

A Mixed Integer Programming Model for Optimizing Multi-level Operations Process in Railroad Yards

Tie Shi

School of Transportation and Logistics
Southwest Jiaotong University
Chengdu, Sichuan, China, 610031
Tel.: (86)-13518128459
Email: tie.shi1986@gmail.com

Xuesong Zhou

School of Sustainable Engineering and the Built Environment
Arizona State University
Tempe, AZ 85287, USA
Tel.: (1)-480-9655827
Email: xzhou74@asu.edu

Abstract: A typical railroad hump yard contains multiple layers of complex operations. The railcars coming with inbound trains through the yard need to be humped into different classification tracks according to the destination, and then assembled to generate the desired outbound trains. During this complex procedure, the processing time of railcars and various resource constraints at different railroad yard facilities could significantly affect the overall performance of yard operations, individually and in combination. It is theoretically challenging to represent a large number of practical operation rules through tractable mathematical programming models. This paper first presents a time-expanded multi-layer network flow model to describe the connection between different layers of yard operations. A mixed integer programming model is developed to optimize the overall performance by jointly considering tightly interconnected facilities. We adopt a cumulative flow count representation to model the spatial capacity constraints in terms of the number of railcars in classification yards. A novel lot-sizing modeling framework and related valid inequality formulations are introduced to model the assembling jobs for outbound trains. We also develop an aggregated flow assignment model and earliest due date-based heuristic rules to determine the humping jobs sequence for reducing the search space. Numerical experiments are conducted to examine the solution quality and computational efficiency under different types of formulation strategies.

Keywords Railroad Yard, Operations Plan, Mixed Integer Programming, Cumulative Flow Count, Valid Inequality

1. Introduction

Railroad yards, i.e. marshalling or shunting yards, play an important role as consolidation nodes in rail freight transportation networks. The performance of railroad yard operations is extremely important for railroad industries to consolidate and redistribute shipments efficiently. In this complex process, a critical technical document called railroad yard operations plan determines the schedule of various tasks within a railroad yard. Specifically, inbound trains are disassembled or humped, and the railcars are then reorganized to generate outbound trains via a system of tracks and switches. Using this consolidation and redistribution procedure, railcars can be sent through the network efficiently without providing a large number of end-to-end services. In practice, the processing time of freight railcars in a railroad yard represents a large proportion of the total railroad end-to-end transportation or trip time, so continuously improving the efficiency of railroad yard operations has received significant attention by decision makers and operations researchers in the

rail industry.

The railroad yard operations plan problem (YOP) aims to design an optimal tactical task schedule of critical activities subject to different practical rules. Typically, the number of railcars processed during a certain period and the average waiting time (i.e. processing time) of railcars are important measures of effectiveness (MOE) of a railroad yard. There are a number of important practical constraints, e.g., the maximum number of cars that can be stored in each track, or specific requirements related to the humping sequence for inbound trains and the combination of destinations for outbound trains. Moreover, the yard operation problem has to consider multiple types of commodities, e.g., railcars, inbound trains and outbound trains, as well as the schedule sequences for two layers of servers: humping engines and pull-back engines. Thus, how to seamlessly integrate different layers of flow and activities has been a very challenging question, especially with respect to theoretically rigorous optimization models.

The railroad yard operation problem has been widely studied by many researchers. The early seminal work by Beckmann, McGuire and Winsten (1956) systematically describes the main facilities, common operations, and the corresponding classification policies in a typical railroad yard. Their work also analyzed the major factors affecting railcar transit times. The reviews by Assad (1980), Cordeau et al. (1998), Li et al. (2011) and Boysen et al. (2012) have covered various YOP modeling frameworks, including discrete event simulation-based, queueing models and analytical (probabilistic) models. Recently, the INFORMS Railway Application Section Problem Solving Competition (RAS, 2013) further offers a comprehensive description of various railroad yard facilities in YOP.

In the last few decades, significant research efforts have been devoted to developing analytical models for quantifying train delay under different classification strategies. Petersen (1977a, 1977b) developed a simplified steady-state queueing model to analyze the impact of yard resources on the throughput and capacity of the yard. Turnquist and Daskin (1982) analyzed the impact of classification delays and connection delays on the train service and then developed car-based queueing models to minimize the railcar delay times. Daganzo et al. (1983) studied different multistage sorting strategies under constraints on the number of classification tracks. Daganzo (1986, 1987a, 1987b) further examined how different blocking strategies affect the number of switching tasks with homogeneous and heterogeneous traffic.

Another active research area is how to construct mathematical programming and simulation YOP models. Researchers have mainly focused on individual sub-problems of YOP, such as an early study on blocking plans by Zhu and Zhu (1983). Dahlhaus et al. (2000a, 2000b) considered humping yard reclassification of outbound trains as a combinatorial optimization problem that aims to minimize the essential sorting steps for modifying the sequence of railcars in outbound trains. Newton et al. (1998) and Barnhart et al. (2000) focused on constructing a network-wide blocking plan involving multiple railroad yards. In their studies dealing with aggregated flow, network design models were developed to minimize the total mileage, delay and handling costs. Bohlin et al. (2011a, 2011b, 2012) and Gestrelus et al. (2013) proposed mixed integer programming models of the track allocation plan which aim to minimize the number of extra classification operations subject to several operational constraints.

More attention recently has been paid to the topic of integrating different facilities in railroad yards. Jacob et al. (2007, 2011) and Márton et al. (2009) presented new methods of encoding humping schedules for outbound trains. The integer programming models they developed aim to minimize the time of executing the entire schedule or the number of sorting steps, subject to the spatial capacity constraints of each track. In studies by He et al. (2000, 2003), large-scale binary integer programming models are used to represent many practical railroad yard operation requirements, such as dispatching preference, operation plan flexibility, train size requirements, as well as technical inspection time constraints. Heuristic methods were used to maximize railroad yard throughput and on-time service of yard operation plans.

In an INFORMS RAS 2013 problem solving competition (INFORMS RAS, 2013), detailed

problem statements for YOP were provided for various layers of operations processes. Wang et al. (2013) developed a mixed integer programming model and heuristics algorithms in Python/C++ with a commercial solver ILOG CPLEX to solve YOP as a destination-based multi-commodity flow optimization problem. It should be remarked that, their flow-based formulation has not specifically considered the spatial capacity constraint, which is expressed as the number of cars on different tracks at any given time. Other representative solution methods from this competition include queueing-oriented priority rules by Selvam and Borjian (2013), and a Genetic Algorithm-based method by Zhou et al. (2013). For practical applications, a simulation modeling approach is proposed by Lin and Cheng (2009, 2011) for solving YOP under realistic constraints for medium-scale and large-scale problems.

To fully exploit the potential of optimization models for YOP, this study aims to address the following modeling challenges:

(1) How to capture the interconnection between different layers of scheduling? Many existing studies specifically focus on one task of the entire scheduling process, such as humping, classification track allocation, or pull-back engine scheduling. Without a fully integrated model, it is difficult to identify time-sensitive bottlenecks occurring at different parts of the system and exploit the full benefit of seamless coordination between different layers of tasks. For example, without precisely considering the full-scale impact to the subsequent tasks (e.g. resource constraints at the assembling task), an individually optimized schedule for one single task (e.g. humping) does not necessary lead to the system-wide performance improvement or even a fully feasible operational plan subject to all resource requirements at each component.

(2) How to model non-trivial real-world operational requirements through computationally tractable optimization models? In a railroad yard, humping jobs and assembling jobs can be viewed as two specific types of queueing processes with a number of important constraints, e.g., different types of customers (i.e., railcar destination), the spatial capacity at each classification track, block-to-track assignment requirement, and minimum and maximum train size constraints for outbound trains. These practical rules are associated with specific considerations for different types of limited fixed and moving resources. An optimization model without considering real-world requirements cannot ensure the full feasibility of generated solutions, which is critically important for decision-support system deployment. On the other hand, simply introducing a large number of non-linear or if-then constraints could make the resulting program extremely difficult and complicated to solve through standard optimization packages.

(3) How to develop strong formulations and heuristic rules for mixed integer programming models? Valid inequality constraints have been widely utilized in many generic optimization problems, e.g., lot-sizing problems. But constructing effective valid inequalities for practical YOP cases is very challenging because multiple layers of operations are tightly connected with different modeling objects of interest, e.g. inbound trains at receiving yards, railcars at classification yards, and outbound trains at departure yards. In addition, effective heuristic rules are critically needed for finding close-to-optimal YOP solutions, while these heuristic techniques should be developed through a careful analysis of the underlying operation processes and practical technical considerations.

In this paper, we present a number of new modeling methods to construct a theoretically sound railroad yard optimization model that formulates many essential practical requirements. Specifically, we introduce a cumulative flow count-based representation in a time-expanded network to describe various real-world constraints, such as minimum task processing time, spatial capacity on tracks, and minimum headway intervals between humping jobs. We also introduce a lot-sizing modeling framework to concisely represent the assembling jobs for outbound trains. A tight valid inequality formulation is adopted from the lot-sizing problem to improve the strength of the linear programming relaxation for the proposed YOP mixed integer programming model. The proposed valid inequality, which can be easily incorporated in standard optimization solvers, aims to cut off infeasible fractional values for binary decisions related to essential assembling jobs while

maintaining valid integer solutions under different possible conditions. Focusing on determining inbound trains humping sequences, a flow assignment-based heuristic approach and related rules are also introduced to narrow down the humping precedence relationship of inbound trains. Furthermore, a set of comprehensive numerical experiments are designed to evaluate the strength of different relaxation formulations and the effectiveness of valid inequalities in terms of reducing computational time and improving solution quality.

This paper is organized as follows. Section 2 aims to systematically describe the railroad yard operation problem, followed by an introduction on the cumulative flow counts for different types of facility resource and processing time constraints in Section 3. Section 4 presents an integrated integer programming model to minimize the total processing time of railcars for different layers of operations. Section 5 develops a lot-sizing oriented valid inequality formulation at the departure yard. Section 6 constructs an aggregated flow assignment-based approach to heuristically determine the humping jobs sequences. Numerical experiments using standard optimization solvers are presented in Section 7 for a simplified network.

2. Problem Statement and Notations

A railroad yard typically consists of a receiving yard, a hump, a classification yard and a departure yard. Fig. 1 first describes the essential process of different tasks and different facilities in a yard. Tables 1 and 2 show the notation and input parameters for the proposed optimization model.

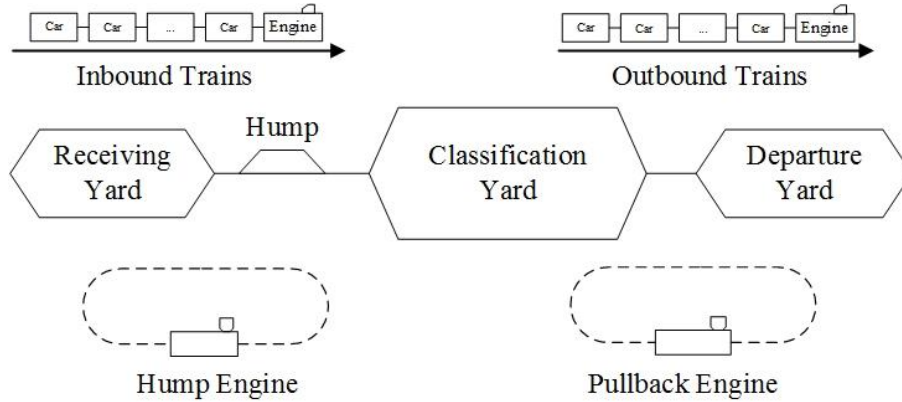


Fig. 1 Layout of the railroad yard

The essential operations can be described as follows.

1) A receiving yard has a number of tracks to receive, store and perform technical inspection for inbound trains. After the technical inspection task, an inbound train is available to be pulled out from the receiving yard by the humping engine.

2) The humping engine pushes one available inbound train so that its associated railcars will be separated and humped into corresponding tracks in the classification yard according to their destinations (i.e., block type). At any given time, only one inbound train can be humped.

3) The classification yard uses a number of tracks to sort, store and organize railcars that are humped from inbound trains to form outbound trains with predetermined destination combinations. Herein, one track only can hold one same destination of railcars in the plan.

4) The departure yard forms and stores outbound trains after the pull-back engine(s) assemble railcars from the classification yard. The number of railcars in each outbound train should also satisfy specific requirements in terms of outbound train size (e.g., between 30-120 railcars).

There are mainly three types of tasks, namely humping, sorting and assembling. Considering the complexity of the YOP problem, another type of the common yard tasks, rehumping tasks that

reorganize railcars in local outbound trains to the desired sequence matching the destinations sequence, will not be considered in the current YOP model. This study considers a fixed time horizon, indexed by $t=0,1,2, \dots T$. The inbound trains and outbound trains are identified as $i \in \{1, 2, 3, \dots I\}$ and $o \in \{1, 2, 3, \dots O\}$ respectively. The destination index $b \in \{1, 2, 3, \dots B\}$ indicates the next major stations/yards for railcars. The input data for the YOP include the arrival time of inbound trains $T^{ARR}(i)$ and the departure time of outbound trains $T^{DEP}(o)$, with $N^{IT}(b, i)$ as the number of railcars per destination b from train i . The other major input data consist of $\varphi(k, b)$ as the block-to-track assignment, where k is the index of classification tracks. In this paper, we use the “sorting by block” strategy mentioned in previous research (Daganzo et al., 1983), and assume that a classification track only accepts one single destination of railcars. As an example, Table 3 illustrates $\varphi(k, b)$ as a block-to-track assignment matrix. The block-to-outbound train matrix $\sigma(b, t)$, shown in Table 4, represents which set of destinations are allowed to depart at time t .

Table 1 General subscripts

Symbol	Description
t, t'	time index, $0, 1, 2, \dots T$
i, j	inbound trains index, $1, 2, 3, \dots, I$
o	outbound trains set, $1, 2, 3, \dots, O$
b	destination index of railcars, $1, 2, 3, \dots, B$
k	classification track index, $1, 2, 3, \dots, K$

Table 2 Input parameters

Symbol	Description
$T^{ARR}(i)$	inbound train arrival time
$N^{IT}(b, i)$	number of railcars from block b in inbound train i
H^{TI}	technical inspection duration time
$H^{HJ}(i)$	humping job duration time for train i
H^{HI}	minimum humping interval of two consecutive humping jobs
$A(k, t = 0)$	number of railcars staying in classification track k at the beginning time of the planned horizon: $t=0$
N_{max}^{CT}	maximum number of railcars can be stored on each classification track (i.e. spatial capacity)
$\varphi(k, b)$	block-to-track assignment matrix, a coefficient =1 when classification track k accepts railcars of destination b ; otherwise 0
H^{AJ}	one assembling job duration time for each track
$T^{DEP}(o)$	scheduled departure time of outbound train o
$\lambda(t)$	outbound train departure coefficient, =1 when t equals the departure time of any outbound train; otherwise 0
$\sigma(b, t)$	block-to-outbound train assignment matrix, a coefficient = 1 when destination b of railcar is allowed to depart on one outbound train which is scheduled to depart at time t ; otherwise 0
N_{max}^{OT}	maximum number of railcars in a single outbound train
μ	shortage allowance coefficient of outbound train that defines the minimum number of railcars in relation to its maximum train size, that is, $N_{min}^{OT} = \mu \times N_{max}^{OT}$
M	big M , a large positive number associated with the artificial variables for if-then constraints

Table 3 Illustrative example of block-to-track assignment matrix

$\varphi(k, b)$ value	$b1$	$b2$	$b3$	$b4$...	B
Track 1	1	0	0	0	...	0
Track 2	0	1	0	0	...	0
Track 3	0	0	1	0	...	0
Track 4	0	0	0	1	...	0
Track 5	0	1	0	0	...	0
Track 6	0	0	0	1	...	0
...
Track K	0	0	0	0	...	0

Table 4 Illustrative example of block-to-outbound train assignment matrix

$\sigma(b, t)$ value	$b1$	$b2$	$b3$	$b4$...	B
$T^{DEP}(1)$	1	0	1	0	...	0
...
$T^{DEP}(o-1)$	0	1	0	0	...	0
$T^{DEP}(o)$	0	0	1	0	...	1
$T^{DEP}(o+1)$	1	0	0	0	...	0
...
$T^{DEP}(O)$	0	1	0	0	...	1
Not departure time	0	0	0	0	0	0

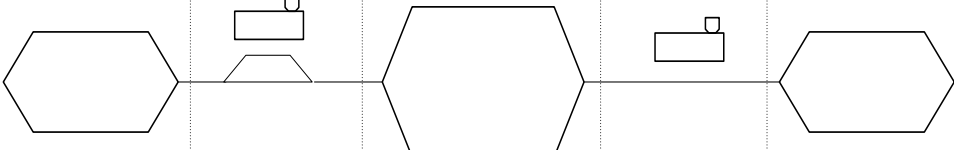
Table 5 defines the decision variables of the YOP model. The humping task essentially aims to decide $\tau^s(i)$ and $\tau^e(i)$ as the starting time and ending time of one humping job for an inbound train through a set of binary sequence variables $s(i, j)$ for inbound trains i and j . Furthermore, binary variables $x(i, t)$ correspond to the ending time of the humping job for inbound train i . All the above decision variables are related to the humping tasks at the receiving yard and the hump, as in the 2nd and 3rd columns of Table 6.

Table 5 Variables

Symbol	Description
$\tau^s(i)$	the starting time of the humping job for inbound train i
$\tau^e(i)$	the ending time of the humping job for inbound train i
$s(i, j)$	Binary variable of humping job sequence, =1 when inbound train i is humped before inbound train j , where $i < j$; otherwise 0
$x(i, t)$	Binary variable as humping job ending time indicator, = 1 when humping job of inbound train i ends at time t , otherwise 0
$A(k, t)$	cumulative arrival count of railcars to each track k in classification yard by time t , where $t > 0$
$D(k, t)$	cumulative departure count of railcars from each track k in classification yard by time t
$y(k, t)$	Binary variable of classification track operations: =1 when a pull-back engine starts to process classification track k at time t ; otherwise 0
$f^{in}(b, t)$	number of railcars with destination b processed by pull-back engine at time t from classification yard
$f^{out}(b, t)$	number of railcars with destination b processed by pull-back engine at time t to departure yard

$st(b, t)$	number of railcars with destination b stored at departure yard at time t
$d^{out}(b, t)$	number of railcars with destination b ready for outbound trains at time t which is the departure time of the corresponding outbound train
$n^{OT}(o)$	the number of railcars assembled into outbound train o

Table 6 Railroad yard operations process

Yard Operations	Humping task		Sorting task	Assembling task and departure plan	
Structure Figure of Railroad Yard facilities					
Modeling Objects/resou rces	Inbound trains	Inbound trains	Railcars	Railcar flow	Outbound trains
Input parameters	$T^{ARR}(i)$ $N^{IT}(b, i)$ H^{TI}	$H^{HJ}(i)$ H^{HI}	N_{max}^{CT} $\varphi(k, b)$ $A(k, t = 0)$	H^{AJ}	$T^{DEP}(o)$ $\sigma(b, t)$ N_{max}^{OT} N_{min}^{OT}
Arrival event variables for activity		$\tau^S(i)$	$A(k, t > 0)$	$f^{in}(b, t)$	$d^{out}(b, t)$
Departure event variables for activity	$\tau^S(i)$	$\tau^e(i)$	$D(k, t)$	$f^{out}(b, t)$	$n^{OT}(o)$
Key Binary Decision variables		$x(i, t)$ $s(i, j)$	$y(k, t)$		

At time $t=0$, the number of railcars on each classification track k is initialized as $A(k, t = 0)$. In the classification tracks, two sets of cumulative flow count variables, $A(k, t > 0)$ and $D(k, t)$, are defined to represent the overall system status in terms of the number of railcars being stored in each classification track k at time t . When the pull-back engine enters one track to assemble railcars, the track-open status variable $y(k, t) = 1$, otherwise $y(k, t)=0$. For each classification track, the number of railcars at any time is restricted by the track spatial capacity N_{max}^{CT} . These variables are related to the sorting jobs listed in the 4th column of Table 6.

The railcars pulled out from the classification track are assembled into the outbound trains. The tasks are processed by one pull-back engine. For each assembling job, the inflow of railcars $f^{in}(b, t)$ and the outflow $f^{out}(b, t)$ are the number of railcars being processed from the classification yard and to the departure yard, respectively. The decision variables related to the assembling jobs are listed in the 5th column of Table 6. After being assembled, the railcar flow will enter outbound trains as a composition part according to the block-to-outbound train coefficient $\sigma(b, t)$. The variable $d^{out}(b, t)$ is the number of railcars with destination b assembled into

outbound train which departs at time t . The outbound train size $n^{OT}(o)$, the number of railcars in the outbound train, must satisfy the minimum and maximum size constraints related to N_{max}^{OT} . All the decision variables related to departure trains are listed in the 6th column of Table 6.

The proposed optimization model aims to design an efficient operation schedule that can minimize the total processing time of railcars passing through the yard subject to different types of processing time constraints. The processing time starts from the arrival time of railcars coming with the related inbound trains or the starting time of the planned time horizon for the railcars existing in the yard when the time horizon starts. Similarly, the processing time ends the departure time of railcars leaving with outbound trains or the ending time of the planned time horizon for the railcars staying in the yard after outbound trains depart. Each inbound train must wait at least H^{TI} minutes after arriving at the receiving yard before departing to be humped. Between two consecutive humping jobs, there is a required safety or humping interval H^{HI} . The duration time of the humping job for inbound train i is given as $H^{HJ}(i)$. This problem considers only one humping engine and one pull-back engine, so only one classification track can be processed at any given time. As the schedules of inbound trains are given as parameters, the capacities of receiving yard and departure yard are assumed as large enough to process the corresponding number of scheduled trains.

3. Using Cumulative Flow Count Variables to Model Non-trivial Constraints

To represent the spatial capacity on each track and minimum processing time constraints, we adopt a cumulative flow count approach for modeling general queueing systems (Makagami et al., 1971; Newell, 1982). There are a wide range of studies using cumulative flow counts to describe traffic streams in highway systems, and recently Meng and Zhou (2014) applied the method in a train rerouting and rescheduling problem to capture the arrival and departure activities at railroad sections.

In our problem, for each type of facility resource, e.g., track in the receiving yard, classification yard or departure yard, we can use time-dependent cumulative flow counts to capture the arrival and departure activities of traveling objects/agents, as shown in Fig. 2. The horizontal and vertical differences, respectively, between cumulative arrival curve $A(t)$ and cumulative departure curve $D(t)$ correspond to (1) the total time of a certain agent spent in the system and (2) the number of agents in the system at a given time t . As a result, one can easily compute a number of key statistics in a typical queueing system, to name a few, the total waiting time of all objects, spatial capacity, inflow rate and outflow rate.

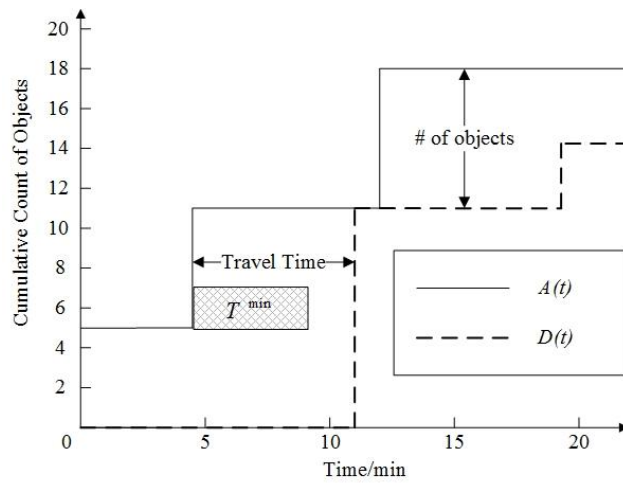


Fig. 2 Cumulative arrival and departure curves: the time-dependent number of objects in general queueing system and derived travel times by assuming the same agent arrival and departure order

Obviously, the two sets of cumulative variables must be non-decreasing, that is, $A(t+1) \geq A(t) \geq 0$ and $D(t+1) \geq D(t) \geq 0$. Let us denote N^{max} as the maximum number of objects that can be stored. The spatial capacity restriction is then expressed as Eq. (1):

$$A(t) - D(t) \leq N^{max} \quad (1)$$

On the other hand, the temporal capacity can be represented below for given arrival and departure rates R^{ARR} and R^{DEP} per unit time.

$$A(t+1) - A(t) \leq R^{ARR} \quad (2)$$

$$D(t+1) - D(t) \leq R^{DEP} \quad (3)$$

In a first-in-first-out queueing system, an object traveling through the system must satisfy the minimum required travel time, denoted as T^{min} .

$$A(t) \geq D(t + T^{min}) \quad (4)$$

4. Mathematical model for the railroad yard operation plan

4.1 Objective functions

The goal of the proposed YOP model is to minimize the total waiting time (or more precisely processing time) of all railcars, which is conceptually illustrated by the shaded area in Fig. 3, surrounded by the cumulative arrival and departure curves. Let us specifically define $A(t)$ and $D(t)$ as the cumulative arrival and departure count of railcars in the entire yard system at time t . The objective function can be expressed as Eq. (5), where $A(t)$ is predetermined by the number of railcars in inbound trains arriving before time t , that is, $\sum_{b=1}^B N^{IT}(b, i)$ for all trains satisfying $T^{ARR}(i) \leq t$. One can easily verify that $D(t) = \sum_{t'=0}^t \sum_{b=1}^B d^{out}(b, t')$.

$$\text{Min } W = \sum_{t=0}^T [A(t) - D(t)] \quad (5)$$

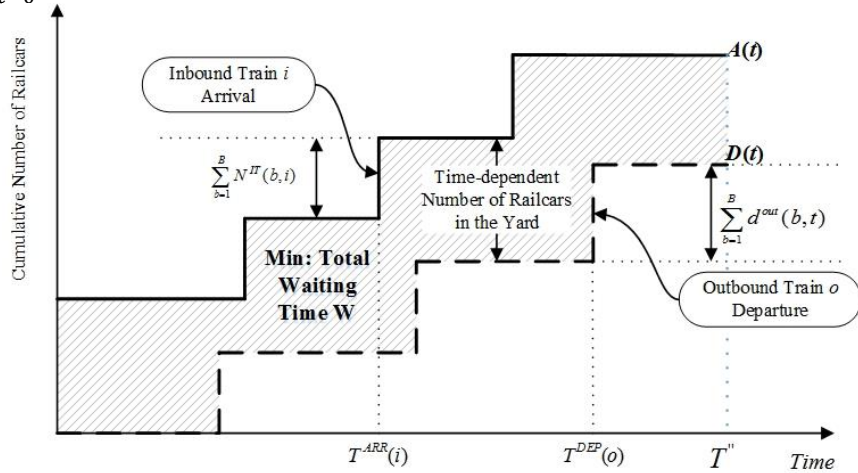


Fig. 3 Overall system cumulative arrival curve and cumulative departure curve

4.2 Humping task constraints

The humping jobs are carried out by one humping engine at the receiving yard and the hump. After technical inspection, an inbound train is available to be humped after time $T^{ARR}(i) + H^{TI}$ as shown in Eq. (6), and the humping job duration for inbound train i is represented in Eq. (7).

Technical inspection time constraints:

$$\tau^S(i) \geq T^{ARR}(i) + H^{TI}, \forall i \quad (6)$$

Humping job duration constraints:

$$\tau^e(i) - \tau^s(i) = H^{HJ}(i), \forall i \quad (7)$$

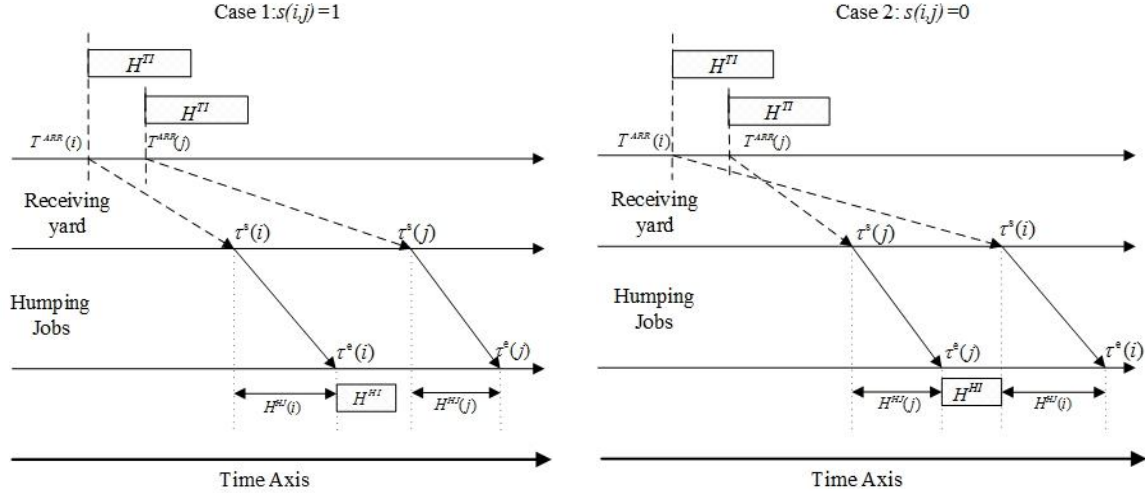


Fig. 4 Sequencing decision in humping jobs process

Eqs. (8-9) use a common integer programming formulation to model the humping sequence and minimum headway constraints between any train pair (i, j) , so as to ensure that there is no overlap between humping jobs taken by one humping engine, as illustrated in Fig. 4. These “if-then” type constraints have been used in the related train timetabling area (Zhou and Zhong, 2007).

Time headway interval and sequence constraints for humping jobs:

$$\tau^s(j) - \tau^e(i) \geq H^{HI} - [1 - s(i, j)] \times M, \forall i, j \text{ and } i < j \quad (8)$$

$$\tau^s(i) - \tau^e(j) \geq H^{HI} - s(i, j) \times M, \forall i, j \text{ and } i < j \quad (9)$$

4.3 Sorting task constraints

In the classification yard, the sorting process reorganizes the railcars humped from the inbound trains into different classification tracks according to the block-to-track assignment. After the number of railcars in one track reaches a certain number, these railcars will be pulled out by the pull-back engine and assembled into one outbound train before the number exceeds the spatial capacity. We now use the cumulative flow count method to model each classification track k as a separate sub-system in the classification yard. That is, as shown in Tables 5 and 6, there are a set of cumulative arrival/departure variables $A(k, t)$ and $D(k, t)$ for track k . Obviously, all cumulative count variables must satisfy the non-decreasing constraints.

One of the critical challenges in modeling a multi-layer system is how to handle the transition and connection for commodities between different layers, namely inbound trains and railcars in this case. In particular, Eq. (10) shows that the number of railcars received at classification track k is equal to the outflow count from inbound train i , where the binary variables $x(i, t)$ are utilized to indicate the ending statuses of the humping job for inbound train i in Eqs. (10-13).

Flow balance constraints associated with humping job:

$$A(k, t) - A(k, t - 1) = \sum_{i=1}^I \sum_{b=1}^B [x(i, t) \times N^{IT}(b, i) \times \varphi(k, b)], \forall k, t \geq 1 \quad (10)$$

Connection constraint between ending status indicator and ending time of humping job

$$\tau^e(i) = \sum_{t=0}^T [x(i, t) \times t], \quad \forall i \quad (11)$$

$$\sum_{t=1}^T x(i, t) \leq 1, \quad \forall i \quad (12)$$

In our implemented model, we also set $x(i, t) = 0$ before train i arrival time $T^{ARR}(i) + H^{TI}$. Similar to the general principle related to spatial capacity in Eq. (1), we now have the spatial capacity constraints for each classification track k at any time t , shown in Eq. (13).

Spatial capacity constraints for classification track:

$$A(k, t) - D(k, t) \leq N_{max}^{CT}, \quad \forall k, t \quad (13)$$

As there is only one pull-back engine, only one track can open to be processed by the pull-back engine at any given time. Recall that, we have introduced binary variable $y(k, t)$ to represent the open or close status of classification track k at time t . Moreover, each assembling job takes H^{AJ} minutes, so at most one variable $y(k, t)=1$ during a period of H^{AJ} minutes, as illustrated in Table 7.

Table 7 Illustrative example for open/close statuses of track k

$y(k, t)$ value	...	$t-2$	$t-1$	t	$t+1$	$t+2$	$t+3$...	$t+H^{AJ}-1$	$t+H^{AJ}$...	T
Track 1	...	0	0	0	0	0	0	...	0
Track 2	...	0	0	0	0	0	0	...	0
...	...	0	0	0	0	0	0	...	0
Track k	...	0	0	1	0	0	0	...	0
...
Track K	...	0	0	0	0	0	0	...	0
Assembling Jobs				One Assembling Job on track k lasts H^{AJ}								

Assembling job process time constraints:

$$\sum_{t'=0}^{H^{AJ}-1} \sum_{k=1}^K y(k, t + t') \leq 1, \quad \forall t \leq T - H^{AJ} \quad (14)$$

Outflow rate constraints Eqs. (15-16) in the classification yard specify the maximum number of railcars that can be pulled out once by the pull-back engine, limited by the available number of railcars at time $t-1$ on track k , that is, $A(k, t-1) - D(k, t-1)$.

Outflow rate constraints in classification yard:

$$D(k, t) - D(k, t-1) \leq N_{max}^{CT} \times y(k, t), \quad \forall k, t \geq 1 \quad (15)$$

$$D(k, t) - D(k, t-1) \leq A(k, t-1) - D(k, t-1), \quad \forall k, t \geq 1 \quad (16)$$

In short, for the above layer of classification yard, we have used the cumulative flow count-based approach to concisely capture the critically important rules as well as the system dynamic of railcars.

4.4 Assembling task constraints

To optimize the assembling schedule carried out by the pull-back engine, this layer of decisions

needs to determine the starting time and ending time of each assembling job. Again, one assembling job can only pull railcars stored in one classification track, although there may be multiple assembling jobs to form one outbound train. Therefore, feasible destination combinations, train size limit and the departure time of outbound trains have to be jointly considered in this stage. That is, a good schedule of assembling tasks needs to pull out enough railcars from the classification yard into outbound trains with specific destination requirements and scheduled departure time. To reflect the time-dependent flow characteristics of railcars going into outbound trains, we now introduce a time-dependent network representation in Fig. 5 to systematically model the assembling process. For each destination b , the corresponding inflow of each assembling job $f^{in}(b, t)$, as a number of railcars pulled by the pull-back engine, leaves the classification track k (with a predetermined destination b defined by $\varphi(k, b)$).

Flow balance between sorting jobs and assembling jobs:

$$f^{in}(b, t) = \sum_{k=1}^K \{[D(k, t) - D(k, t-1)] \times \varphi(k, b)\}, \quad \forall b, t \geq 1 \quad (17)$$

In the proposed time-expanded framework, each assembling job is considered as a railcar processing arc between time t and time $t+H^{AJ}$. There is another flow balance relationship on each travelling link of the time-expanded network, that is the corresponding outflow of each assembling job $f^{out}(b, t)$. This relationship can be viewed as the assembling job duration time constraints (18).

Assembling job duration constraints:

$$f^{in}(b, t) = f^{out}(b, t + H^{AJ}), \quad \forall b, t \leq T - H^{AJ} \quad (18)$$

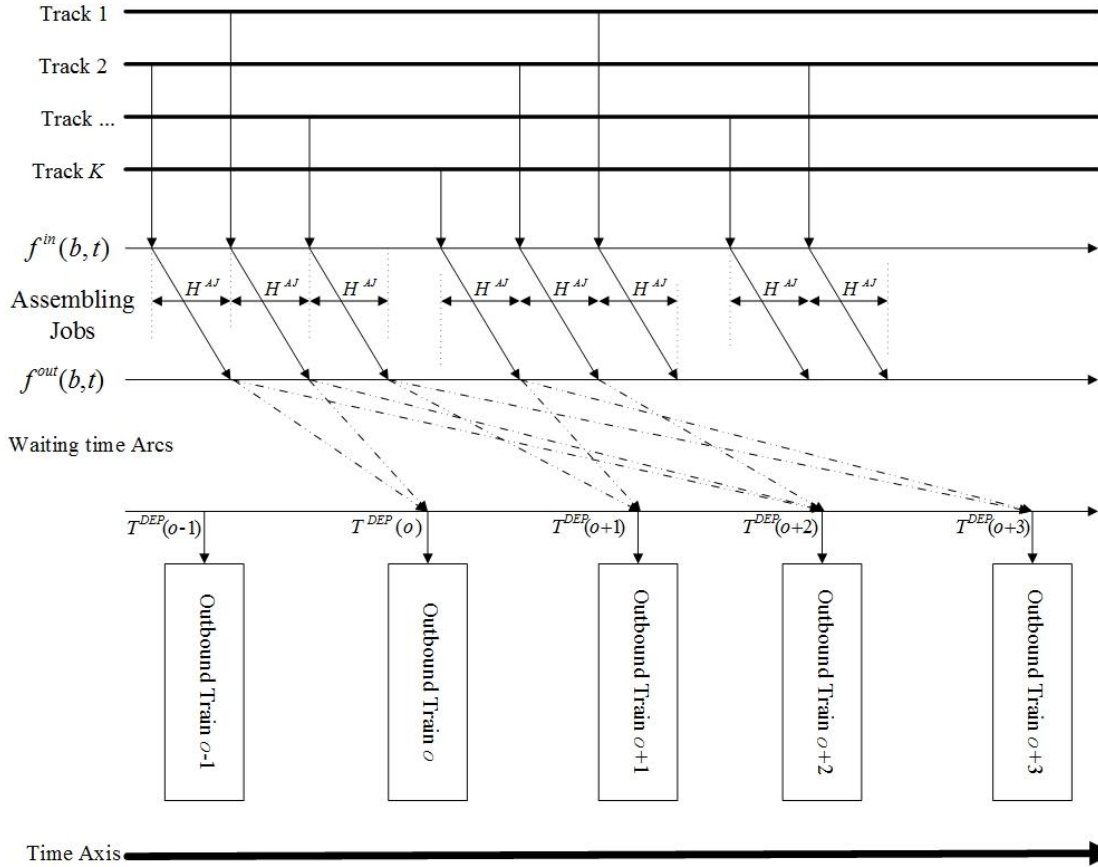


Fig. 5 Illustrative assembling jobs process

4.5 Departure plan of outbound trains

The train departure plan design is a key scheduling task in railroad yards. A good train departure plan can not only decrease the total waiting time of railcars, but also balance the use of railroad yard facilities. In the “waiting time arcs” part of Fig. 5, the forming process of the outbound train plan can be modeled as a time-expanded network as shown in Fig. 6.

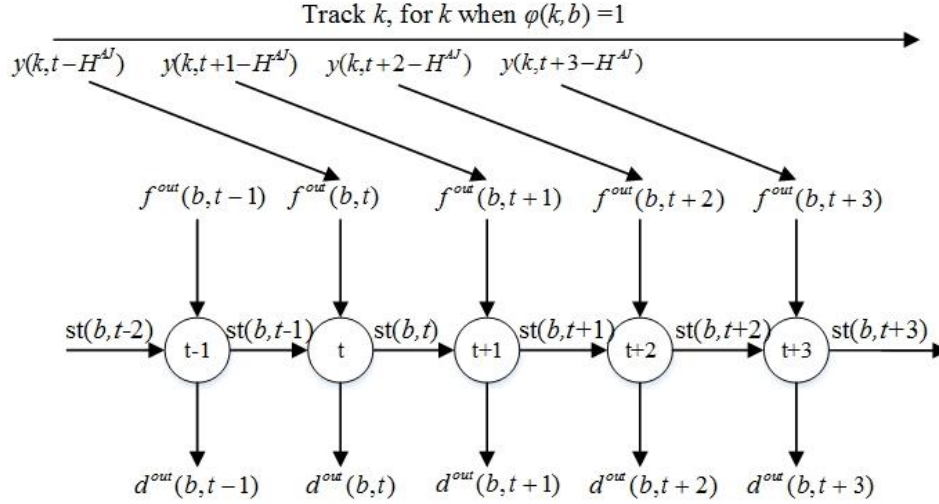


Fig. 6 Time-expanded network structure of assembling jobs for one track

The inflows of the time-expanded network come from the outflow of assembling jobs $f^{out}(b, t)$, which is limited by the corresponding binary variable $y(k, t - H^{AJ})$ when railcars going to destination b are stored in classification track k . For each time point, each classification track has the corresponding flow balance constraints, represented as Eqs. (19-21).

Flow balance constraints in time-expanded network:

$$f^{out}(b, t) + st(b, t - 1) = st(b, t) + d^{out}(b, t), \quad \forall k, t \geq 1 \quad (19)$$

$$f^{out}(b, t) \leq N_{max}^{CT} \times \sum_{k=1}^K [y(k, t - H^{AJ}) \times \varphi(k, b)], \quad \forall k, t \geq H^{AJ} \quad (20)$$

The set of variables $d^{out}(b, t)$ are used to receive the outflow of railcars going to destination b in an outbound train at time t . But the railcars going to destination b are not only accepted by one outbound train. So the variables $d^{out}(b, t)$ should satisfy the constraint in Eq. (21).

Block-to-outbound train constraint:

$$d^{out}(b, t) \leq N_{max}^{OT} \times \sigma(b, t), \quad \forall b, t \quad (21)$$

Outbound train size definitional constraints for number of railcars:

$$n^{OT}(o) = \sum_{b=1}^B [d^{out}(b, t = T^{DEP}(o))], \quad \forall o \quad (22)$$

Eq. (22) defines the outbound train size in terms of the variables $d^{out}(b, t)$, and Eqs. (23-24) specify the maximum and minimum allowed numbers of railcars, where $\mu \leq 1$ is a shortage-allowed coefficient that indicates the shortage allowed degree of the outbound train size. In practice, Eq. (23) is called a “Full Size” rule, while Eq. (24) is called a “Shortage Allowance Size” rule.

$$n^{OT}(o) \leq N_{max}^{OT}, \quad \forall o \quad (23)$$

$$n^{OT}(o) \geq N_{max}^{OT} \times \mu, \quad \forall o \quad (24)$$

5. Valid Inequality Constraints

The above YOP model has a multi-layer space-time network structure, and at its last layer of assembling jobs for outbound trains, we can find some similarities with the lot-sizing model. This section will further extend valid inequality formulations from the lot-sizing problem to improve the strength of the linear programming relaxation for the proposed YOP mixed integer programming model.

5.1 Valid inequality for the lot-sizing model as illustrative example

We first illustrate the concept of valid inequality constraints for the lot-sizing problem proposed by Barany et al. (1984) and Wolsey (1998). Specifically, the goal of the Uncapacitated Lot-Sizing (ULS) problem is to find a minimum cost production plan within time $t = 1, 2, \dots, T$ that satisfies all the nonnegative outflow demand $d(t)$, given to the costs of inflow production $p(t)$, storage flow $h(t)$ and set-up $f(t)$. $x(t)$, $s(t)$ and $y(t)$ donate the production inflow variable, the storage variable and the integer variable at time t , respectively.

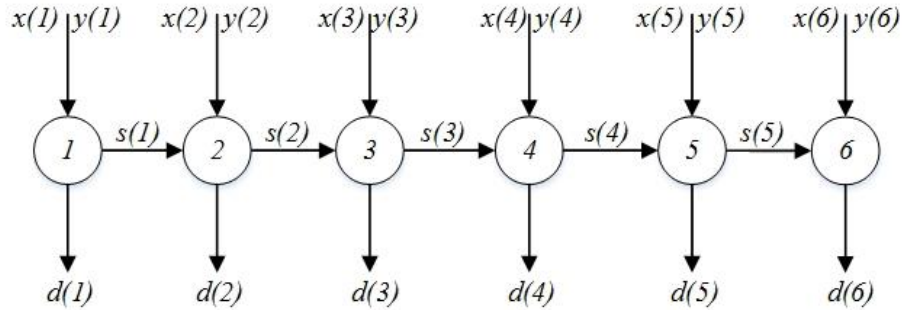


Fig. 7 Time-expanded network structure in the lot-sizing problem

The integer programming model of the lot-sizing problem is listed below:

$$\text{Min} \sum_{t=1}^T p(t) \times x(t) + \sum_{t=1}^T h(t) \times s(t) + \sum_{t=1}^T f(t) \times y(t) \quad (25)$$

$$s(t-1) + x(t) = d(t) + s(t), \quad \forall t \quad (26)$$

$$x(t) \leq \left[\sum_{t'=t}^T d(t') \right] \times y(t), \quad \forall t \quad (27)$$

$$s(0) = 0, s(t), x(t) \geq 0, y(t) \in \{0,1\}, \quad \forall t \quad (28)$$

Eq. (29) is to relax the binary variable $y(t)$ as a linear positive variable to cut off infeasible solution spaces for the integer programming problem. As a result, the values of many variables $y(t)$ become fractional numbers, which are invalid solution values.

$$y(t) \leq 1, \quad \forall t \quad (29)$$

To ensure the integrality of binary variables y from the relaxed model, valid inequality Eq. (30) is introduced to turn more fractional values into integers in the refined solutions.

$$s(t-1) \geq d(t) \times [1 - y(t)], \quad \forall t \quad (30)$$

5.2 Valid inequalities for yard operation problem

In principle, both the lot-sizing problem and assembling job scheduling problem can be modeled through a time-expanded network structure. In this study, we first relax the binary variables $y(k, t)$ as continuous variables with the following range constraint.

$$y(k, t) \leq 1, \quad \forall k, t \quad (31)$$

It should be noted that the time-expanded network structure in the YOP is much more complicated than that in the ULS problem. The YOP considers multi-commodity (i.e. multi-destination) flow, and the inflow $f^{out}(b, t)$ is further limited by the existing number of railcars or the capacity of the corresponding track, compared to the unlimited inflow in ULS. Furthermore, the outflow $d^{out}(b, t)$ is a variable in YOP while the outflow rate in the ULS problem has a fixed demand value.

In a simple case with one single destination for an outbound train, that is, $\sum_{b=1}^B \sigma(b, t) = 1$, we can first develop valid inequalities for the railcar flow of each destination. Similar to Eq. (30) for the ULS problem, we can develop valid inequality Eq. (32) for a single-destination-type outbound train plan.

$$st(b, t-1) \geq \mu \times N_{max}^{OT} \times \sigma(b, t) \times \left\{ 1 - \sum_{k=1}^K [y(k, t - H^{AJ}) \times \varphi(k, b)] \right\}, \quad \forall b, t \geq H^{AJ} \quad (32)$$

The inequality Eq. (32) can be interpreted as follows: If the lump sum of inflow binary variable $y(k, t)$ is zero, then the volume of stored flow $st(b, t-1)$ for a given destination b should be sufficient to meet the production demand $\mu \times N_{max}^{OT}$ in the outbound train.

If there are multiple destinations in an outbound train, railcar flows of different destinations merge at a special timestamp t in the time-expanded network when they are going into outbound trains. The assignment of railcars of different destinations in one outbound train further influences the effect of the valid inequalities as mentioned above. We can accordingly develop the valid inequality Eq. (33) for a multiple-destination plan. The proposed two valid inequalities Eqs. (32-33) can be easily coded in standard optimization solvers, and both formulas are expected to cut off infeasible fractional values for binary decisions $y(k, t)$ related to assembling jobs, while their effectiveness needs to be systematically examined in different cases through numerical experiments.

$$\sum_{b=1}^B [st(b, t-1) \times \sigma(b, t)] \geq \mu \times N_{max}^{OT} \times \lambda(t) \times \left\{ 1 - \sum_{b=1}^B \sum_{k=1}^K [\sigma(b, t) \times y(k, t - H^{AJ}) \times \varphi(k, b)] \right\}, \quad \forall t \geq H^{AJ} \quad (33)$$

6. Assignment-based heuristics for determining incoming train humping sequence

To reduce the solution search space of the original problem, we propose an assignment-based heuristic approach and related rules for determining incoming train humping sequences. By using a set of aggregated classification yard constraints, we aim to narrow down the humping precedence relationship of inbound trains based on a few technical heuristics rules. The humping sequence considerations from the assignment result will be applied to the original YOP model as additional constraints on the humping sequence variables $s(i, j)$ to reduce the search space. It should be noted that these additional constraints are built on a simplified model representation, so it is still possible that those newly introduced restrictions could cut off exact optimal solutions. Thus, one needs to carefully construct heuristic rules to systematically balance the needs for (i) significantly reducing the solution space through practical technical considerations and (ii) keeping potentially close-to-optimal solution candidates to be further examined in the full optimization model.

From a project scheduling perspective (Malakooti, 2013), in a single machine case, one can use various heuristic rules, such as Shortest Processing Time (SPT) or Earliest Due Date (EDD). The

commonly used EDD rule uses an order of non-decreasing due dates to sort the sequence of remaining jobs. However, in our complex YOP context, each inbound train has multiple “due dates” associated with different connected outbound trains, and the processing times of railcars are difficult to determine as they are related to various bottlenecks in the yard operations. To construct effective heuristic rules, we first determine the possible connections between inbound and outbound trains through an aggregated flow-based assignment model, and identify the earliest due dates across all outbound trains to determine the humping sequence.

6.1 Constructing flow assignment model based on a simplified representation

We first introduce a set of variables $w(i, b, o)$ to represent the railcar flow volume of destination b from inbound train i to outbound train o . Accordingly, the objective function aims to minimize the total “assignment cost” in terms of the processing time $(T^{DEP}(o) - T^{ARR}(i))$ for the railcar flow $w(i, b, o)$. Recall that the FIFO (First-In-First-Out) rule is widely used in practice to determine the humping sequences, especially for trains that share similar or the same destination b sets for railcars. One can construct an objective function $\sum_{i=0}^I \sum_{b=0}^B \sum_{o=0}^O w(i, b, o) \times (T^{DEP}(o) - T^{ARR}(i))$, and add penalties for excessively large processing times through a quadratic cost function in Eq. (34) in the following flow assignment model.

$$\text{Min } W' = \sum_{i=0}^I \sum_{b=1}^B \sum_{o=1}^{O+1} w(i, b, o) \times (T^{DEP}(o) - T^{ARR}(i))^2 \quad (34)$$

s.t.

Flow conservation constraints of railcars from inbound trains:

$$\sum_{o=1}^{O+1} w(i, b, o) = N^{IT}(b, i), \quad \forall b, i \quad (35)$$

Lower bound constraints for the number of railcars assigned to outbound trains:

$$\sum_{i=0}^I \sum_{b=1}^B w(i, b, o) \geq N_{max}^{OT} \times \mu, \quad \forall o \quad (36)$$

Upper bound constraints for the number of railcars assigned to outbound trains:

$$\sum_{i=0}^I \sum_{b=1}^B w(i, b, o) \leq N_{max}^{OT}, \quad \forall o \quad (37)$$

Time duration constraint for feasible inbound-to-outbound train flow assignment:

$$w(i, b, o) \times (T^{DEP}(o) - T^{ARR}(i) - H^{TI} - H^{AJ} - H^{HJ}(i)) \geq 0, \quad \forall i, b, o \quad (38)$$

Block-to-outbound train constraint:

$$\sum_{i=0}^I w(i, b, o) \leq N_{max}^{OT} \times \sigma(b, t = T^{DEP}(o)), \quad \forall b, o \leq O \quad (39)$$

$$w(i, b, o) \geq 0, \quad \forall i, b, o \quad (40)$$

Eq. (35) ensures the total flow conservation of railcars from inbound trains, while Eqs. (36-37) defines the upper bound and lower bound limits for the size of outbound trains. The time duration requirements of essential operations are enforced by Eq. (38). That is, if the minimal travel time requirement is not met, $T^{DEP}(o) - T^{ARR}(i) - H^{TI} - H^{AJ} - H^{HJ}(i) < 0$, then the flow assignment from train i to train o must be zero (as a result of the nonnegative flow constraint $w(i, b, o) \geq 0$). If the time difference between train i and train o can satisfy the minimal travel time requirement, then the corresponding flow assignment values can be zero or positive. Eq. (39) restricts the railcars

of different destinations which are acceptable by the corresponding outbound trains. As the train arrival time and departure time $T^{ARR}(i)$ and $T^{DEP}(o)$ are given, our constructed optimization model has a linear objective function and a set of linear constraints, so it can be easily solved.

With a planning horizon denoted as $t = 0, 1, \dots, T$, we need to handle the boundary conditions for railcars already on the yard at time $t = 0$ and possible remaining railcars after $t = T$. Accordingly, we introduce two dummy trains, namely inbound train “0” with $T^{ARR}(i = 0) = 0$, and outbound train “ $O+1$ ” with $T^{DEP}(o = O + 1) = T$, where O is the total number of all original outbound trains. Those dummy trains can accept railcars of all destinations without loss of generality. Therefore, Eq. (39) is applied only for physical outbound trains through the condition of $o \leq O$.

6.2 Earliest Due Date-based heuristic rules for determining humping sequences

While the earliest due date is a commonly used heuristic rule for machine scheduling, the due date for each inbound train in the YOP is not clearly defined. Based on the aggregated flow assignment result, we can first find the railcar flow connections between inbound trains and outbound trains, then accordingly, for each inbound train, use the earliest departure time of the related outbound trains as the “earliest require time”. This can be viewed as an approximate to the “earliest due date” measure, which can help to determine the precedence relationship of humping jobs to some extent. In the following, we first describe an algorithm for determining the earliest required time in the heuristic rules, and then we present a sample solution based on the illustrative example in Fig. 8.

Algorithm for determining humping sequence

Step 1: Find the earliest outbound train for each inbound train (excluding dummy trains)

Obtain all feasible assignment for $w(i, b, o)$. For each inbound train i , find the earliest outbound train with $w(i, b, o) > 0$, denoted as $EO(i)$, with the departure time of $T^{DEP}(o = EO(i))$.

Step 2: Identify the earliest required time for outbound railcar flow

Consider a pair of inbound trains i and j , where $i < j$. Compare the related earliest departure time from assigned flow at the corresponding outbound trains, denoted as Earliest Required Time $ERT(i)$, $ERT(j)$, where $ERT(i) = T^{DEP}(o = EO(i))$ and $ERT(j) = T^{DEP}(o = EO(j))$.

Step 3: Determine humping sequence based on earliest required time

Apply the following rules:

If $ERT(i) < ERT(j)$, that is, train i 's earliest departure time for its assigned outbound trains is earlier than the earliest departure time for the assigned outbound trains for train j , then inbound train i should be humped before train j . This sets the humping sequence variable as $s(i, j) = 1$.

Similarly, if $ERT(i) > ERT(j)$, then $s(i, j) = 0$, otherwise, we have $ERT(i) = ERT(j)$, where trains i and j have their earliest assigned railcar flow to the same outbound train. In this case, we cannot simply determine the sequence and leave it to the full optimization model for further selections. The above rules can be summarized as Eq. (41).

$$s(i, j) = \begin{cases} 0, & ERT(i) > ERT(j) \\ 1, & ERT(i) < ERT(j) \\ \text{uncertain}, & ERT(i) = ERT(j) \end{cases} \quad (41)$$

An illustrative example

Fig. 8 describes the aggregated flow assignment relationship from 4 inbound trains to 4 outbound trains. For each inbound train, the dashed line shows the flow going to the corresponding earliest outbound train, such as $EO(i - 1) = o - 1$, $EO(i) = o - 1$, $EO(i + 1) = o + 2$ and $EO(i + 2) = o$. One can easily verify that the Earliest Required Times for each inbound train are $ERT(i - 1) = T^{DEP}(o - 1)$, $ERT(i) = T^{DEP}(o - 1)$, $ERT(i + 1) = T^{DEP}(o + 2)$ and $ERT(i + 2) = T^{DEP}(o)$.

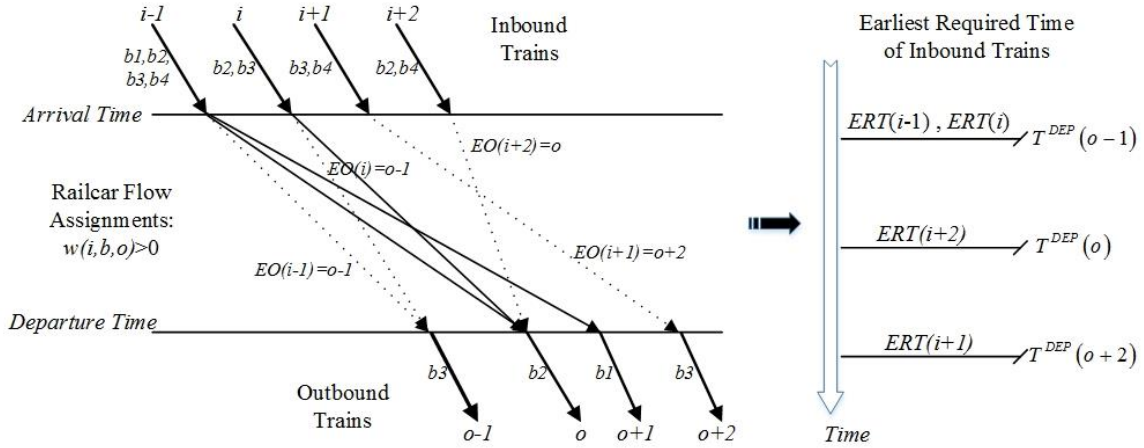


Fig. 8 Determine the humping sequence process by heuristic rules

Finally, compare all $ERT(i)$ to determine the values of humping sequence variables by using Eq. (41), and the results are shown in Table 8. For example, since $ERT(i-1) = ERT(i)$, $ERT(i) < ERT(i+1)$ and $ERT(i+1) > ERT(i+2)$, the values of $s(i-1, i)$, $s(i, i+1)$ and $s(i+1, i+2)$ are uncertain, 1 and 0, respectively.

Table 8 Example solution for humping sequence variables $s(i, j)$

$s(i, j)$	i	$i+1$	$i+2$
$i-1$	Uncertain	1	1
i	—	1	1
$i+1$	—	—	0
$i+2$	—	—	—

“—”: the corresponding variable does not exist

7. Numerical Experiments

This section aims to test the computational efficiency and effectiveness of the proposed mixed integer programming models (IP), including the linear relaxed integer programming model (RP) and the relaxation model with valid inequalities (RP+VI), as well as the model that embeds proposed heuristic rules (HR) to determine the humping sequence. We use commercial optimization software GAMS with the CPLEX solver to obtain the solutions for different cases on a computer with a 2.69 GHz processor and 8 GB RAM. GAMS (Brooke et al., 2006) is a high-level modeling system for mathematical programming and optimization which enables us to easily implement different sets of valid inequality constraints. As the solvers might fail to solve large-scale instances within a reasonable amount of CPU time, for consistency, the termination rules for all models are setup with a solver relative gap of 0.0005 (i.e. 0.05%) in GAMS, while a smaller solution gap could be used for relatively small problem instances.

We consider a hypothetical yard with 13 classification tracks, and this yard needs to process trains consisting of railcars of 13 different destinations. The incoming train data are adapted from the INFORMS RAS problem solving competition (RAS 2013), using data from day 3 in data set no. 2. Specifically, Tables 9 and 10 show the input parameter values and the Block-to-Track plan, respectively. In Table 11, we list 7 different test cases, corresponding to a range of 8 to 20 inbound trains, 8 to 20 outbound trains, and 541 to 1451 railcars in the planned horizons from 465 minutes to 750 minutes, respectively. The technical inspection time is assumed to be 30 minutes in order to consider tighter inbound train arrival time intervals. As the RAS data set does not provide the

outbound train departure times and the designated destination combination, we generate the outbound train timetables, each outbound train carries railcars with a single destination for simplicity, and the destination split of outbound trains is largely proportional to the destination split from the inbound trains.

Consider the 3rd day's schedule of RAS dataset no. 2, with the first two days being used as the warm-up period. As shown in Table 10, the existing cars at time $t=0$ at the beginning of the 3rd day are generated from the first 2 days using simulation with simple priority rules.

Table 9 Input Parameters of Yard Operations Plan

Symbol	Description	Value
B	number of destinations	13
K	number of classification tracks	13
H^{TI}	technical inspection duration time (min)	30
H^{HI}	minimum humping jobs interval (min)	10
H^{AJ}	one assembling job duration time for each track (min)	10
N_{max}^{CT}	spatial capacity of classification track	125
N_{max}^{OT}	maximum number of railcars in a single outbound train	120
μ	shortage allowance coefficient of outbound train size	0.25
M	a large positive number associated with the artificial variables for if-then constraints	10000

Table 10 Block-to-track plan defined by $\varphi(k, b)$ and the initial number of railcars $A(k, t=0)$

$\varphi(k, b)$	$k1$	$k2$	$k3$	$k4$	$k5$	$k6$	$k7$	$k8$	$k9$	$k10$	$k11$	$k12$	$k13$
$b1$	1	0	0	0	0	0	0	0	0	0	0	0	0
$b2$	0	1	0	0	0	0	0	0	0	0	0	0	0
$b3$	0	0	1	0	0	0	0	0	0	0	0	0	0
$b4$	0	0	0	1	0	0	0	0	0	0	0	0	0
$b5$	0	0	0	0	1	0	0	0	0	0	0	0	0
$b6$	0	0	0	0	0	1	0	0	0	0	0	0	0
$b7$	0	0	0	0	0	0	1	0	0	0	0	0	0
$b8$	0	0	0	0	0	0	0	1	0	0	0	0	0
$b9$	0	0	0	0	0	0	0	0	1	0	0	0	0
$b10$	0	0	0	0	0	0	0	0	0	1	0	0	0
$b11$	0	0	0	0	0	0	0	0	0	0	1	0	0
$b12$	0	0	0	0	0	0	0	0	0	0	0	1	0
$b13$	0	0	0	0	0	0	0	0	0	0	0	0	1
$A(k, t=0)$	9	3	15	2	3	2	6	3	10	7	16	3	2

Table 11 Problem size in different test cases

Case ID	Number of Railcars	Number of Inbound Trains	Number of Outbound Trains	Length of Planned Horizon (minutes)
1	541	8	8	465
2	651	10	10	539
3	781	12	12	578
4	971	14	14	619
5	1151	16	16	656
6	1301	18	18	690
7	1451	20	20	750

7.1 Setup of different models

In our proposed YOP models, there are three groups of binary variables, $s(i, j)$, $x(i, t)$ and $y(k, t)$. To examine the computational efforts for different relaxation methods and the proposed valid inequalities, we construct a number of optimization models as shown below:

- (1) IP: the original integer programming model;
- (2) RP-A: the corresponding linear programming model with all binary variables relaxed as linear variables within domain $[0, 1]$;
- (3) RP-Y: the mixed integer programming model with the binary variables $y(k, t)$ relaxed as linear variables within domain $[0, 1]$;
- (4) RP-X: the mixed integer programming model with the binary variables $x(i, t)$ relaxed as linear variables within domain $[0, 1]$;
- (5) RP-S: the integer programming model with the binary variables $s(i, j)$ relaxed as linear variables within domain $[0, 1]$;
- (6) IP+VI: IP with proposed valid inequality
- (7) RP-Y+VI: RP-Y with proposed valid inequality
- (8) IP+HR: the integer model with heuristic rules

We investigate the relaxed optimization models to examine which set of constraints are difficult to solve, so we can better design the corresponding heuristic rules to reduce the search space. The numbers of constraints, variables and integer variables of the IP models can first be summarized in Table 12.

Table 12 Problem size and complexity of IP models.

Case ID	# of constraints	# of variables	# of integer variables	# of $s(i, j)$	# of $x(i, t)$	# of $y(k, t)$
1	69726	40129	9814	28	3728	6058
2	81221	47596	12465	45	5400	7020
3	87793	52213	14541	66	6948	7527
4	94754	57174	16831	91	8680	8060
5	101243	61927	19173	120	10512	8541
6	107383	66544	21574	153	12438	8983
7	117392	73849	24973	190	15020	9763

7.2 Computational time and solution quality of the relaxation models

The GAMS outputs a “solver relative gap” measure, which corresponds to the final termination condition in terms of the relative gap between the current objective value and the possible optimal objective value reported. Specifically, “solver relative gap” equals 0 when the model obtains the optimal solution. The maximal limit on the computational time for all models is set as 4000 seconds through GAMS.

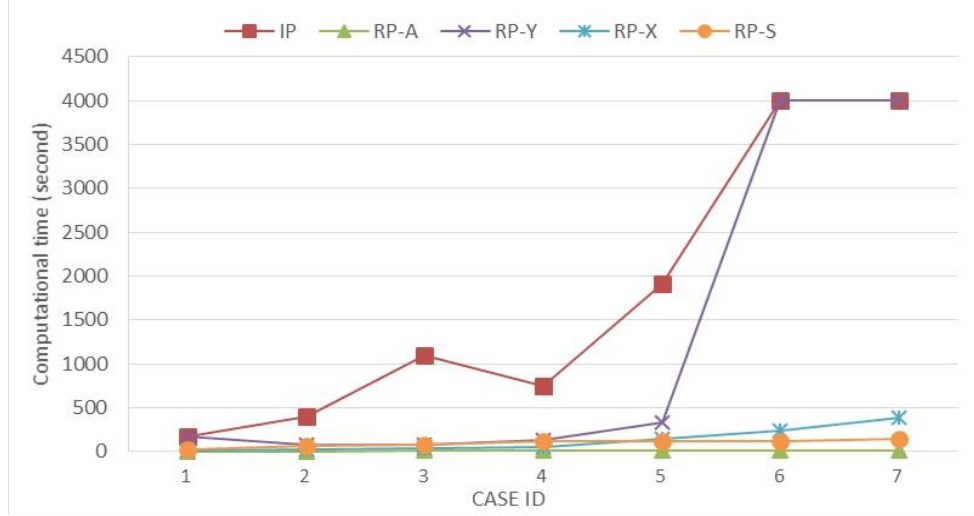


Fig. 9 Computational time comparison between IP models and relaxation models

Fig. 9 shows that the increase in problem size leads to significantly longer computational time, especially for the IP and RP-Y models. The IP models and the related relaxation model can all solve cases 1 to 5 to optimality. Due to the computational time limit of 4000 seconds, the IP, RP-Y models in cases 6 and 7 were terminated before obtaining optimal solutions. Overall, sophisticated models IP and RP-Y need more computational time compared to relaxation models with simpler forms such as RP-A, RP-X and RP-S. The IP model terminates with a solver relative gap of 0 % for the first 5 cases.

Across all relaxation models, the RP-A model can be easily solved as all integrality constraints are relaxed, and the RP-Y model requires more computational time, which indicates that the sequencing decision/constraints associated with variables $s(i, j)$ and $x(i, t)$ are in fact very complex and “hard” to optimize in the current experimental setting. This observation motivates us to develop effective humping sequence rules to reduce the search space associated with binary variable $s(i, j)$.

7.3 Effectiveness of valid inequalities

To evaluate the effectiveness of the proposed valid inequality method (VI), we conduct experiments to compare the computational times between IP and IP+VI models, as shown in Fig. 10.

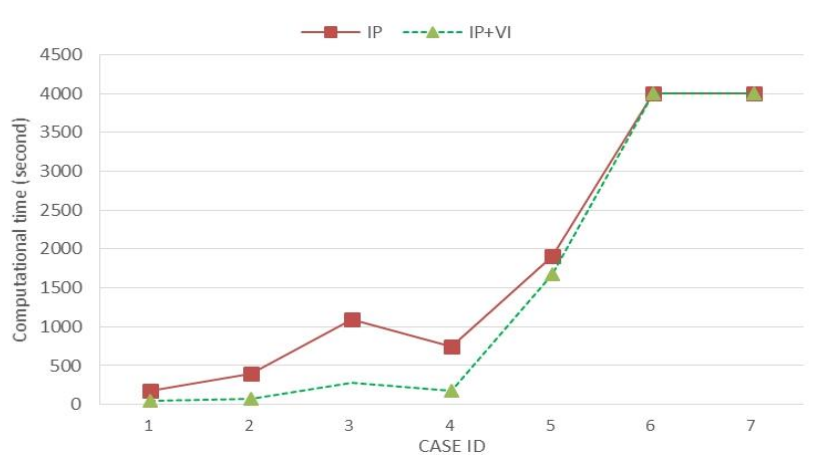


Fig. 10 Computational time comparison between IP models and relaxation models

When the problem size increases, the computational time required by the IP+VI model still significantly increases, but in general it takes much less CPU time compared to the original IP model. The valid inequality can save computational efforts, by 63.4% on average, to obtain the optimal solution in small cases 1 to 5. In the larger cases 6 and 7, though, the effectiveness of the VI method is still limited as it exceeds the given computational time limit and still has difficulty solving the resulting large IP models.

Table 13 describes the number of $y(k, t)$ being integral and fractional values in the RP-Y and RP-Y+VI models. In general, the valid inequalities significantly reduce the number of the fractional variable values in the original RP-Y solutions. None of the fractional variable values exist in the RP-Y+VI solutions except in case 7, as no feasible solution is found within the computational time limit. The results indicate the valid inequality can effectively ensure the integrality constraints associated with variables $y(k, t)$.

Table 13 Effectiveness of valid inequality in terms of cutting off fractional values of $y(k, t)$.

Case ID	Model	# of $y(k, t) = 0$	# of fractional values $0 < y(k, t) < 1$	# of $y(k, t) = 1$
1	IP	6042	0	16
	RP-Y	6050	8	0
	RP-Y+VI	6050	0	8
2	IP	7006	0	14
	RP-Y	7010	10	0
	RP-Y+VI	7010	0	10
3	IP	7496	0	31
	RP-Y	7515	12	0
	RP-Y+VI	7513	0	14
4	IP	8034	0	26
	RP-Y	8046	14	0
	RP-Y+VI	8046	0	14
5	IP	8511	0	30
	RP-Y	8525	16	0
	RP-Y+VI	8525	0	16
6	IP	8935	0	48
	RP-Y	8965	18	0
	RP-Y+VI	8965	0	18
7	IP	9720	0	43
	RP-Y	9743	20	0
	RP-Y+VI	—	—	—

“—”: no value

7.4 Effectiveness of heuristic rules

In the IP+HR model, the aggregated flow assignment models are first constructed for each case to obtain the railcar flow results. As the aggregated flow assignment models are linear models, the computational times of all cases are all less than 0.05 seconds. Then, we calculate the values of $s(i, j)$ through Eq. (41). The number of obtained $s(i, j)$ values of in each model are shown in Table

14.

Table 14 Counts of sequencing variable $s(i, j)$ for different cases

Case ID	# of $s(i, j)$ fixed through HR	Total # of $s(i, j)$	Percentage of $s(i, j)$ being fixed (%)
1	17	28	60.71
2	33	45	73.33
3	54	66	81.82
4	78	91	85.71
5	107	120	89.17
6	140	153	91.50
7	177	190	93.16

After adding constraints associated with the fixed humping sequence $s(i, j)$ into the IP model, the solution quality and computational time of IP+HR models for different cases are compared with the IP model in Table 15 and Fig. 11. For all solvable cases 1-5 by the IP model, the IP+HR model obtains the same objective values for cases 1 to 4, and has a very small solution gap of 0.09% in case 5.

Table 15 Comparison between IP and IP+HR models for all cases

Case ID	Model	Objective value	Solution quality in terms of percentage $(Z^{HR}-Z^{IP})/Z^{IP}$ (%)
1	IP	137331	0
	IP+HR	137331	0
2	IP	168786	0
	IP+HR	168786	0
3	IP	195393	0
	IP+HR	195393	0
4	IP	235658	0
	IP+HR	235658	0
5	IP	272136	0
	IP+HR	272385	-0.09

As shown in Fig. 11, the IP+HR model can significantly reduce computational time compared to the original IP model, with runtime decreasing by 14.4%, 24.9%, 49.9%, 72.4%, 84.7%, respectively, for cases 1-5. For large cases 6 and 7, the proposed heuristic rules can obtain feasible IP solutions within the time limit. Overall, the proposed heuristic rules can reach a good balance in obtaining optimal solutions and reducing computational search efforts for the complex YOP model.

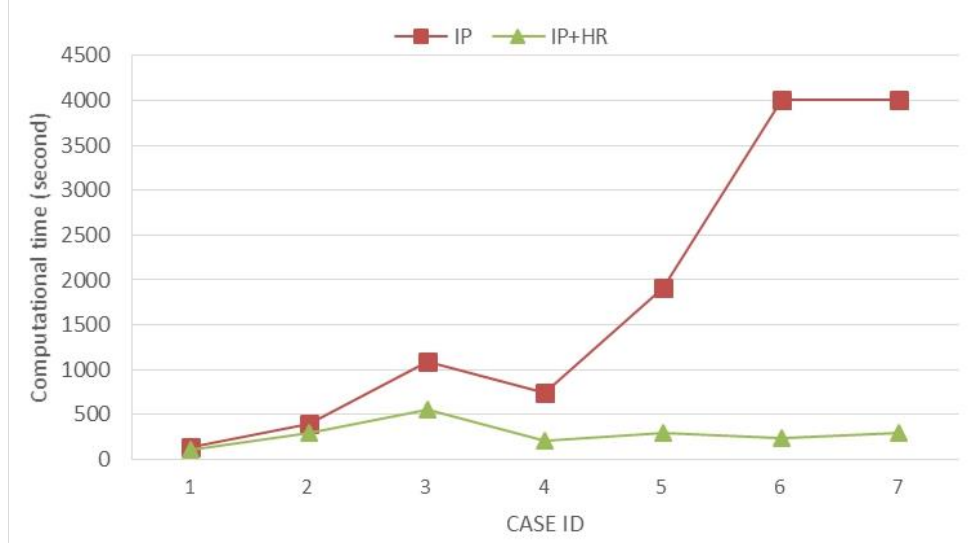


Fig. 11 Computational time comparison between IP and IP+HR models

8. Concluding remarks

Designing optimal yard operations plans is a critical decision-making problem in railroad industries. This paper presents a set of theoretically rigorous mixed integer programming models for formulating and solving the YOP problem. We specifically introduce the cumulative flow count to capture the activities of railcars arriving at and departing from the yard system, and we show how the cumulative flow count can describe the yard operations as a system of queues with waiting time/processing time as optimization criteria, subject to tight spatial capacity constraints, and essential discharge rate constraints and the processing time constraints.

In this study, the time-expanded network representation is also applied to build a systematic model for the assembling job process as interconnected layers in a railroad yard. By considering the departure layer of the YOP as a special lot-sizing problem, we further develop valid inequality constraints. To further reduce the search space associated with binary humping sequence variables, we adopt specific heuristic rules based on an aggregated assignment representation. These heuristic rules hold the promise of dramatically improving the computational efficiency by using domain knowledge and insight. In this study, a number of numerical experiments of different cases are used to verify the effectiveness of our proposed valid inequalities and heuristic rules through systematic comparisons between different models.

The problem under consideration assumes that inbound and outbound trains operate under given train schedules. Thus, it is important to examine different ways for reducing the system waiting time given the fixed schedules. First, the major part of the system waiting time is due to the time gap between timetables of inbound and outbound trains, subject to the destination mix compatibility constraints for railcars, as illustrated as our proposed heuristic algorithm for aggregated assignment model. Second, the maximum number of railcars in each outbound train could also impact the cumulative departure rates of railcars (with different destinations), as well as the system-wide waiting time performance. To further reduce the system waiting time for a yard with various capacity bottlenecks, we need to design efficient operational plans for humping and pull-back engines (with limited processing capacities) to make the required flow connections between inbound and outbound schedules, subject to spatial capacity in classification tracks and other processing time requirements.

In our paper, we assume assembling trains in the departure yard and a constant time duration for one assembling job by a pull-back engine. The processing time durations of pull-back engines are dependent on the traveling distance of pull-back engines and the (partial) block size every time it

carries a block to the departure yard. One can reduce or even eliminate train assembling jobs in the departure yard by better preparing railcars as large groups (to the same destination) in the classification yard so that we can pull out a group of railcars from the classification tracks directly to form an outbound train. For simplicity, we also use a constant processing time for pull-back engines, similar to the assumption used in the INFORMS RAS problem solving competition 2013, based on the following assumed typical conditions. We consider a running speed of a pull-back engine in the yard as 20~30 km/hour, and the travelling distance of one assembling job (back and forth) as about 5 km. Accordingly, the average travelling time of one pull-back engine for each assembling job around 10~15 minutes. In the future study, we could consider more practical processes and factors with a specific shunting yard to capture all essential constraints in the assembling process.

We also have the following remarks on the shortage allowance parameter used in this study. First, the YOP problem under consideration assumes (1) a feasible locomotive cycle plan is available for all planned freight trains, and (2) each outbound train runs exactly according to the schedule. In practice, when there is insufficient railcars to transport, the corresponding outbound train could be cancelled or delayed, by systematically considering the impact of reducing locomotive operational costs and changes to the existing locomotive cycle plans. Our future study should further introduce a flexible shortage allowance parameter and allow train service cancellation/delay in order to improve the overall rail system operational efficiency.

In our future research, we need to further consider many practically important constraints and considerations to construct an operational yard optimization plan. The following modeling issues will be further investigated:

(1) As the railroad yard operation plan is a complex systematic problem involving many factors, the proposed YOP focuses on how to handle the deterministic problem with fixed train schedules, fixed processing time for operation and given inbound train composition. Ideally, many of the above factors should be considered as random parameters in a stochastic or robust programming framework to handle various sources of uncertainty.

(2) At the current modeling stage, some operations closely related to the main process of YOP are not considered or incorporated, for example, sorting of cars, locomotive cycle plans, multiple alternative departure time decisions of outbound trains, technical maintenance plans of damaged railcars, movement plans of loading and unloading railcars between railroad yard and freight stations, the sequencing requirement of railcars assembled in outbound trains. A more practically useful YOP model should be further developed with the above constraints being modeled systematically.

(3) In this YOP study, the “sorting by block” strategy is adopted as the block-to-track plan. A multiple and dynamic block-to-track plan is necessary for practical operations. Therefore, we will further investigate methods for utilizing a day-dependent block-to-track plan

(4) The presented YOP model is a complex integrated model which can solve both the single-destination-type plan and the multiple-destination-type plan. The valid inequalities for the single-destination-type plan have been specifically verified in this study. In future research, specialized valid inequalities and related computationally tractable algorithms for the multiple-destination-type plan will be developed.

(5) In this paper, we adopt relaxation and valid inequality techniques to solve the YOP model. Other advanced optimization and integer programming technologies, such as Lagrangian relaxation and Dantzig-Wolfe decomposition will be studied to solve larger-scale instances.

Acknowledgements

Special thanks to anonymous reviewers for their constructive and insightful comments. The first author would like to thank the support from the China Scholarship Council and the Innovative Practice Program for Graduate Students of Southwest Jiaotong University. Our study has also

benefited from great comments from Dr. Qiyuan Peng from Southwest Jiaotong University, China, Dr. Shiwei He, Dr. Jinjin Tang, and Dr. Yuguang Wei from Beijing Jiaotong University, China, as well as Jeffrey Taylor from University of Utah. The second author is one of the problem owners in 2013 INFORMS Railway Application Section Problem Solving Competition, and the authors would also like to thank the other RAS problem solving competition committee members for their contributions and feedback on the related problem statement. The authors are, of course, responsible for all results and opinions expressed in this paper.

References

- Assad, A. A., 1980. Models for rail transportation. *Transportation Research Part A: General*, 14(3), 205-220.
- Barany, I., Van Roy, T. J., Wolsey, L. A., 1984. Strong formulations for multi-item capacitated lot sizing. *Management Science*, 30(10), 1255-1261.
- Barnhart, C., Jin, H., Vance, P. H., 2000. Railroad blocking: A network design application. *Operations Research*, 48(4), 603-614.
- Beckmann, M., McGuire, C. B., Winsten, C. B., 1956. *Studies in the Economics of Transportation*. Yale University Press: New Haven, 113-171.
- Bohlin, M., Flier, H., Maue, J., Mihal'ak, M., 2011a. Hump yard track allocation with temporary car storage. *Proc. of the 4th International Seminar on Railway Operations Modelling and Analysis*, Rome, Italy.
- Bohlin, M., Flier, H., Maue, J., Mihal'ak, M., 2011b. Track allocation in freight-train classification with mixed tracks. In *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 20 of *Open Access Series in Informatics*, 38-51, Dagstuhl, Germany.
- Bohlin, M., Dahms, F., Flier, H., Gestrelus, S., 2012. Optimal freight train classification using column generation. In D. Delling and L. Liberti, editors, *12th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, volume 25 of *Open Access Series in Informatics*, 10-22, Dagstuhl, Germany.
- Boysen, N., Fliedner, M., Jaehn, F., Pesch, E., 2012. Shunting yard operations: Theoretical aspects and applications. *European Journal of Operational Research*, 220(1), 1-14.
- Brooke, A., Kendrick, D., Meeraus, A., & Raman, R., 2006. *GAMS-Language Guide*. GAMS Development Corporation. Washington D. C..
- Cassidy, M. J., 1999. Traffic Flow and Capacity. *Handbook of Transportation Science*, edited by R. W. Hall, Kluwer Academic, Boston, 155-190.
- Cordeau, J. F., Toth, P., Vigo, D., 1998. A survey of optimization models for train routing and scheduling. *Transportation Science*, 32 (4), 380-404.
- Czyzyk, J., Mesnier, M., Mor, J., 1998. The NEOS Server. *IEEE Journal of Computational Science and Engineering*, 5, 68-75.
- Daganzo, C. F., Dowling, R.G., Hall, R.W., 1983. Railroad classification yard throughput: The case of multistage triangular sorting. *Transportation Research Part A*, 17(2), 95-106.
- Daganzo, C. F., 1986. Static blocking at railyards: Sorting implications and track requirements. *Transportation Science*, 20(3), 189-199.
- Daganzo, C. F., 1987a. Dynamic blocking for railyards: Part I. homogeneous traffic. *Transportation Research Part B*, 21(1), 1-27.
- Daganzo, C. F., 1987b. Dynamic blocking for railyards: Part II. heterogeneous traffic. *Transportation Research Part B*, 21(1), 29-40.
- Dahlhaus, E., Horak, P., Miller, M., Ryan, J.F., 2000a. The train marshalling problem. *Discrete Applied Mathematics*, 103(1-3), 41-54.

- Dahlhaus, E., Manne, F., Miller, M., Ryan, J., 2000b. Algorithms for combinatorial problems related to train marshalling. *Proceeding 11th Australasian Workshop Combinatorial Algorithms (AWOCA-00)*, 7-16, Hunter Valley, Australia.
- Gestrelus, S., Dahms, F., Bohlin, M., 2013. Optimisation of simultaneous train formation and car sorting at marshalling yards. In *5th International Seminar on Railway Operations Modelling and Analysis Rail Copenhagen*, Copenhagen, Denmark. Available at <http://soda.swedish-ict.se/5531/>
- He, S., Song, R., Chaudhry, S. S., 2000. Fuzzy dispatching model and genetic algorithms for railyards operations. *European Journal of Operation Research*, 124(2), 307-331.
- He, S., Song, R., Chaudhry, S. S., 2003. An integrated dispatching model for rail yards operations. *Computers & Operations Research*, 30(7), 939-966.
- INFORMS, Railway Application Section, Problem Solving Competition (2013). Railroad Yard Operational Plan. <https://www.informs.org/Community/RAS/Problem-Repository>
- Jacob, R., Marton, P., Maue, J., Nunkesser, M., 2007. Multistage methods for freight train classification. *Proceedings of the 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS 2007)*, 158-174. IBFI Schloss Dagstuhl Seminar proceedings, Sevilla, Spain.
- Jacob, R., Marton, P., Maue, J., Nunkesser, M., 2011. Multistage methods for freight train classification. *Networks*, 57(1), 87-105.
- Li, H., He, S., Wang, B., Shen, Y., 2011. Survey of stage plan for railway marshalling station. *Journal of the China Railway Society*, 33(8), 13-22.
- Lin, E., Cheng, C., 2009. YardSim: A rail yard simulation framework and its implementation in a major railroad in the U.S.. *WSC '09 Winter Simulation Conference, SIGSIM ACM Special Interest Group on Simulation and Modeling*, pp. 2532-2541, Austin, USA.
- Lin, E., Cheng, C., 2011. Simulation and analysis of railroad hump yards in North America. In *Proceedings of the Winter Simulation Conference*, pp. 3715-3723. Winter Simulation Conference, Atlanta, USA.
- Makagami, Y., Newell, G.F., Rothery, R., 1971. Three-dimensional representation of traffic flow. *Transportation Science*, 5(3), 302-313.
- Malakooti, B., 2013. *Operations and production systems with multiple objectives*. John Wiley & Sons, New York, NY, 803-807.
- Marton, P., Maue, J., Nunkesser, M., 2009. An improved train classification procedure for the hump yard Lausanne Triage. *Proceeding 9th Workshop Algorithmic Methods Models Optimal Railways (ATMOS-09)*, IBFI Schloss Dagstuhl, Wadern, Germany.
- Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an N-track network: A model reformulation with network-based cumulative flow variables. *Transportation Research Part B*, 67, 208-234.
- Newell, G. F., 1982. *Applications of Queueing Theory*, 2nd ed. Chapman and Hall, London.
- Newton, H. N., Barnhart, C., Vance, P. H., 1998. Constructing railroad blocking plans to minimize handling costs. *Transportation Science*, 32 (4), 330-345.
- Petersen, E. R., 1977a. Railyard modeling: Part I. Prediction of put-through time. *Transportation Science*, 11(1), 37-49.
- Petersen, E. R., 1977b. Railyard modeling: Part II. The effect of yard facilities on congestion. *Transportation Science*, 11(1), 50-59.
- Selvam, K. K. and Borjian, S. 2013. INFORMS RAS: 2013 problem solving competition final report. *INFORMS Annual Meeting 2013, Railway Application Section, Problem Solving Competition*, Minneapolis, U.S.A.
- Turnquist, M. A., Daskin, M. S., 1982. Queueing models of classification and connection delay in railyards. *Transportation Science*, 16(2), 207-230.
- Wang, H., Jin, J., and Lin, M., 2013. Solving the Railway Yard Operation Problem: Greedy Heuristics, Integer Programming Models, and Waiting Time Approximations. *INFORMS*

- Annual Meeting 2013, Railway Application Section, Problem Solving Competition, Minneapolis, U.S.A.
- Wolsey, L., 1998. Integer Programming. John Wiley & Sons, Inc, New York.
- Zhou, W., Deng, L. and Zhou, Z., 2013. Solution Report: Optimizing the Operational Plan in Railway Classification Yard by Combining Genetic Algorithm and Sub-period Rolling. INFORMS Annual Meeting 2013, Railway Application Section, Problem Solving Competition, Minneapolis, U.S.A.
- Zhou, X., Zhong, M., 2007. Single-track train timetabling with guaranteed optimality: branch-and-bound algorithms with enhanced lower bounds. *Transportation Research Part B*, 41(3), 320-341.
- Zhu, Y., Zhu, R., 1983. Sequence reconstruction under some order-type constraints. *Scientia Sinica Series A*, 26(7), 702-713.