

## Optimal scheduling and routing of free-range AGVs at large scale automated container terminals

F. CORMAN<sup>1,2</sup>, J. XIN<sup>1</sup>, R. R. NEGENBORN<sup>1</sup>, A. D'ARIANO<sup>3</sup>, M. SAMÀ<sup>3</sup>, A. TOLI<sup>3</sup>, G. LODEWIJKS<sup>1</sup>

RT-DIA-216-2016

Gennaio 2016

(1) Maritime and Transport Technology, Delft University of Technology,  
Mekelweg 2,  
2628CD Delft, The Netherlands

(2) Center for Industrial Management,  
KU Leuven, Belgium

(3) Università degli Studi Roma Tre,  
Via della Vasca Navale, 79  
00146 Roma, Italy

---

*This research is partially supported by the Maritime Project “ShipDrive: A Novel Methodology for Integrated Modeling, Control, and Optimization of Hybrid Ship Systems” (project 13276) of the Dutch Technology Foundation STW, the State key Laboratory of Railway Research, Beijing Jiaotong University, Grant no RCS2015K003 and the China Scholarship Council under Grant 2010628012.*

## ABSTRACT

This work tackles the problem of controlling operations at an automated container terminal. In the context of large supply chains, there is a growing trend for increasing productivity and economic efficiency. New optimization models and algorithms are provided for scheduling and routing equipment that is moving containers in a quay area, loading/unloading ships, transporting them via Automated Guided Vehicles (AGVs) to Automated Stacking Cranes (ASCs), organizing them in stacks. In contrast with the majority of the approaches in the related literature, this work tackles two dynamics of the system, a discrete dynamic, characteristic of the maximization of operations efficiency, by assigning the best AGV and operation time to a set of containers, and a continuous dynamic of the AGV that moves in a geographically limited area. As an assumption, AGVs can follow free range trajectories that minimize the error of the target time and increase the responsiveness of the system. A novel solution framework is proposed in order to tackle the two system dynamics. Various metaheuristic algorithms are tested to solve the problem in a near-optimal way. Computational experiments are presented in order to show the feasibility of the proposed framework on a practical case study, and to assess the performance of advanced scheduling and routing algorithms on numerous system settings.

**Keywords:** Container Terminal Operations; Scheduling; Free-Ranging Routing; Trajectory Planning; Metaheuristics

# 1 Introduction

Global freight transport is a key characteristic that makes our society possible, and has faced an enormous growth over the last decades, due to international commercial trade. Over 60% of worldwide deep-sea cargo is currently transported by containers. A container terminal is the transport hub in freight transport and represents the interface between the modalities of vessel, barge, train and truck, providing flexibility and scalability for covering different geographical areas. Therefore, the performance of container terminals influences freight transport significantly. The increasing amount of containers that arrive and depart with container ships causes much pressure for terminal operators. In 2000, the capacity of a container vessel was typically 6,000-8,000 TEU; in 2013, the number of containers carried by a container vessel can be up to 18,000 TEU. The turnaround time of a container vessel may therefore increase significantly. Hence, the handling capacity of a container terminal must be maximized to reduce the turnaround time of a container vessel.

For the improvement of the performance at terminals, automated container terminals are built, since they can increase the productivity and reduce wage costs in a cost-efficient way. For a general overview of the research questions associated to automated container terminals, the reader is referred to Carlo et al. [2]. The operations of the automated container terminals are complex, in particular when more autonomous vehicles are developed (e.g., Automated Guided Vehicles (AGVs), that can be further moving in a free-ranging fashion). Therefore, improving the performance of automated container terminals is an active research and practical challenge, as is shown in recent literature reviews [1, 2, 20, 21, 22]. Integrated control of all activities in container terminals is becoming more and more important [27].

Models dealing with the problem of increasing efficiency of automated terminal operations can be categorized into different main streams, depending on where the main focus and constraints of the operations are. A stream of research is about *discrete* event scheduling and routing, i.e., mathematical models which assume equipment (cranes, AGV, etc) operating according to the plans, and model conflict and synchronization of equipment in a discrete way. The scheduling and routing problem is a key problem in this context and can be efficiently represented via discrete dynamics of the operations at the whole terminal. A common solution approach is based on modeling the scheduling and routing problem as a hybrid flow shop model with flexible routing. However, this results in a mathematical formulation of the problem with several integer and/or binary variables that is very difficult to be solved in real-time. For such large-scale mathematical programming problems, one open point is to investigate the extent at which a good solution can be found by standard off-the-shelf optimization software in a short time that would allow usage in real operations. To this end, an emerging trend is the usage of metaheuristics (e.g., tabu search, ant colony or variable neighborhood search), which can provide good solution in applicable time [3, 8, 11].

A drawback relates to the amount of assumptions that mathematical optimization approaches need to consider in order to deliver near-optimal solutions quickly, to the point that some operations that have inherently a *continuous* dynamics can hardly be considered. This includes for instance collision avoidance (to ensure safe operations), energy efficiency, and free-ranging routing of AGV (to decrease the total distance traveled). Another main stream of research thus tackles directly those issues, related to the contin-

uous trajectory and autonomous safe movements of AGV. Typically, this is a trajectory planning problem, that can be solved by non-linear optimization techniques, and at the expense of looking for near-optimal solutions to the combinatorial scheduling and routing problem. An example here is the work of Duinkerken et al. [7]. A major drawback of this second stream of research is the quality of the schedules, which are mostly simulated based on dispatching rules, or optimized to a very limited extent. As a result, there is no knowledge on the performance improvement achievable when fixing the value of ordering and routing variables.

Recently, *hybrid* models have been used to address the contrast between those two streams of research in [24, 25, 26, 23]. Such models merge the advantages of discrete-dynamics models (typically, highly optimized scheduling solution), and continuous-dynamics models (i.e., able to model properly real-life dynamics, speed and trajectories of the vehicles). In line with this stream of research, the contribution of this paper goes along the idea of increasing the accuracy of existing hybrid models for automated container terminal operations and developing real-time algorithms for near-optimal scheduling and routing.

We now summarize the main contributions of this paper:

- The large-scale scheduling and routing problem of interacting equipment in an automated container terminal with a predetermined fixed dynamic is modeled as a hybrid flow shop problem, as proposed in [23]. This hybrid flow shop can be formally described as

FH3 (D,  $P_n$ , D) | blocking; seq-dep setup |  $C_{\max}$

according to the typical three-field Graham’s notation (see Ruiz [15] for an overview of the problem and the classification). Three stages are present, out of which the second stage has  $n$  parallel identical machines (P), while the first and the last ones have a single machine (D) assigned per each job. Here, every machine corresponds to a piece of equipment, i.e. a vehicle or a crane, while every job corresponds to a container moving in the container terminal. The objective is the makespan minimization. All machines are blocking (i.e. they can process at most one job at a time [9]), and the operations on the second stage have sequence dependent setup times This particular hybrid flow shop is modelled as a blocking job shop, with multiple machines, each belonging to one of the stages. Each job is being processed by three machines (one in each stage); the parallel machines of the second stage are now considered as routing variable, and the other two stages have non-parallel, assigned machines. Lately, a consistent stream of research on the blocking job shop considered the alternative graph model of Mascis and Pacciarelli [12]. The formulation proposed in this paper also follows the alternative graph model.

- We present a thorough study of multiple advanced algorithmic approaches to solve the problem. Those algorithms are metaheuristics (a variable neighborhood search and a tabu search introduced in [4, 16, 17]) developed in order to decide the assignment of containers to the machines in the second stage (a routing problem). A truncated branch-and-bound algorithm, introduced in [6], is applied to solve the scheduling problem. The tradeoff between computational complexity, instance size, solution quality is precisely assessed.

- The discrete dynamics solution (i.e., a scheduling and routing solution considering assignment of AGV to containers, and order of movements of containers) is adopted in a hierarchical framework, which uses them as boundary values for determining the precise trajectory of AGV. An iterative scheme of updating the trajectory is proposed, which is able to manage congestions on the AGV terminal areas, to deliver AGV trajectory plans according to free-ranging principles, and to take into account capacity constraints at interchange points.
- The vehicle congestion resulting from the large-scale operations of automated, free-ranging AGVs is investigated for a practical case study with parameters of real container terminals, considering in a detailed manner the continuous dynamics of vehicles. We focus on the problem of efficiently solving traffic congestion during operations. In general, increasing the amount of AGV helps keeping a high throughput for the expensive quay cranes, but having just too many AGV available poses serious problems of traffic congestion, that can ultimately reduce throughput of the overall container terminal.
- The impact on capacity is evaluated by simulating operations at a large-scale automated container terminal. We first assess the proposed methodologies on the practical case study, and then perform an extensive quantitative analysis on a set of realistic instances, starting from the practical case study and varying the amount of AGVs and containers. The experiments on the practical case study show that the key performance indicators of the container terminal can be determined and improved when free-ranging AGVs are employed. The experimental results on realistic instances confirm that metaheuristic approaches can successfully solve large-scale instances (keeping a discrete-dynamic perspective) and this can be coupled successfully with hybrid dynamics, while off-the-shelf commercial optimization models have troubles with large-scale instances. Moreover, the meta-heuristics used allow to compute a good quality solution quickly.

The rest of this paper is as follows. Section 2 describes the background of the proposed approach and the main characteristics of the problem. Section 3 then summarizes the models used for representing the discrete event dynamic and continuous time trajectory planning, as well as the algorithms used to solve the combined AGV scheduling, routing and control problem. Section 4 reports on the experiments on the practical case study, and Section 5 describes performance of the approach on further realistic instances. The implications of the computational results are also discussed. Section 6 summarizes the paper contribution and points out further research in the direction of optimizing hybrid operations at large-scale automated container terminals.

## 2 Approach and Problem description

This paper studies an automated container terminal in which the equipment can be controlled automatically without any human intervention. In an automated terminal, there are multiple quay cranes (QCs), multiple automated guided vehicles (AGVs) and multiple automated stacking cranes (ASCs) for transporting containers from a vessel to the stacking area and vice versa. A vessel is hereby typically considered as consisting of several

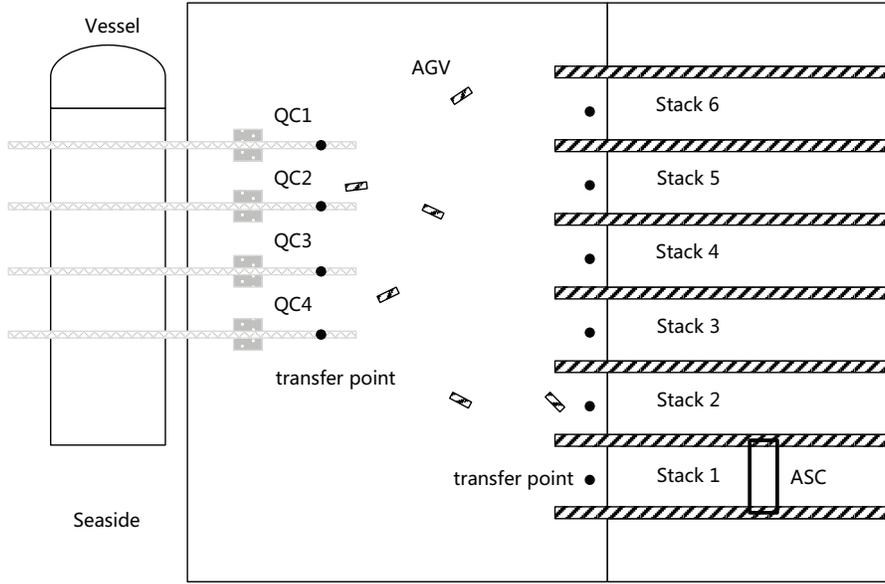


Figure 1: Schematic layout of equipment in an automated container terminal.

bays. Each bay provides storage space for a number of rows which involves several tiers of containers. Each QC considers a particular bay.

In our system, the layout of the equipment is as shown in Figure 1 (see fore details [23]). In a typical unloading cycle, a QC picks up a container from the vessel and then unloads it to an AGV. The AGV moves with the container from the quayside to the stacking area, where a container is unloaded by an ASC. The ASC then transports the container to the position in the stacking area. In a loading cycle these movements are reversed. We call QCs, AGVs, ASCs respectively a stage; all pieces of equipment in each stage are defined as machines or resources. Accelerations, decelerations and the heading (if applicable) of the equipment have to be determined in an optimal way. For safety reasons, collision avoidance must be considered for interaction of AGVs. At the same time, the interaction of AGVs with other types of equipment (i.e., QCs and ASCs) is required for achieving a high terminal throughput. In other words, a key issue is determining the moments in time at which containers are transported from one equipment to the next.

The dynamics of the equipment considered are driven by discrete events when a container is transferred from one equipment to another. Precisely, the discrete dynamics refer to the logical decisions that solve (1) the problem of assigning a container to an AGV (i.e., which particular AGV will be used to transport a particular container) and (2) the relative order between any two operations on any equipment/machine (QCs, AGVs and ASCs). On the other hand, continuous dynamics refer to the position, the speed and the heading (if applicable) of the equipment used by these operations. The dynamics of transporting containers can therefore be represented by the combination of discrete-event dynamics and continuous-time dynamics. The vessel and the stacking area are hereby considered as boundary components that are not actively controlled.

The continuous dynamics of ASC and QC do not depend on the order or the operations of other equipment/machines, while the dynamics of AGV are more complex, as they depend on the interaction and possible collision with the other AGVs that are also moving.

This collision can occur along the same path (if AGV all follow a predefined path along a track); at particular places (if a mesh routing is followed), or at any place of the quayside yard (in case free-ranging AGVs are used).

Due to this strong difference of dynamics, we use the same approach as in [23] to separate the key characteristics of the two problems by means of a decomposition technique, considering the scheduling and routing problem part and the trajectory planning connected by a hierarchical architecture. The overall control architecture is schematically illustrated in Figure 2.

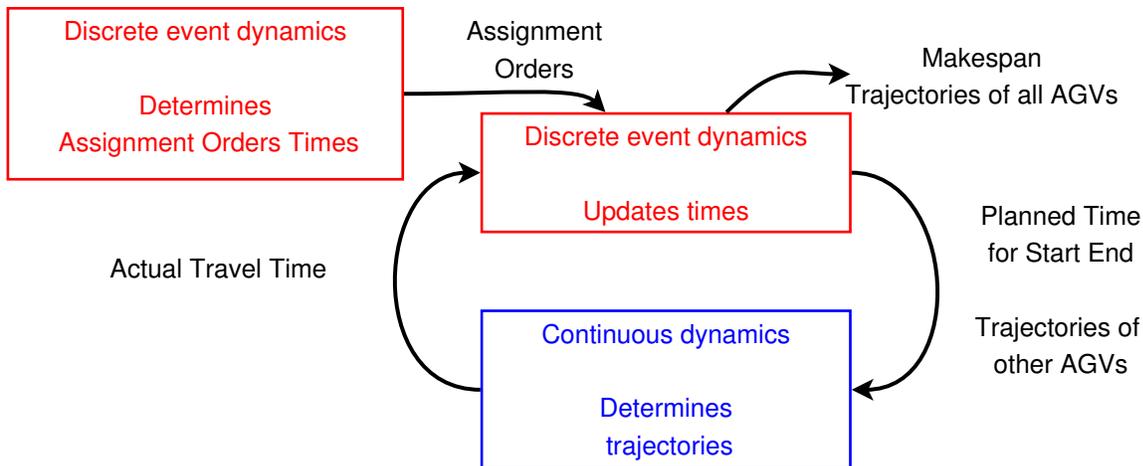


Figure 2: The hierarchical control architecture.

The discrete event dynamics delivers two key solutions, namely the assignment between containers and AGVs; and the order of containers at QCs, AGVs, ASCs. Any feasible plan can be used in the next steps. The discrete event dynamics also compute starting times of operations, which are based on the a priori given off-line duration for all of them. In theory, all operations could be completed exactly as planned without any delays or disturbance. In practice, however, operations might not happen as planned. We neglect the possibility of having exogenous delays (such as failures, malfunctions, temporarily unavailability of resources) in first instance.

In any case, the operation of AGVs might not be as planned, as the discrete event dynamics disregard their trajectory and extra time required to avoid any collision. This is driven by the continuous-time dynamics of each equipment. The continuous trajectory generation of an AGV involves collision avoidance with request to other AGVs and static obstacles. After receiving the starting time and point for performing a particular operation, the continuous dynamics controller of an AGV aims to compute the AGV movement by taking into account any obstacle. Based on the continuous-time dynamics and constraints for collision avoidance, and using a given cost function, a trajectory planning problem is formulated and solved. The obtained trajectory of the AGV is used for collision-free trajectory planning of other AGVs. The actual completion time of the AGV determines whether the trajectory is operated as planned (in case it is equal to the planned off-line one) or not. In this latter case, the duration of operations will not correspond anymore to the off-line planning, and the overall schedule needs to be updated.

The updating procedure can be integrated in the hierarchical control in many ways. One can for instance react by changing order or assignment, but the interplay of the

order, the assignment and the trajectory makes the solution of this quite complex, while not impossible in cases in which the operations are mutually influencing each other. Thus, the hierarchical control updates only the duration of operations without any change in order and assignment of containers. We moreover choose for a sequential update, i.e., one at a time the trajectory of a couple AGV-container is updated, and all times of operations that are depending on it are updated (typically, postponed). The continuous dynamics of QCs and ASCs never get updated.

Of course, the final solution might result in a sub-optimal solution of the overall problem. Convergence of the scheme is ensured by the fact that every trajectory is checked for collisions and updated in a single step. Thus, every trajectory can be updated only once, and will not be updated as a consequence of later trajectory updates, in a hierarchical sequential planning fashion.

Multiple schemes for updates are still possible, depending on the choice of the sequence of containers to update, and/or the possibility to re-update to a limited extent trajectories that have been already planned/updated. Anyway, the final output of the overall problem is the collision-free trajectory of AGVs which addresses a performance function specified at the level of discrete dynamics system (e.g., makespan).

### 3 Mathematical Models

#### 3.1 Optimization of discrete event dynamics

For modeling and solving the large-scale scheduling and routing problem described by discrete dynamic operations, we use the alternative graph formulation of Mascis and Pacciarelli (2002), that is based on two sets of constraints: *fixed constraints* need to be satisfied in any solution, while *alternative constraints* are pairs of alternative constraints. Operations are the movement of a container (job) by an equipment/machine on a container terminal resource. In case of AGVs, there is a choice of which particular AGV can be used. This additional decision is modelled as a routing decision (i.e., the assignment of an AGV vehicle to a container). The problem variables are the following:

$y_{jr}$  models whether container (job)  $j$  uses route  $r$ , associated with a particular AGV vehicle.  $n_{agv}$  is the amount of AGV, which corresponds to the routing alternatives for job  $j$ . In a feasible schedule, we need to select one routing alternative  $r$  (i.e. one AGV) for each job  $j$ . In total, there are  $J$  jobs.

$x_{(jkr,imp),(ikp,jmr)}$  represents the order between operations  $jkr$  and  $ikp$ , related to jobs  $j$  and  $i$  over the resource  $k$ , along two possibly different routes, respectively  $r$  and  $p$ . Each route differs in the choice of a particular AGV chosen. Here we assume that routes are in one-to-one correspondence to particular AGV vehicles to perform the movement.

$t_{jkr}$  represents the starting time of operation  $jkr$ , performed by the container (job)  $j$  on resource  $k$  using the route  $r$ .

The interacting sequence of operations can be described as the following hybrid flow shop model with flexible routing:

$$\begin{aligned}
& \min t_n - t_0 \\
& t_{jkr} - t_{jmr} \geq w_{jmr\_jkr}^F + G(1 - y_{jr}) \quad (Fix) \\
& t_{imp} - t_{jkr} \geq w_{jkr\_imp}^A + \\
& \quad G(2 + x_{(jkr,imp),(ikp,jmr)}) - y_{ip} - y_{jr} \quad (Alt) \\
& t_{jmr} - t_{ikp} \geq w_{ikp\_jmr}^A + \\
& \quad G(3 - x_{(jkr,imp),(ikp,jmr)}) - y_{ip} - y_{jr} \quad (1) \\
& \sum_{r=1}^{n_{agv}} y_{jr} = 1 \quad j = 1, \dots, J \\
& y_{jr} \in \{0, 1\} \\
& x_{(jkr,imp),(ikp,jmr)} \in \{0, 1\}
\end{aligned}$$

Operation 0 is a dummy operation that precedes all the other operations, to give a common temporal reference  $t_0$ . Operation  $n$  is a dummy operation that follows all the other operations, and is used to keep track of the makespan.  $G$  is a sufficiently large number. Here we define the makespan as the time  $t_n$  at which the last container leaves the vessel.

The fixed constraints are associated to sequences of operations  $(jmr, jkr) \in Fix$ , corresponding to the physical movement of a container from the resource (machine)  $k$  to the resource  $m$ , where  $r$  is the choice of which particular route considered. Those describe naturally a chain of operations, whose duration is determined by the off-line planning, or by the trajectory planning module. A constraint in  $Fix$  is active when job  $j$  uses route  $r$ , i.e.  $y_{jr} = 1$ , and implies that  $t_{jkr} \geq t_{jmr} + w_{jmr\_jkr}^F$ . The weight  $w_{jmr\_jkr}^F$  is the minimum *processing time* (i.e., a minimum time interval) between the two consecutive operations  $jmr$  and  $jkr$ .

The alternative constraints are in pairs, each of them representing an ordering decision between two conflicting operations. Those are defined for each alternative pair  $((jkr, imp), (ikp, jmr)) \in Alt$  that is associated to a pair of operations  $jkr$  and  $ikp$  which are conflicting being on the same resource  $k$ . A constraint of an alternative pair  $((jkr, imp), (ikp, jmr)) \in Alt$  is active when job  $i$  uses route  $p$ , i.e.  $y_{ip} = 1$ , and job  $j$  uses route  $r$ , i.e.  $y_{jr} = 1$ . Regarding the two constraints of this alternative pair, the constraint  $t_{imp} \geq t_{jkr} + w_{jkr\_imp}^A$  is active when  $x_{(jkr,imp),(ikp,jmr)} = 0$ , while the constraint  $t_{jmr} \geq t_{ikp} + w_{ikp\_jmr}^A$  is active when  $x_{(jkr,imp),(ikp,jmr)} = 1$ . The weights  $w_{jkr\_imp}^A$  and  $w_{ikp\_jmr}^A$  are the minimum *setup times* between the conflicting operations. This time interval depends on a variety of factors, including the safety zone around an AGV, the movement back and forth for the cranes, and it is also determined by the off-line planning and by the trajectory planning module. The sequencing problem is sequence dependent when  $w_{jkr\_imp}^A \neq w_{ikp\_jmr}^A$ .

To compute a solution to this model the following approach is used, based on the structure in [5, 4]. Basically the problem is decomposed by considering the routing variables, (i.e., the assignments of containers to AGV) separately than the scheduling variables (i.e., the orders on the machines, and the related timing). Iteratively, the solver computes a

new schedule for a given assignment of AGV to containers; and changes those assignments in search for improvements. Alternatively, the overall scheduling and routing formulation can be solved by a commercial Mixed-Integer Linear Programming (MILP) solver.

The scheduling procedure with a given assignment of AGV to containers is based on a branch-and-bound scheduling algorithm in [6], which is an exact algorithm truncated at a given maximum computation time. The algorithm computes a starting solution via a set of scheduling heuristics based on dispatching rules and graph properties. A near-optimal solution is computed in a short time by the branch-and-bound algorithm, starting from a reference scheduling solution obtained via greedy heuristic procedures and using a lower bound computed on a simplified version of the scheduling problem. In particular, the algorithm is based on a binary branching scheme in which the branching decision is a sequencing order between two jobs in a resource.

The procedure changing assignments of AGV is instead based on a local search and a metaheuristic procedure, which, starting from any given scheduling and routing solution, iteratively changes assignment, in search for routing improvements. The procedure returns the best schedule and the best assignment between AGV and containers after a stopping criteria is reached, in this case a maximum computation time. We consider two metaheuristics in this paper: a Variable Neighborhood Search (VNS) and a Tabu Search (TS). These algorithms have been implemented in the AGLIBRARY optimization environment, that is a solver developed at Roma Tre University and used in many different fields of applications so far [18, 19].

The VNS metaheuristic is based on the combination of different neighborhoods. Based on a local solution to the scheduling and routing problem, and an assignment AGV-container, a move is the assignment of one or more containers to a different AGV. This is done based on neighborhoods related to the makespan and the structure of the problem, similar to [16, 17]. The variable neighborhood search alternates a local search phase in which the neighbors of a solution are evaluated and the best one is selected, to a perturbation phase in order to escape from a local minimum in case such situation is reached. This latter phase consists in changing the neighborhood, in this case considering a neighborhood of larger size. When the neighborhood size increases, the number of changed routing decisions increases. The VNS algorithm used is an adaptation of the “Basic” VNS described in Hansen et al. [10].

The TS metaheuristic is a metaheuristic based on local search, which makes extensive use of memory for guiding the search. Each neighbourhood strategy restricts the set of moves to be explored in order to speed up the search of the best move. The neighbourhood strategies tested in this paper are based on routing decisions on the critical path, similar to [5, 4]. When no potentially better solution is found on the incumbent solution neighbourhood, the search alternates neighbourhood strategies based on the critical path method [5, 4] with a diversification strategy, which consists of changing at random the AGV assigned to a number of containers at the same time. All neighbours are evaluated via some scheduling heuristics. The best neighbour is set as the move to be made, and re-evaluated via the branch-and-bound scheduling algorithm; the resulting best solution is set as the new incumbent solution. The inverse of the chosen move is stored in a tabu list of a given length, that is used to avoid being trapped in local optima and revisiting the same solution. The moves in the tabu list are forbidden for a number of iterations and no aspiration criteria is used.

The overall framework returns thus a detailed schedule in which an AGV is assigned

to every container in the studied time horizon of prediction, and all potential sequencing (not on the trajectory) conflicts on cranes and AGV are solved.

### 3.2 Optimal control for continuous trajectory planning

This section focuses on the continuous-time dynamics of the machines. We here call (QCs, AGVs and ASCs) equipment, to underline that a continuous dynamics is considered. The models of QCs and ASCs are less complex than AGVs as they have single-dimension movements, and no interaction with each other and do not require particular attention. For more details, see [23].

We assume that each AGV to have identical dynamics. For each AGV, a discrete point-mass model is used to approximate the dynamical behavior in two-dimensional space according to a given acceleration and speed as follows: at time instant  $k$ ,  $\forall p \in [1, \dots, n_{\text{agv}}]$

$$\begin{bmatrix} \mathbf{r}_p(k+1) \\ \mathbf{v}_p(k+1) \end{bmatrix} = \begin{bmatrix} I_2 & \Delta t I_2 \\ 0_2 & I_2 \end{bmatrix} \begin{bmatrix} \mathbf{r}_p(k) \\ \mathbf{v}_p(k) \end{bmatrix} + \begin{bmatrix} 0.5(\Delta t)^2 I_2 \\ \Delta t I_2 \end{bmatrix} \mathbf{u}_p(k), \quad (2)$$

where AGV  $p$  has a position  $\mathbf{r}_p(k) = [r_p^x(k) \ r_p^y(k)]^T$  and a velocity  $\mathbf{v}_p(k) = [v_p^x(k) \ v_p^y(k)]^T$ . Each AGV is assumed to respond to accelerations, which is the only control action possible  $\mathbf{u}_p(k) = [u_p^x(k) \ u_p^y(k)]^T$ . Acceleration and speed have theoretical constraints  $|\mathbf{u}_p(k)| \leq u_{\text{max}}$  and  $|\mathbf{v}_p(k)| \leq v_{\text{max}}$ . Those latter limits on velocity and acceleration of AGVs are approximated by polygons using linear equations [14] as follows:  $\forall m \in [1, \dots, M]$

$$v_p^x(k) \sin\left(\frac{2\pi m}{M}\right) + v_p^y(k) \cos\left(\frac{2\pi m}{M}\right) \leq v_{\text{max}} \quad (3)$$

$$u_p^x(k) \sin\left(\frac{2\pi m}{M}\right) + u_p^y(k) \cos\left(\frac{2\pi m}{M}\right) \leq u_{\text{max}}, \quad (4)$$

where  $M$  is an arbitrary number used to achieve a quasi-circular approximation of the constraints. Large  $M$  results in a better approximation; a value of  $M=10$  is used.

Besides the vehicle dynamics, two types of obstacles of AGVs are also considered. The first one is the static obstacle represented by the stacking area near the transfer point. There are two tracks of a stacking crane on one side of the stack where containers are handled. For security reasons, AGVs should not approach the area of these tracks. The other possibility is a collision with another moving AGV. To avoid both types of the collision safely, we consider each AGV has a rectangle safety zone which should at all times be free from obstacles. The avoidance requirement can be described using linear equations in order to force the position of the obstacles to be outside of the forbidden area. Static obstacles are represented by their extreme coordinates ( $s^{\text{high},x}$ ,  $s^{\text{high},y}$ ,  $s^{\text{low},x}$ ,  $s^{\text{low},y}$ ), and correspond to areas where the AGV cannot move closer than a distance  $d$ :

$$\begin{aligned} r_p^x(k) &\leq s^{\text{low},x} - d + Rb_{\text{in},1} \\ r_p^x(k) &\geq s^{\text{high},x} + d - Rb_{\text{in},2} \\ r_p^y(k) &\leq s^{\text{low},y} - d + Rb_{\text{in},3} \\ r_p^y(k) &\geq s^{\text{high},y} + d - Rb_{\text{in},4} \end{aligned} \quad (5)$$

$$\sum_{\tau=1}^4 b_{\text{in},\tau} \leq 3, \forall p \in [1, \dots, n_{\text{agv}}] \quad (6)$$

where  $R$  is a large positive number and  $b_{in,\tau} \in \{0, 1\}$  is a binary variable for modeling the AGV position with regard to the obstacle.

In the case when multiple AGVs are transporting containers to different destinations, the collisions between vehicles can be described in a similar manner. The moving obstacle AGV  $p_2$ , seen from the point of view of AGV  $p_1$ , can be described as a sequence of static obstacles over time as follows:

$$\begin{aligned} r_{p_1}^x(k) &\leq r_{p_2}^x(k) - 2d + Rb_{m,1}(k) \\ r_{p_1}^x(k) &\geq r_{p_2}^x(k) + 2d - Rb_{m,2}(k) \\ r_{p_1}^y(k) &\leq r_{p_2}^y(k) - 2d + Rb_{m,3}(k) \\ r_{p_1}^y(k) &\geq r_{p_2}^y(k) + 2d - Rb_{m,4}(k) \end{aligned} \quad (7)$$

$$\sum_{\tau=1}^4 b_{m,\tau}(k) \leq 3, \forall k \quad (8)$$

The trajectory planning problem is the problem of determining a trajectory (sequence of points and feasible speed/ accelerations) such that the destination point is reached within a target time, and the collision-free trajectory for a single AGV is generated by considering the trajectories of the other AGVs. The origin  $\mathbf{r}_p^0$  and destination  $\mathbf{r}_p^f$  depend on the transfer point of the container to be processed at the QC and the ASC. The starting time  $t_p^0$  and the target ending time  $t_p^f$  are given by the discrete-event dynamics.

Suppose  $T$  is a number larger than  $t_p^f$ . Within a given interval  $[0, \dots, T - 1]$ , AGV  $p$  reaches  $\mathbf{r}_p^f$  at time  $k$ . These constraints can be represented [14] as follows:  $\forall k \in [1, \dots, T - 1], \forall p \in [1, \dots, n_{agv}]$ ,

$$\begin{aligned} r^x(k) - r_p^{fx} &\leq R(1 - b_p(k)) \\ r^x(k) - r_p^{fx} &\geq -R(1 - b_p(k)) \\ r^y(k) - r_p^{fy} &\leq R(1 - b_p(k)) \\ r^y(k) - r_p^{fy} &\geq -R(1 - b_p(k)) \end{aligned} \quad (9)$$

$$\sum_{k=1}^{T-1} b(k) = 1, \forall k \in [1, \dots, T - 1] \quad (10)$$

where the constraints in (9) are active only when  $b_p(k) = 1$ . Equations (9) and (10) force the position  $\mathbf{r}_p^f(k)$  of AGV  $p$  to reach the target  $\mathbf{r}_p^f$  with condition  $b_p(k) = 1$ .

Then the trajectory planning problem is to minimize the deviation on the arrival time, which can be described as  $\sum_{k=1}^{T-1} |kb_p(k) - t_f|$ , where  $kb_p(k)$  is the ending time of the trajectory. Here we also consider improve energy efficiency of the planned trajectory and minimize the sum of accelerations with a small penalty  $\lambda$  ( $=0.01$ ) in the objective function, considered as an additional secondary objective. Thus, the resulting minimal-time optimization problem can be formulated as follows:

$$\min_{\mathbf{u}, \mathbf{b}} \sum_{k=1}^{T-1} |kb_p(k) - t_p^f| + \lambda \sum_{k=0}^{T-1} (|u_x(k) + u_y(k)|), \quad (11)$$

subject to (2)-(10), where  $\mathbf{u} = [u(0), u(1), \dots, u(T - 1)]^T$  denotes continuous decision variables and  $\mathbf{b} = [b_p(0), b_p(1), \dots, b_p(T - 1)]^T$  denotes binary decision variables.

The optimization problem above can be modelled as a MILP formulation [23]. After the trajectory is planned, the actual time for processing the operation of the AGV and the ending time of the operation is updated in the discrete-event dynamics.

## 4 Practical case study

We provide here the solution for a practical case study at a container terminal based on 4 QCs, 6 AGVs and 6 stacking cranes, in a standard layout. We simulate 40 containers to be transported from the QC to the ASC, considering only a single-directional setup. This is without loss of generality; the general applicability of the framework holds also for the opposite direction, and also for mixed loading/unloading operations. The instance corresponds to about 16819 ordering decisions variables  $x_{(jkr,imp),(ikp,jmr)}$  across QCs, AGVs, ASCs, and 240 routing variables  $y_{ip}$ .

### 4.1 Evaluation of discrete event dynamics

We here analyze the performance of the discrete event scheduler. The makespan of the best solution computed by the AGLIBRARY optimization environment used, before trajectory planning, is 466 seconds. We note here that the approach used in [23] and solved with the commercial solver CPLEX returns no integer solution after one hour of computation time.

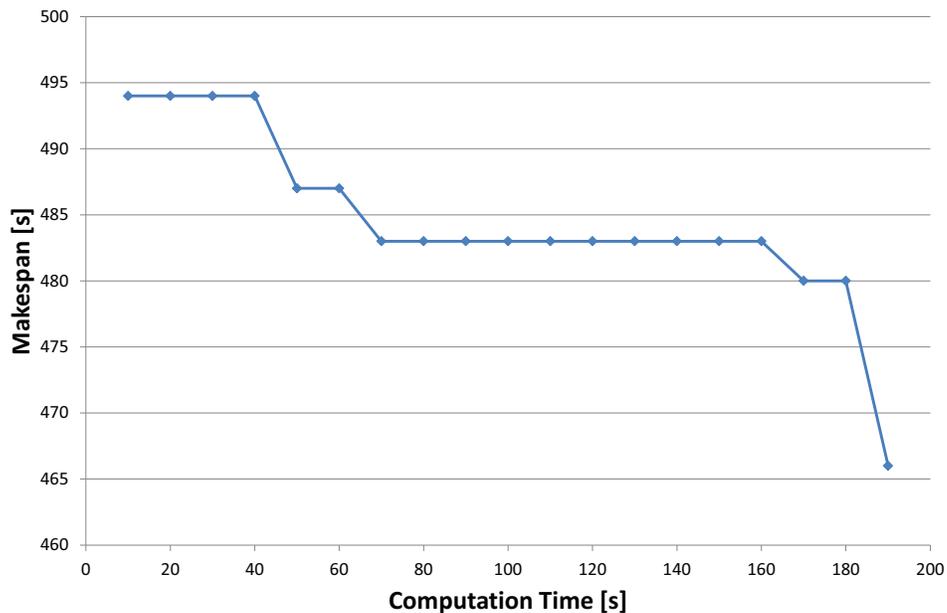


Figure 3: Solution quality over computation time.

The maximum computation time given to AGLIBRARY is 10 minutes. The evolution of solution quality against time is plotted in time is reported in Figure 3. After around the first 3 minutes of computation, the solution does not improve.

A first solution is found in less than 10 seconds, with value 494 for the makespan. The best solution is then found just around 180 seconds. The gap between the first solution found and the best solution is in this case limited by 6%.

Figure 4 reports a Gantt chart of the solution. On the Y-axis the equipment/machines are reported, i.e. from top to bottom 4 QCs, 6 AGVs and 6 ASCs. Containers are reported as boxes in the Gantt chart, with their ID inside the box. The solution has some slack, in the sense that 7 out of the 40 containers can shift in time without changing the makespan or even influencing the completion time of the last containers on their last machine.

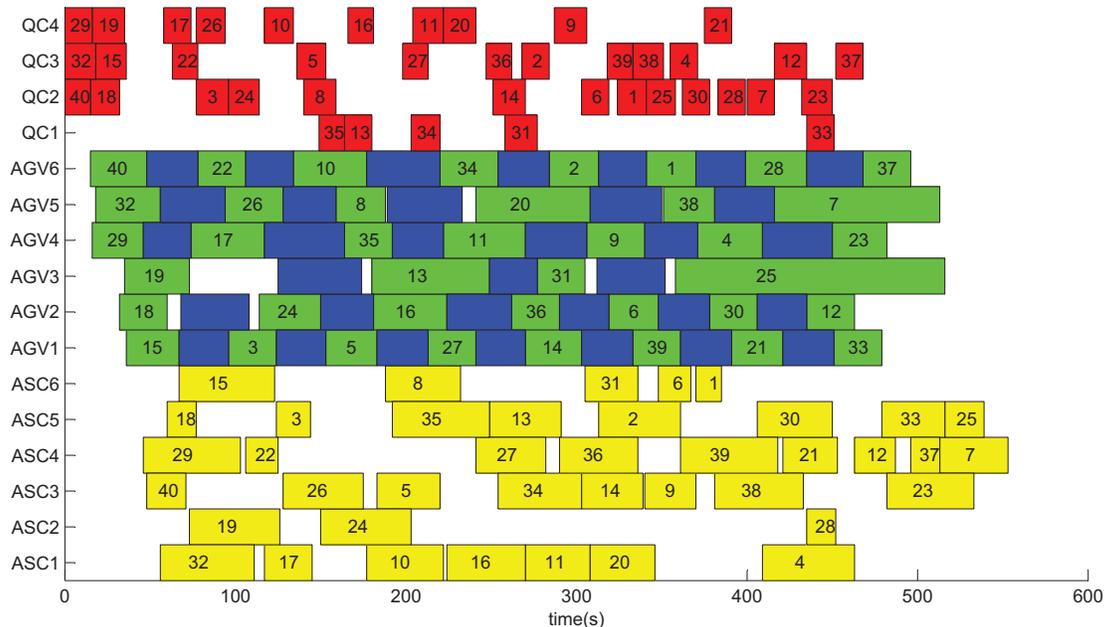


Figure 4: Gantt chart of the solution after the trajectory planning.

## 4.2 Evaluation of trajectory planning

The output of the trajectory planning module results in updated trajectories for 12 of the containers out of 40 (30%). The total makespan raises to 468 seconds (from 466 seconds).

Figure 5 reports the actual trajectories of the AGVs moving the containers, in the bidimensional space of the quay area. We observe that it is evident how for some AGVs the trajectory followed does not correspond to the minimal-distance trajectory, but instead to the time-tracking trajectory, based on the trajectories of all other AGVs.

The average driving distance of free-ranging AGVs is 115.9 meters, which is significantly less than when a mesh routing [7] would be used. The latter routing would require 146.7 meters of average driving distance.

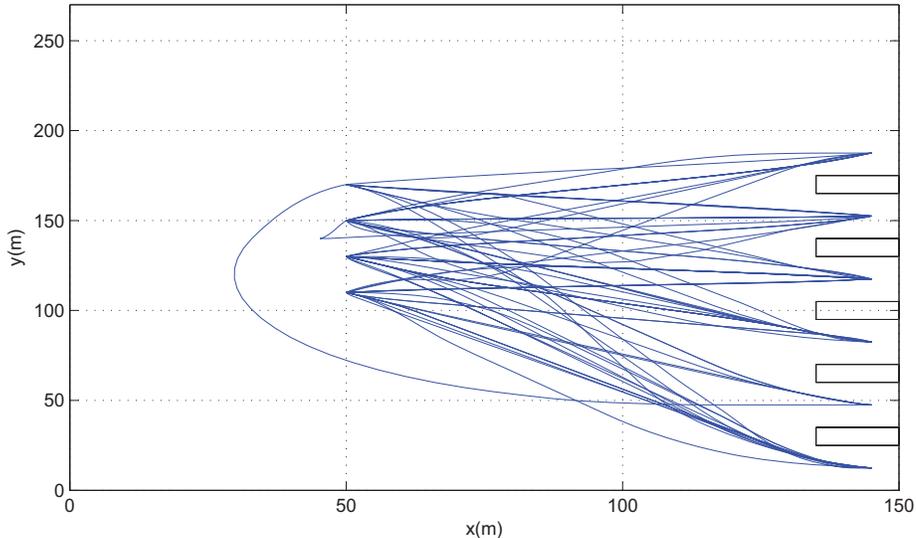


Figure 5: Trajectories of the AGVs.

## 5 Experimental assessment on realistic instances

We extend the computational analysis in this section to a set of realistic instances that generated from the practical case study of the previous section. We generated the realistic instances by varying the number of containers and AGVs in the following range:  $\{10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80\}$  and  $\{6, 8, 10\}$ , respectively. In total, we have 45 realistic instances, that are investigated for 6, 8 or 10 AGVs. The aim of this section is to compare the advanced metaheuristics with the commercial solver CPLEX when dealing with discrete event dynamics.

Table 1 presents the number of ordering ( $x$ ), routing ( $y$ ) and timing ( $t$ ) variables and the number of constraints of the MILP formulation. This information is the average information on the 45 realistic instances for 6, 8 or 10 AGVs. When increasing the number of containers, the ordering (routing) [timing] variables varies from 1014 (60) [80] till 71735 (480) [640] in case with 6 AGVs, from 1662 (80) [100] till 117486 (640) [800] in case with 8 AGVs, from 2470 (100) [120] till 176163 (800) [960] in case with 10 AGVs.

Table 1: Characteristics of the discrete event instances

Num. of AGVs	Num. of Variables			Num. of Constraints
	$x$	$y$	$t$	
6	27094	260	360	55587
8	44297	360	450	90444
10	65857	450	540	134012

Tables 2, 3, 4 report the computational results obtained for 6, 8 and 10 AGVs. We evaluate the performance of the CPLEX solver and the AGLIBRARY algorithms: the truncated branch-and-bound (BB) algorithm for the scheduling problem without routing flexibility, the Variable Neighborhood Search (VNS) and the Tabu Search (TS) for the

overall scheduling and routing problem. For each algorithm/solver, we give the average value on the 45 realistic instances in terms of the makespan (in seconds) and the computation time (in seconds). A maximum computation time of 10 minutes was given for each algorithm/solver. We observe that the CPLEX solver failed to compute a feasible solution within the given computation time for 2 instances with 6 AGVs, 1 instance with 8 AGVs and 4 instances with 10 AGVs. In case of infeasibility, we assigned to CPLEX the value of the worst solution computed by AGLIBRARY in Tables 2, 3, 4.

Table 2: Results for 6 AGVs

Algorithm	Makespan (s)	Comp. Time (s)
BB	669.7	485.7
VNS	560.7	294.7
TS	<b>557.3</b>	435.8
CPLEX	670.9	600.0

Table 3: Results for 8 AGVs

Algorithm	Makespan (s)	Comp. Time (s)
BB	491.8	451.7
VNS	491.6	302.6
TS	<b>489.3</b>	499.6
CPLEX	594.2	600.0

Table 4: Results for 10 AGVs

Algorithm	Makespan (s)	Comp. Time (s)
BB	950.2	600.0
VNS	<b>446.3</b>	263.8
TS	453.8	489.7
CPLEX	512.0	600.0

Comparing the results obtained for the various algorithms/solvers, TS and VNS outperform BB and CPLEX. Specifically, TS is, on average, the best algorithm in terms of solution quality for the instances with 6 and 8 AGVs. This result is due to the fact that TS searches for better routes in well-focused and quite restrictive neighborhoods. Differently, VNS operates more extensive routing modifications during each move compared to TS. The diversification strategy of VNS gives positive results for the large size instance with 10 AGVs, for which VNS outperforms the other approaches in terms of both solution quality and computation time.

Figure 6 presents the improvement between the solution computed by the VNS algorithm, and considering off-line assignment of AGVs for the containers. The makespan improvement is measured in percentage of the makespan, for each instance.

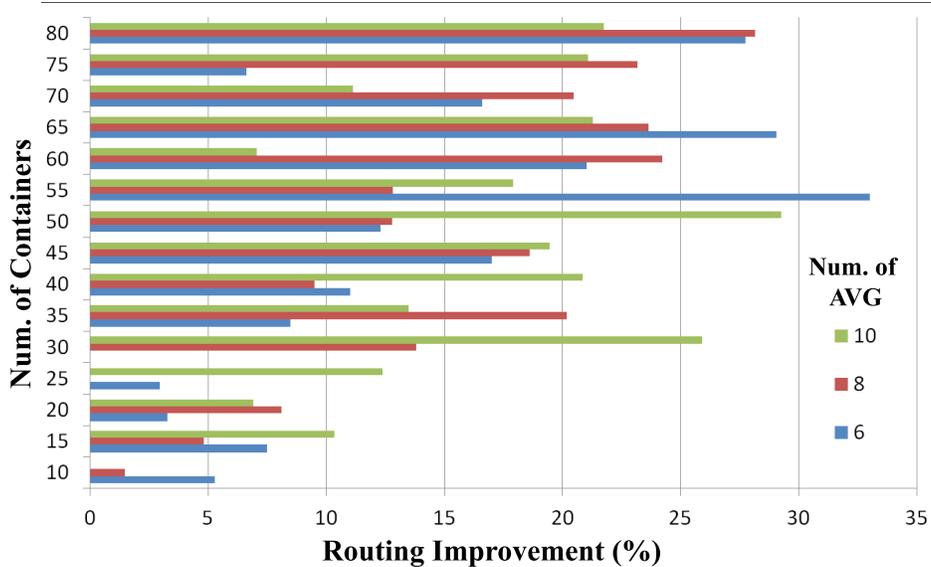


Figure 6: Routing improvement when varying the number of containers.

From the results of Figure 6, the improvement goes beyond 30% for some instances, being larger for the instances with more than 25 containers. Therefore, the larger is the control horizon the more is the potential improvement of optimal routing options.

## 6 Conclusions and further research

This paper considers the problem of controlling operations at an automated container terminal. In particular, a mathematical formulation based on the alternative graph model of [12] is proposed for solving the underlying hybrid flow shop model with flexible routing. The AGLIBRARY solver is subsequently adopted to solve the problem. This solver is able to compute a good quality solution in an acceptable computation time, while the sheer size of larger instances (mainly due to the amount of containers) might not always allow to get a solution timely when the CPLEX solver is used. AGVs exploit free range navigation, saving distance and time. The makespan is affected only to a minor extent by the free range routing.

The hierarchical control approach is able to perform a continuous trajectory planning for all movements of pieces of equipments. The framework allows to handle large instances provided that: a solution can be found by the discrete event scheduler; the trajectory planner can solve its problem; and the update of trajectories does not result in a continuous-dynamics deadlock (i.e. two vehicles which cannot move forward due to the presence of the other vehicle on the respective trajectories).

We note that discrete-dynamics deadlocks are avoided by the discrete event scheduler, while in general the trajectory planning operating one AGV at a time has limited guarantee to deliver a deadlock-free plan by itself. The inclusion of suitable constraints or extended buffer in the discrete event scheduler to prevent the possibility of having continuous-dynamics deadlock should be subject of further work.

The computational results provided in this paper confirm that metaheuristic approaches can successfully solve large-scale instances (keeping a discrete-dynamic per-

spective) and this can be coupled successfully with hybrid dynamics. Specifically, the metaheuristics of AGLIBRARY clearly outperform the results obtained with CPLEX in terms of both solution quality and computation time, thanks to the development of problem dedicated algorithms. Basic key performance indicators of the container terminal can be found when free-ranging AGVs are employed.

A follow up of this work would include a larger set of experiments in order to investigate the impact of different solutions in terms of scheduling and routing planning and in terms of trajectory planning. A special focus should be directed to further study large scale instances in which the capacity of the AGV area might become a bottleneck. Moreover, future research can be directed to the study of integrated approaches to solve quickly the problem, as well as the inclusion of energy-aware planning of the moving equipment.

## References

- [1] Bierwirth, C., Meisel, F. (2010) A survey of berth allocation and quay crane scheduling problems in container terminals. *European Journal of Operational Research* 202 615–627.
- [2] Carlo, H., Vis I., Roodbergen, K. (2014) Transport operations in container terminals: Literature overview, trends, research directions and classification scheme. *European Journal of Operational Research* 236 1–13.
- [3] Chen, Lu, Nathalie Bostel, Pierre Dejax, Jianguo Cai, Lifeng Xi (2007) A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal. *European Journal of Operational Research* 181 40–58.
- [4] Corman, F., D’Ariano, A., Pacciarelli, D., Pranzo, M. (2010) A tabu search algorithm for rerouting trains during rail operations, *Transportation Research, Part B* 44(1) 175–192.
- [5] D’Ariano, A., Corman, F., Pacciarelli, D., Pranzo, M. (2008) Reordering and local rerouting strategies to manage train traffic in real time, *Transportation Science* 42(4) 405–419.
- [6] D’Ariano, A., Pacciarelli, D., Pranzo, M. (2007) A branch and bound algorithm for scheduling trains in a railway network, *European Journal of Operational Research* 183(2) 643–657.
- [7] Duinkerken, M., Ottjes, J., Lodewijks, G. (2006) Comparison of routing strategies for agv systems using simulation. *Proceedings of the Winter Simulation Conference (WSC 2006)*. Monterey, CA. 1523–1530.
- [8] Expósito-Izquierdo, Christopher, Belén Melián-Batista, J. Marcos Moreno-Vega (2011) Variable Neighbourhood Search for the Quay Crane Scheduling Problem. *Proceedings of the 11th IEEE International Conference on Intelligent Systems Design and Applications*.
- [9] Hall, N.J., Sriskandarajah, C. (1996). A survey on machine scheduling problems with blocking and no-wait in process. *Operations Research*, 44 510–525.

- [10] Hansen, P., Mladenovic, N., Moreno Pérez, J.A. (2008) Variable neighbourhood search: methods and applications, *4OR - Quarterly Journal of Operations Research* 6 (4) 319–360.
- [11] Lee, Y. H., Kang, J., Ryu, K. R., Kim, K. H. (2005) Optimization of Container Load Sequencing by a Hybrid of Ant Colony Optimization and Tabu Search, *Lecture Notes in Computer Science* 3611 1259–1268.
- [12] Mascis, A., Pacciarelli, D. (2002) Job shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research* 143(3) 498–517.
- [13] Meersmans, P.J.M., Wagelmans A.P.M. (2001) *Effective algorithms for integrated scheduling of handling equipment at automated container terminals*. Technical Report, Econometric Institute, Erasmus University Rotterdam.
- [14] Richards, R., How, J.P. (2002). Aircraft trajectory planning with collision avoidance using mixed integer linear programming, *Proceedings of the 2002 American Control Conference*. Anchorage, Alaska. 1936–1941.
- [15] Ruiz, R., Vzquez-Rodrguez, J.A. (2010) The hybrid flow shop scheduling problem. *European Journal of Operational Research* 2015(1) 1–18.
- [16] Samà, M., D’Ariano, A., Toli, A., Pacciarelli, D., Corman, F. (2015) A variable neighborhood search for optimal scheduling and routing of take-off and landing aircraft. *4th International IEEE Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)* 3–5 June 2015. Budapest, Hungary.
- [17] Samà, M., D’Ariano, A., Toli, A., Pacciarelli, D., Corman, F. (2015) Metaheuristics for real-time near-optimal train scheduling and routing. *16th International IEEE Conference on Intelligent Transportation Systems (IEEE ITSC 2015)* 15–18 September 2015. Las Palmas de Gran Canaria Spain.
- [18] Samà, M., D’Ariano, A., Pacciarelli D. (2013) Rolling Horizon Approach for Aircraft Scheduling in the Terminal Control Area of Busy Airports. *Transportation Research, Part E*, 60(1) 140155, ISSN: 1366-5545
- [19] Samà, M., D’Ariano, A., D’Ariano, P., Pacciarelli D. (2014) Optimal aircraft scheduling and routing at a terminal control area during disturbances. *Transportation Research, Part C*, 47(1) 6185, ISSN: 0968-090X
- [20] Stahlbock, R., Voss, S. (2008) Operations research at container terminals: a literature update. *OR Spectrum* 30 1–52.
- [21] Steenken, D., Voss, S., Stahlbock, R. (2004) Container terminal operation and operations research - A classification and literature review. *OR Spectrum* 26 3–49.
- [22] Vis, I., de Koster, R. (2003) Transshipment of containers at a container terminal: An overview. *European Journal of Operational Research* 147 1–16.
- [23] Xin, J., Negenborn, R.R., Lodewijks, G. (2014) Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control. *Transportation Research Part C* 44 214–230.

- [24] Xin, J., Negenborn, R.R., Lodewijks, G. (2013) Hybrid MPC for balancing throughput and energy consumption in an automated container terminal, *Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems*. The Hague, The Netherlands. 1238–1244.
- [25] Xin, J., Negenborn, R.R., Lodewijks, G. (2013) Hierarchical control of equipment in automated container terminals, *Proceedings of the 4th International Conference on Computational Logistics*. Copenhagen, Denmark. 1–17.
- [26] Xin, J., Negenborn, R.R., Lodewijks, G. (2014) Trajectory planning for AGVs in automated container terminals using avoidance constraints, *Proceedings of the 19th IFAC World Congress*. Cape Town, South Africa. 9828–9833.
- [27] Zijverden, R.J.H.A. van, Negenborn, R.R. (2012) Survey of approaches for integrated control of intermodal container terminals, *Proceedings of the 2012 IEEE International Conference on Networking, Sensing and Control*. Beijing, China. 67–72.