TITLE: An agent-based approach to schedule crane operations in rail-rail transshipment terminals
AUTHORS: Heshmati, S ; Kokkinogenis, Z ; Rossetti, RJF ; Maria Antónia Carravilla ; Oliveira, JF ;
SOURCE: Lecture Notes in Economics and Mathematical Systems, VOLUME: 682, PAGES: 91-97, PUBLISHED: 2016
DOI: http://dx.doi.org/10.1007/978-3-319-20430-7_12

An Agent-based Approach to Schedule Crane Operations in Rail-Rail Transshipment Terminals

Sam Heshmati *, Zafeiris Kokkinogenis †, Rosaldo J. F. Rossetti†, Maria Antónia Carravilla*, José Fernando Oliveira*

Abstract Rail-rail transshipment terminals (RRTT) play the hub role in hub-and-spoke rail networks. In this type of terminals, containers are transshipped among freight trains by gantry cranes. This study aims scheduling crane operations with an agent-based simulation approach. The decisions to make include: the positions of containers on outbound trains, the assignment of container moves to cranes, as overlapping areas for the cranes' operation are considered, the sequence of transshipments. The goal is to minimize the total transshipment time. The contribution of this paper is an agent-based approach to the crane operations scheduling problem in a RRTT framework. This approach is validated on a set of instances taken from the literature and the results compared against a variable neighbourhood descent algorithm.

Key words: Railways system; Transshipment yards; Scheduling crane operations; Agent-based simulation.

1 Introduction

A modern rail-rail transshipment terminal (RRTT) consists of a number of rail tracks, where freight trains are positioned in bundles to be served (one train per track), and rail-mounted gantry cranes move containers between different freight trains, without exchanging the wagons. This is a complex system with sophisticated operations which requires good schedules and collaboration of resources.

A RRTT has to be planned to guarantee an acceptable level of service in terms of trains' waiting time. One of the critical factors in the transshipment of containers

* INESC TEC, Faculty of Engineering, University of Porto, 4200-465, Porto, Portugal,
e-mail: prdeig1103803@fe.up.pt, mac@fe.up.pt, jfo@fe.up.pt ·
† Faculty of Engineering, University of Porto, 4200-465, Porto, Portugal,
e-mail: pro08017@fe.up.pt, rossetti@fe.up.pt

among trains in terminals is how to best use the gantry crane. In this study we aim at analyzing container transshipment processes in RRTTs, from an operational point of view. This involves decisions on the position of containers on outbound trains, on the assignment of container moves to cranes, as predefined overlapping areas for the cranes are considered, and on sequencing the transshipments. The goal is minimizing the total transshipment time.

For operational planning and control of RRTTs it is important to be able to deal with uncertainty (e.g. unscheduled container shipping requests). The use of simulation allows to accurately represent the movements of the crane, containers, and the trains in each terminal, while dealing with uncertainty. However, the complexity of this type of terminals makes it hard to model the problem using standard simulation approaches. An alternative that in the last years has received a lot of attention is the multi-agent system (MAS) approach. As in other application areas, the use of agent-based simulation (ABS) is growing considerably to simulate and analyze various transportation problems, from traffic flow to seaport container terminals. However, although ABS has been applied in transportation problems, literature does not report significant work with RRTT problems.

In this study we introduce an agent-based approach for the container transshipment problem in RRTTs, where intelligent crane agents decide and plan their own schedule. The use of the agent concept allows us to come up with solutions in highly dynamic scenarios (such as the unscheduled shipping of containers), while fixed dispatching rules do not have the same capability. The objective is to assign positions for containers on outbound trains and determine the sequence of container movements for each crane, such that all containers are positioned on the appropriate train or on the yard.

The remainder of this paper is structured as follows. In the next section the existing literature on rail-rail transshipment yards is reviewed. Then, in Section 3 the overview of the problem and its characteristics is presented. Afterwards, the simulation model is described in Section 4. Section 5 contains the computational results, where the performance of our simulation model is tested and discussed. Finally, the conclusion and opportunities for future work are explained in Section 6.

2 Literature Review

The RRTTs are described as an essential part of a hub-and-spoke architecture and an emerging technology in railway systems (Macharis and Bontekoning 2004). Several studies in the literature claim that the global rail-rail transshipment problem is too complicated to be monolithically solved. Based on studies in the literature (Boysen et al., 2012a and Souffriau, 2009) we can decompose the rail-rail transshipment problem into seven sub-problems:

1. Schedule the service slots of trains
2. Assign a destination to each train
3. Assign each train to a parking position

4. Determine container positions on outbound trains (load plan)
5. Assign container moves to cranes
6. Schedule the shuttle cars in the sorter
7. Determine the sequence of container moves per crane

In the literature, only some of the sub-problems of the rail-rail transshipment problem were approached. The first decision problem is to assign each train to a bundle. This sub-problem was studied by Boysen et al. (2011), who formulated a basic transshipment yard scheduling problem that minimizes a weighted objective function, which considers split moves between trains that are assigned to different bundles and the number of revisits by trains. The study was extended in Boysen et al. (2012b), with an additional objective function that minimizes the number of containers that can not be transshipped to the assigned trains in time. In Kellner et al. (2012), sub-problem (3) was tackled. The problem assigns a track to each train of a given bundle and additionally decides on each train's longitudinal parking position along the yard, in order to minimize container moves. Boysen and Fliedner (2010), considered static crane areas when assigning container moves to cranes. The study aimed at minimizing the makespan of train processing. A dynamic programming procedure was used for solving the problem and the solutions were tested against typical real-world policies.

Alicke (2002) tackled sub-problems (5), (6), and (7). The overall problem was modeled as a constraint satisfaction problem. Another study considering several sub-problems is Souffriau et al. (2009), which jointly assigns the destination to trains (2), determines the load plan (4), and the sequence of the transshipment (7). The destination was assigned to trains based on estimates of the minimum transversal length of the transshipments. The load plan was hence determined by minimizing the transportation cost of container moves using a mathematical programming model. The crane schedule, which distributes container moves among cranes and sequences moves per crane, was solved by a variable neighborhood descent algorithm.

Despite the valuable contributions of the above mentioned studies, questions regarding the integration of the above problems into a holistic procedure remained untackled. In this study we are aiming at the integrated resolution of three sub-problems: (4) "determine container positions on outbound trains", (5) "Assign container moves to cranes", and (7) "determine the sequence of container moves per crane".

3 Problem Description

This section describes a typical transshipment yard and the three constitutive sub-problems of scheduling crane operations (SCO). A transshipment yard consists of a number of parallel tracks, gantry cranes, and a quay with ground vehicles to move the containers on the quay. Rail-mounted gantry cranes move containers between different freight trains, avoiding the need for exchanging the wagons. The trains (one per track) arrive in bundles to the yard, to be served simultaneously, and the destination of each train has already been assigned. A train is composed of a number of wagons, which carry containers. The (several) gantry cranes process the container moves in parallel. The cranes have areas assigned by static or dynamic policies (section 3.3).

The SCO is an operational problem that has to be solved for every bundle of trains arriving at a transshipment yard. It includes three sub-problems of a typical rail-rail transshipment problem. First, the SCO assigns each container a position on the outbound train with the proper destination in such a way that the overall containers' movement distance is minimized. Next, it assigns container moves to cranes. This can be executed under a static or dynamic policy, when managing crane influence areas. Finally, the schedule of container moves per crane is determined.

3.1 Positions of containers on outbound trains

The objective of this sub-problem is to determine the load pattern in such a way that the total container movement is minimized. In this problem the tracks and quay are characterized by a Y coordinate. A train is composed by a number of wagons of different types, which can hold more than one container. Associated to the wagon type is the length and the configuration of the wagon. The configuration of a wagon is a set of container position slots, each of them having a length and an X coordinate. The X coordinate of the first position slot of a train is 0 and it increases cumulatively with the length of the following position slots.

Containers are currently located on specific wagons of the inbound train. Each one of them has a destination and a type. Associated to the container type is its length. The width is the same for all containers and therefore can be ignored for SCO planning purposes. A container can be placed on a position slot if the position slot is free and its length is equal to the container's length.

A transshipment starts from the position slot where the container is currently located on the inbound train and ends at the position slot of the outbound train to which the container will have to be moved. The Y coordinate of the final position slot is known (because of the containers' predefined destination/outbound train). This sub-problem finds the final position slot in terms of the X coordinate.

3.2 Assign container moves to cranes

The objective is to split the overall workload evenly among the cranes in order to accelerate train processing. Direct transshipment is a type of transshipment when the initial and the final position slots of a transshipment are in the working area of one crane. Whenever the final position slot of a transshipment is in the area of another crane, the first crane has to pick up the container and to put it on the ground vehicle on the quay. The ground vehicle carries the container along the yard to the working area of the second crane; afterwards the second crane finishes the transshipment. This type of transshipment is called indirect transshipment. It is better to have direct transshipments than indirect transshipments because of the cost of extra pick-ups and drops.

The cranes can operate in disjoint or in overlapping areas. This approach assumes assignment of transshipments to cranes considering an overlapping area. The assignment should be done dynamically depending on the utilization of the cranes and it has to be considered in the simulation model. By having overlapping working areas for cranes the number of indirect transshipments will be reduced but it has to be ensured that the cranes do not collide. This is done by blocking the overlapping area, for the other crane, when a crane enters it. Large overlapping areas may lead to high idle times of the crane which is not operating in the overlapping area.

3.3 Determine the sequence of container moves per crane

Finally, it is necessary to determine the sequence of container moves per crane. The resulting problem is similar in structure to a sequential ordering problem (Montemanni et al., 2013). One important aspect is that crane moves are asymmetric in distance. A transshipment job can be carried out only when the destination position is free. In this study we are trying to avoid potential deadlocks (which happens when the crane needs to directly swap the place of two containers) by determining the final position of a transshipment only when the move is going to be executed.

4 Simulation Model

To analyze the effectiveness of RRTTs, this study focuses on gantry crane operations. The model is parameterized in order to simulate one service slot of a terminal located near Antwerp, Belgium. The yard consists of eight tracks, two cranes with an overlapping influence area, ground vehicles, and a quay. The trains are placed on given tracks and each train has a number of wagons of several types. Position slots are generated for each wagon based on the wagon type. Each position slot can hold one container with the same length of the slot. The working area of each crane agent is 60% of the length of the longest train, resulting in a 20% overlap of influence areas between the two cranes.

The model has two rules, called “set target rules”, for agent cranes to decide which transshipments to choose to process. Each time an agent crane is going to start a transshipment job it chooses randomly one of these rules to apply. The **First rule** is to pick up the nearest unprocessed container, and the **Second rule** is to pick up the nearest unprocessed container on the train with the highest number of unprocessed containers. Whenever a crane cannot find a free position slot on the destination train, inside its own influence area, it will drop the container on the ground vehicle on the quay. The ground vehicle moves the container along the yard to the area of the other crane and the second crane will process the container. If there is also no free position slot in that part of the train, the container will remain on the yard, waiting for the next train with the same destination. The model assumes that the lateral and longitudinal speeds of the cranes are equal, and in each time unit of the simulation the crane travels one distance unit. Thus the traveling time can indeed be replaced by the Chebychev distance.

Our model is implemented in Netlogo (Wilensky, 1999), an agent-based simulation platform and programming language. The cranes and ground vehicles are implemented as agents. The cranes can move over the trains on the yard both in the X and Y directions, and the ground vehicles can move on the quay along the X axis. When a crane agent calls for a ground vehicle the model creates one ground vehicle agent on the quay with the X coordinate of the caller crane. To prevent the collision between cranes, in the overlapping area, the overlapping area is blocked whenever a crane enters it. During the overlapping area blockade, provoked by one crane, the other crane chooses a container to process out of the overlapping area. If a crane blocks the overlapping area while the other crane's target was already set inside the overlapping area, then the other crane stays idle at the border until the overlapping area is freed.

This model implements a discrete event simulation (DES) where every tick (time step) corresponds to one second of real-world time. However, our implementation differs from the standard DES in a number of ways. Firstly, in this model each agent has its own behavior and makes decisions locally. On the contrary, the behavior of the entities in a DES model is determined by the system. Another key difference is that DES is built around networks of priority queues of (time, event) pairs, sorted by time. In this agent-based model there is no concept of queues. Alternatively, there is a control and monitoring action that has to be executed constantly. To setup the terminal, the model generates the terminal, quay, trains, position slots, containers, and cranes based on the input file. The input file includes the following data for each entity:

Trains:track number, length, and number of wagons in each train.

Position slots: train number, X coordinate, and type.

Containers: coordinates, type, and destination.

Our implementation has a main loop that is called at every tick. The simulation terminates when all the existing trains in the yard are served and ready to leave the yard. At each tick, the model asks the cranes that have unprocessed containers in their area to execute. In this "execute" method, if the crane is loaded, it moves toward the final position of the transshipment. If the crane is not loaded, it will set a target by itself and start moving toward the target container. In the following ticks the crane continues moving toward its target container, the pickup method will be executed and the crane becomes a loaded crane. This loop terminates when both cranes have no more unprocessed containers in their working areas.

5 Simulation Results

The simulation model has been validated on a set of problem instances taken from the literature (Souffriau et al., 2009). For this terminal, instances have 8 tracks, 5 destinations and 5 trains, with 20 wagons each train. Two wagon types are considered: 60ft and 85ft. Possible container lengths are 20ft, 30ft and 40ft. The load factor is defined as the total length of the containers divided by the total length of the trains. The following range of load factors is used: $\{0.1, 0.2, \ldots, 0.9\}$ (step of 0.1). There are five instances for each load factor. The results presented below are averaged over these five instances per load factor. The computational experiments were run on a personal computer with an Intel i7-2620M, with a clock speed of 2,70, GHz and 8 GB of RAM.

Following Souffriau et al. (2009), and in order to be possible to compare the two approaches, the results of the simulation model are presented as the improvement over a random solution. Table 1 presents the load factor, the number of containers (# Containers), the number of containers left on the quay (# Containers on yard), the execution time, and the average percentual improvement. As expected these results show that the problem becomes harder to solve when the load factor increases. Moreover, there is a considerable improvement (greater than 70%) for instances with a load factor equal or smaller than 0.6, which then drops markedly until a value of 12% for a load factor of 0.9. The average improvement of the agent-based simulation model is 63%, which is similar to the result obtained by Souffrriau et al. (2009): 62% of improvement.

However the computational time of our approach is significantly lower, when compared to theirs. In Souffrriau et al. (2009) the variable neighbourhood descendent algorithm was run with a number of perturbations ranging from 0 to 10, with increasing computing times. Even if we take the results with 0 perturbations, which only deliver an average improvement of 53.9%, the execution times range from 0.4 seconds, for a load factor of 0.1, to 3278.0 seconds, for a load factor of 0.9 (Intel Xeon with a clock speed of 2.5 GHz and 4 GB of RAM). Even taking into account the different hardware platforms, it is significant that for a load factor of 0.9 our approach is 40 times faster than the VND algorithm in its fastest configuration. If the more time-consuming configuration is taken, this factor grows for a value of approximately 160.

An important performance indicator is also the number of containers left on the quay. In instances with a load factor greater than 0.6 there are indeed containers left behind on the quay, with a maximum value of 14.2 (in average) for instances with a load factor equal to 0.9. This behaviour is common to the approach of Souffrriau et al. (2009), although with lower values: 0.2, 2.8, and 4.0 containers left on the quay, for load factors 0.7, 0.8 and 0.9 respectively.

Table 1 Simulation results*

Load Factor	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
# Containers	29	55.6	83.4	110.6	138.2	168.2	198	237	251.4
# Containers on yard	0	0	0	0	0	0	0.8	6.4	14.20
Execution times (sec)	0.60	0.90	1.34	1.89	2.39	3.13	9.54	32.96	77.70
Average improvement (%)	72.47	77.51	76.99	78.14	75.52	72.91	58.7	39.82	12.00

* The results are averaged over five instances per load factor

6 Conclusions

This paper deals with the scheduling crane operations (SCO) problem in rail-rail transshipment terminals (RRTT). In this combinatorial optimization problem the containers have to be transshipped among trains by multiple cranes. The SCO problem involves three decisions: (1) to define the position of containers on outbound trains, (2) to assign transshioments to cranes, and (3) to sequence transshipments for each crane. The problem is tackled by an agent-based simulation model, in which agent cranes deal with the three decisions and transship the containers, while working with a 20% overlap in their influence areas. A set of problem instances taken from literature is used to validate the model and the results are compared against Souffriau et al. (2009). The model achieved up to 75% improvement, a value similar to Souffriau et al. (2009) results, but in a fraction of the time used by these authors.

Future work includes applying dynamic yard assignment for cranes, where no specific influence area is defined for cranes and agent cranes cooperate and coordinate through negotiation processes to avoid collision. It is expected that this more thorough implementation of the agent-based simulation framework will contribute for the main drawback of the current model that is the number of containers left on the quay, which is greater than the one obtained by Souffriau et al. (2009).

Acknowledgements Project "NORTE-07-0124-FEDER-000057" is financed by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT).

References

1. Alicke, K.: Modeling and optimization of the intermodal terminal Mega Hub. OR Spectrum. **24**, 1–18 (2002)
2. Boysen, N., Fliedner, M., Jaehn, F., Pesch, E.: A survey on container processing in railway yards. Transportation Science. **1655**, 1–18 (2012)
3. Boysen, N., Fliedner, M.: Determining crane areas in intermodal transshipment yards: The yard partition problem. European Journal of Operational Research. **204**, 336–342 (2010)

4. Boysen, N., Jaehn, F., Pesch, E.,: New bounds and algorithms for the transshipment yard scheduling problem. Journal of Scheduling. **15**, 499–511 (2012)
5. Boysen, N., Jaehn, F., Pesch, E.,: Scheduling Freight Trains in Rail-Rail Transshipment Yards. Transportation Science. **45**, 199–211 (2011)
6. Kellner, M., Boysen, N., Fliedner, M.: How to park freight trains on rail-rail transshipment yards: the train location problem. OR Spectrum. **34**, 535–561 (2012)
7. Macharis, C., Bontekoning, Y.: Opportunities for OR in intermodal freight transport research: A review. European Journal of Operational Research. **153**, 400–416 (2012)
8. Montemanni, R., Mojana, M., Di Caro, G., Gambardella, L. M.: A decomposition-based exact approach for the sequential ordering problem. Journal of Applied Operational Research. **5**, 2–13 (2013)
9. Souffriau, W., Vansteenwegen, P., Berghe, G., Oudheusden, D.: Variable Neighbourhood Descent for Planning Crane Operations in a Train Terminal. In: Sörensen, K., Sevaux, M., Habenicht, W., Geiger, M. J. (eds.) Lecture Notes in Economics and Mathematical Systems, pp. 83-98. Springer Berlin Heidelberg (2009)