scheduling subproblems, we can update the values of the GUB and the GLB to decrease the gap between them. We proceed iterations, i.e. to update the set of bottleneck job scheduling subproblems, to improve bounds of the respective partial solutions and to update the values of the GUB and the GLB, until the GUB equals to the GLB and we have proven the optimality of the best found feasible solution. In this way, TBDP has to solve only a few bottleneck job scheduling subproblems to optimality.

For our computational experiments we randomly generate instances that mimic typical German transshipment yards (see Table 1, [for details, see 4]). We generate five data sets with 25 instances each, with 2 to 4 cranes and 2 to 4 tracks, e.g. in Table 1 data set *C4T2* denotes instances with 4 cranes and 2 tracks. The generated instances contain 14 to 75 jobs. TBDP takes 30 seconds per instance on average, and maximally 23 minutes. Compared to common planning routines of yard partitioning into equal crane zones and job scheduling according to the earliest deadline rule, TBDP is able to reduce the resulting makespan by 18% to 28% on average. Standard solver IBM ILOG CPLEX was able to find optimal solutions only for 9 of 125 instances within one hour of run time.

Table 1: Performance of TBDP

Data set	*C4T2*	*C4T3*	*C4T4*	*C3T2*	*C2T2*
TBDP					
Avg. CPU time, seconds	2	10	113	4	27
Avg. rel. improvement to common planning routines, %	23	28	27	22	18
IBM ILOG CPLEX 12.6, 1 hour time limit					
# of cases, where optimum was found	3	0	0	2	4

References

1. Boysen, N., Fliedner, M., Jaehn, F., Pesch, E.: A survey on container processing in railway yards. Transportation Science **45**, 312–329 (2013)
2. EU Commission: Mitteilung der Kommission an den Rat und das Europäische Parlament: Über die Überwachung der Entwicklung des Schienenverkehrsmarkts (2007)
3. Kille, C., Schmidt, N.: Wirtschaftliche Rahmenbedingungen des Güterverkehrs. Studie zum Vergleich der Verkehrsträger im Rahmen des Logistikprozesses in Deutschland. Fraunhofer IRB Verlag (2008)
4. Otto, A., Li, X., Pesch, E.: Two-way bounded dynamic programming approach for operations planning in transshipment yards. Transportation Science (in press)
5. Tilk, C., Irnich, S.: Dynamic programming for the minimum tour duration problem. Discussion Paper Series of Gutenberg School of Management and Economics **1408** (2014)
6. Verband der Automobilindustrie: Auto Jahresbericht 2006 (2006)