

Research Article

A Simulated Annealing Approach for the Train Design Optimization Problem

Federico Alonso-Pecina¹ and David Romero²

¹Universidad Autónoma del Estado de Morelos, 62209 Cuernavaca, MOR, Mexico

²Instituto de Matemáticas, Universidad Nacional Autónoma de México, 62210 Cuernavaca, MOR, Mexico

Correspondence should be addressed to Federico Alonso-Pecina; federicoalonsopecina@hotmail.com

Received 10 March 2017; Revised 20 June 2017; Accepted 9 July 2017; Published 10 August 2017

Academic Editor: Jorge Magalhaes-Mendes

Copyright © 2017 Federico Alonso-Pecina and David Romero. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Train Design Optimization Problem regards making optimal decisions on the number and movement of locomotives and crews through a railway network, so as to satisfy requested pick-up and delivery of car blocks at stations. In a mathematical programming formulation, the objective function to minimize is composed of the costs associated with the movement of locomotives and cars, the loading/unloading operations, the number of locomotives, and the crews' return to their departure stations. The constraints include upper bounds for number of car blocks per locomotive, number of car block swaps, and number of locomotives passing through railroad segments. We propose here a heuristic method to solve this highly combinatorial problem in two steps. The first one finds an initial, feasible solution by means of an ad hoc algorithm. The second step uses the simulated annealing concept to improve the initial solution, followed by a procedure aiming to further reduce the number of needed locomotives. We show that our results are competitive with those found in the literature.

1. Introduction

An ample range of investigations have been undertaken to develop optimization algorithms for various problems encountered in the realm of rail systems, like routing [1], scheduling [2–4], crew assignment [5, 6], and blocking [7, 8], among others [9–12]. However, to our knowledge, the Train Design Optimization Problem as defined below is the only attempt to simultaneously deal with block-to-train assignment, train routing, and crew assignment [13], arising as one of the most fundamental and difficult combinatorial optimization problems formulated in the railroad industry [13–17], with a huge potential to benefit from the application of operations research techniques.

In rail systems, a railroad car, railcar, or train car (*car*, for short), is a vehicle used for the carrying of cargo. Such cars, when coupled together and hauled by one or more locomotives, form a *train*. A car block (*block*, for short) is a semipermanently arranged formation of cars. Trains are then built of one or more blocks coupled together as needed. Also, in the operation of rail systems a *block swap* can occur,

namely, when a locomotive delivers a block at a station distinct from the block's destination, and another locomotive picks up the block afterwards. Each time a train stops en route at a station to pick up and/or to deliver blocks, a *work-event* takes place.

In this paper we address the Train Design Optimization Problem (TDOP) arising in the operation of railroad freight transport as part of the logistics chain. It consists in determining, at minimal total cost, and subject to capacity and operational constraints, the number of locomotives and crews, together with the logistics related to the movement of locomotives, blocks, and crews through a railway network, so as to transport goods from a set of shippers to a set of destinations.

The total cost depends on the number of assigned locomotives, the distance traveled by locomotives and cars, the distance traveled by crews to return to their departure stations, the number of block swaps and work-events, the number of blocks not arriving to destination, and the difference between the number of locomotives arriving to and departing from stations. The constraints include maximum number of

blocks per locomotive, maximum number of block swaps, maximum number of work-events per locomotive, maximum length, weight, and number of trains passing through a railroad segment, and crew limitations as to the route to follow.

Only a handful of approaches have been developed for the solution of the TDOP. In [18] a mixed integer programming model is proposed; block routes are first generated together with train routes to cover them, then a matching is sought to minimize the objective function, and finally several greedy and local search rules are iteratively invoked to update the block-paths and train routes so as to improve the solution quality. In [19, 20] the problem is formulated as one of integer programming where the number of variables and constraints is exponential, proposing for its solution a column-row generation heuristic followed by clever tabu search methods. An iterative procedure is suggested in [21] to solve two subproblems of the TDOP: train design and block-to-train assignment, where the former consists in determining train routes to be operated, and the latter deals with the routing decision for blocks; both subproblems are solved by integer programming techniques. In [22, 23] a column-generation approach is designed: first, a set of promising train routes is generated based on the crew segments; then an integer linear programming model is developed for the subsequent decisions including train route selection and block-to-train assignment.

We propose hereafter a method to heuristically solve the TDOP in two steps. By means of an ad hoc procedure the first step aims to produce an initial feasible solution, namely, a solution satisfying all the constraints. The second step uses the simulated annealing method to improve the initial solution, followed by simple, specialized procedures that attempt to reduce the number of needed trains without increasing the overall cost. To test our proposal we solved the only three published instances to date, finding it competitive with other approaches in the literature. Also, we randomly generated 20 synthetic instances as well as (improvable) lower bounds for their corresponding optimal solutions; our heuristic came up with results averaging an error close to 25% on these bounds.

The rest of the paper is organized as follows: Section 2 provides the necessary terminology and makes a detailed description of constraints and objective function of the TDOP; a toy example is also furnished to help understand the problem. In Section 3 the TDOP is formulated in mathematical programming terms. Our solution approach—basically combining an ad hoc algorithm to find an initial feasible solution followed by the metaheuristic known as simulated annealing—is explained in Section 4. Further, Section 5 is devoted to the computational experiments that we conducted to test our procedures; for this experimentation we used the three available known instances as well as a new set of 20 randomly generated ones. Finally, Section 6 presents our conclusions and a proposal for future work.

2. Problem Description

This section is aimed to describe the TDOP, borrowing some terminology from [20]. An alternative description

can be found in [13]. The TDOP terminology, constraints, and objective function are provided in Section 2.1. To help in grasping our description, an instance of the TDOP is presented in Example 1 by means of a toy example, followed by one of its feasible solutions in Section 2.2.

2.1. Terminology, Constraints, and Objective Function

Terminology

- (i) A *block* is a formation of cars sharing origin and destination. Trains are built of one or more blocks coupled together as needed.
- (ii) A *train* is a locomotive carrying or not carrying blocks.
- (iii) A *block swap* occurs when a block is moved from one train to another.
- (iv) A *block-path* is a sequence of railroad segments through which a block can be feasibly routed from its origin to its destination.
- (v) A *work-event* occurs each time a train stops en route at a station to pick up and/or deliver blocks. The train stop at its destination (or origin) station is not considered as a work-event.
- (vi) A *crew segment* is a minimal length route between two stations, called *end points*, on which crews operate trains in either direction.
- (vii) The *crew imbalance* on a crew segment, defined by say end points x and y , is the absolute difference between the number of crews going from x to y , and the number of crews going from y to x .
- (viii) The *train imbalance* on a station, say x , is the absolute difference between the number of trains originating in x and the number of trains terminating in x .
- (ix) A car is *missed* if it is not transported from its origin to its destination.

Constraints

- (i) *Blocks per train*: trains are constrained by an upper bound on the number of blocks they carry.
- (ii) *Swaps per block*: each block is constrained by an upper bound on the number of times it can be swapped.
- (iii) *End points*: crews must start and end traveling at end points.
- (iv) *Crew-to-train assignment*: every train has to be assigned to a crew on each crew segment, and it must originate and terminate at the end points of a crew segment, even if the train has to move part of the way along a crew segment not carrying any blocks (the entire train routes should be decomposed in crew segments).
- (v) *Upper bounds on segments*: railroad segments are constrained by upper bounds on the number, length, and weight of trains traversing them in either direction.

The objective of the TDOP is to minimize the sum of eight components:

- (1) *Locomotive cost*: product of the number of scheduled locomotives and the unit locomotive cost C_1 .
- (2) *Locomotive travel cost*: product of the total traveled miles by all scheduled locomotives, and the per mile locomotive travel cost C_2 .
- (3) *Work-event cost*: product of the total number of work-events of all scheduled trains, and the unit work-event cost C_3 .
- (4) *Car travel cost*: product of the total traveled miles by all cars, and the per mile car travel cost C_4 .
- (5) *Block swap cost*: let $s(v)$ be the unit block swap cost in station v . For a given block b , denote $\beta(b)$ the set of stations where block b is swapped; then its (total) block swap cost is $\sum_{v \in \beta(b)} s(v)$.
- (6) *Crew imbalance cost*: product of the number of all crew imbalances and the unit crew imbalance cost C_5 .
- (7) *Train imbalance cost*: product of the number of all train imbalances and the unit train imbalance cost C_6 .
- (8) *Missed car cost*: product of the total number of missed cars and the cost per missed car C_7 .

Example 1. Consider a railroad network with five stations A, B, C, D, and E and six railroad segments as schematically depicted in Figure 1, where distances (in miles) are assumed symmetrical and all segments are bidirectional. Seven blocks must be delivered. Table 1 furnishes the relevant data.

The only nonadjacent end points defining a crew segment are B and D. The corresponding crew segment is BCD because its length is minimal among all possible paths connecting B and D. Clearly, the other crew segments are trivially found. Thus, if the route of some train is, say, $D \rightarrow C \rightarrow B \rightarrow A$, then at D a crew from crew segment BCD could be assigned to this train in the subroute $D \rightarrow C \rightarrow B$, and subsequently at B a crew from crew segment BA could be assigned to the train in the subroute $B \rightarrow A$. When a train crosses over from one crew segment to another, the onboard crew gets off the train and a new crew gets onboard. Further, crew segments are bidirectional. Hence, crews in crew segment BA can take a train from A to B or vice versa. It is assumed that crews always travel along the shortest path between any pair of stations.

2.2. A Feasible Solution of Example 1. Consider the solution shown in Table 2. Two trains, t_1 and t_2 , are scheduled to transport blocks. Note that train t_1 picks up 63 cars at station D and then goes to C, where it delivers 63 cars and picks up 5 cars. From station C train t_1 travels to B to pick up 42 cars; then it goes to station A to deliver 47 cars and to pick up 13 cars. Finally, it travels to station B to deliver 13 cars.

The distance traversed by both t_1 and t_2 is 1,273 miles. Train t_1 stops to pick up or drop blocks in stations C, B, and A, while train t_2 stops to pick up or drop blocks in stations D, B, and C. Hence, the total number of work-events produced by both trains is six.

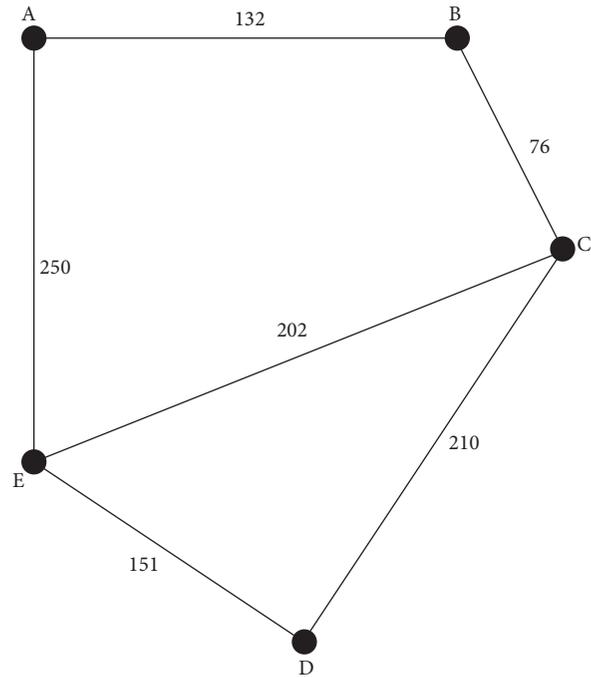


FIGURE 1: Diagram of railroad networks. Example 1.

Table 3 indicates the train on which each block travels between stations. For instance, block b_3 travels first on train t_1 from A to B, then on train t_2 from B to D, yielding one block swap (with cost 60).

The column labeled “Car miles” is computed as the product of the distance traveled by a block on a segment, and the number of cars in the block. Note that the block-to-train assignment satisfies the constraints on the maximum number of block swaps, and on the maximum number of blocks per train (see Tables 1(a) and 1(b)).

In Table 4 the crew-to-train assignment information is provided. It is shown, for instance, that a crew travels from D to B on train t_1 . Then another crew travels from B to A on train t_1 . Finally, this crew travels from A to B on the same train.

Note that there is one train imbalance in station B because it is the origin of no train, and one train terminates there. Similarly, there is one train imbalance in station E. Hence, this solution yields two train imbalances.

The objective function value of this solution is as much as 47,603, computed in Table 5 (costs in \$).

3. The Model

We consider the railroad network as a directed graph $G = (V, E)$ whose set of nodes V corresponds to the set of stations, and the set of arcs E is derived from the railroad segments between stations, associating a couple of arcs with opposite directions to each rail segment. The length of arc $e = (u, v) \in E$ is equal to the distance between stations $u, v \in V$. Also, for $e \in E$, let $\bar{L}(e)$, $\bar{W}(e)$, and $\bar{T}(e)$ denote, respectively, the maximum train length, the maximum weight, and the

TABLE 1: Data of Example 1.

| (a) | |
|---------|-----------------|
| Station | Block swap cost |
| A | 60 |
| B | 60 |
| C | 80 |
| D | 20 |
| E | 20 |

| (b) | | | | | |
|-------|--------|-------------|-----------|---------------|---------------|
| Block | Origin | Destination | # of cars | Length (feet) | Weight (tons) |
| b_1 | C | A | 5 | 290 | 420 |
| b_2 | C | D | 48 | 2,976 | 3,696 |
| b_3 | A | D | 13 | 819 | 962 |
| b_4 | D | B | 4 | 228 | 316 |
| b_5 | E | D | 12 | 708 | 936 |
| b_6 | D | C | 63 | 3,969 | 4,914 |
| b_7 | B | A | 42 | 2,730 | 3,570 |

| (c) | | | | |
|---------|----------------|-----------------------------|-----------------------------|----------------------|
| Segment | Length (miles) | Maximum train length (feet) | Maximum train weight (tons) | Max number of trains |
| BC | 76 | 4,100 | 5,600 | 6 |
| AB | 132 | 4,400 | 6,300 | 12 |
| DE | 151 | 5,700 | 5,400 | 6 |
| CE | 202 | 4,500 | 7,900 | 11 |
| CD | 210 | 4,000 | 10,000 | 9 |
| AE | 250 | 6,200 | 6,500 | 6 |

| (d) |
|--|
| End points defining five crew segments |
| {B, A} |
| {B, D} |
| {A, E} |
| {C, E} |
| {D, E} |

| (e) | |
|---|---------|
| Description | Value |
| Unit locomotive cost (C_1) | \$400 |
| Train travel cost per mile (C_2) | \$10 |
| Cost per work-event (C_3) | \$350 |
| Car travel cost per mile (C_4) | \$0.75 |
| Unit crew imbalance penalty (C_5) | \$600 |
| Unit train imbalance penalty (C_6) | \$1,000 |
| Cost per missed car (C_7) | \$5,000 |
| Maximum number of blocks per train (M_B) | 8 |
| Maximum number of swaps per block (M_S) | 3 |
| Maximum number of work-events per train (M_W) | 4 |

TABLE 2: Train routes solution of Example 1.

| Train | Sequence | Station | Cumulative miles | Deliver cars | Pick up cars | Cars in the train | Crew change |
|---------------|----------|---------|------------------|--------------|--------------|-------------------|-------------|
| t_1 | 1 | D | 0 | 0 | 63 | 63 | NO |
| | 2 | C | 210 | 63 | 5 | 5 | NO |
| | 3 | B | 286 | 0 | 42 | 47 | YES |
| | 4 | A | 418 | 47 | 13 | 13 | NO |
| | 5 | B | 550 | 13 | 0 | 0 | NO |
| t_2 | 1 | E | 0 | 0 | 12 | 12 | NO |
| | 2 | D | 151 | 12 | 4 | 4 | YES |
| | 3 | C | 361 | 0 | 0 | 4 | NO |
| | 4 | B | 437 | 4 | 13 | 13 | NO |
| | 5 | C | 513 | 0 | 48 | 61 | NO |
| | 6 | D | 723 | 61 | 0 | 0 | NO |
| Total mileage | | | 1,273 | | | | |

TABLE 3: Block-to-train assignment of Example 1.

| Block | Sequence | Train | Start station | End station | block swap cost | Segment miles | # of cars | Car miles |
|--------|----------|-------|---------------|-------------|-----------------|---------------|-----------|-----------|
| b_1 | 1 | t_1 | C | A | 0 | 208 | 5 | 1,040 |
| b_2 | 1 | t_2 | C | D | 0 | 210 | 48 | 10,080 |
| b_3 | 1 | t_1 | A | B | 60 | 132 | 13 | 1,716 |
| b_3 | 2 | t_2 | B | D | 0 | 286 | 13 | 3,718 |
| b_4 | 1 | t_2 | D | B | 0 | 286 | 4 | 1,144 |
| b_5 | 1 | t_2 | E | D | 0 | 151 | 12 | 1,812 |
| b_6 | 1 | t_1 | D | C | 0 | 210 | 63 | 13,230 |
| b_7 | 1 | t_1 | B | A | 0 | 132 | 42 | 5,544 |
| Totals | | | | | 60 | 38,284 | | |

TABLE 4: Crew-to-train assignment of Example 1.

| Crew segment | Train | Forward | Backward |
|--------------|-------|---------|----------|
| BD | t_1 | 0 | 1 |
| BA | t_1 | 1 | 1 |
| DE | t_2 | 0 | 1 |
| BD | t_2 | 1 | 1 |

maximum number of trains allowed to traverse arc (or segment) e in either direction.

Denote C the set of crew segments, where each crew segment $c \in C$ is a path through a set of stations $\chi(c)$ —including end points—and let $H := \bigcup_{c \in C} \chi(c)$. We assume crews are always traveling on crew segments.

Let B be the set of blocks to be delivered. For block $b \in B$, its origin, destination, number of cars, length, and weight are denoted $o(b)$, $d(b)$, $r(b)$, $\ell(b)$, and $w(b)$, respectively.

Let Γ be the set of all possible trains, and let $P(\cdot)$ be a generic sequence of arcs of E , with $L_{P(\cdot)}$ denoting its length. Thus, the arc sequence followed by train $t \in \Gamma$ is $P(t)$, and that of block b is $P(b)$; namely, $P(b)$ is a block-path of b . In case train $t \in \Gamma$ is used, let $CR(t)$ and $WE(t)$ be, respectively, the set of crews traveling on train t and the number of work-events of train t . Our mathematical formulation below closely follows the one proposed in [20].

Let $ct(t)$ denote the cost of train $t \in \Gamma$, which consists of the unit locomotive cost C_1 , plus the train travel cost, and plus the cost of work-events. Hence,

$$ct(t) = C_1 + (C_2 \times L_{P(t)}) + (C_3 \times WE(t)). \quad (1)$$

Each train $t \in \Gamma$ can be seen, in fact, as a sequence of crew segments, forming a path $P(t)$. The subset of trains passing through arc $e \in E$ is $T_e \in \Gamma$, and K_t^e is the number of times train t passes through e . From set Γ and graph G , one can obtain a multigraph $MG = (V, A)$, where A is obtained by replacing each arc $e \in E$ with $\sum_{t \in T_e} K_t^e$ copies of it. If a train passes several times through arc e , then each passage induces a different copy.

The binary variable λ_t states whether train $t \in \Gamma$ is used or not in the solution. Let $\Delta_v(t)$ be equal to -1 if train t starts its route in node v , and equal to $+1$ if train t ends its route in v . Thus, $\Delta_v = |\sum_{t \in \Gamma} \Delta_v(t) \lambda_t|$ is the train imbalance in node v .

Let u and v be the end points of a crew segment $c \in C$. Then $\Delta_c(t)$ denotes the difference between the number of times train t goes through c , from u to v , and the number of times it goes through c from v to u . So, $\Delta_c = |\sum_{t \in \Gamma} \Delta_c(t) \lambda_t|$ is the crew imbalance of crew segment c .

Let Ω_b denote the set of all block-paths $P(b)$ associated with block $b \in B$, which includes a dummy block-path of one arc from $o(b)$ to $d(b)$ with cost $C_7 \times r(b)$. Further, let $\Omega_b^t \subseteq \Omega_b$ be the subset of paths in Ω_b that use train t , and let $\Omega = \bigcup_{b \in B} \Omega_b$ be the whole set of block-paths.

TABLE 5: Total costs of Example 1.

| | |
|----------------------|--|
| Locomotive cost | 2 (number of locomotives) \times 400 = 800 |
| Train travel cost | 1,273 \times 10 = 12,730 (see Table 2) |
| Car travel cost | 38,284 \times 0.75 = 28,713 (see Table 3) |
| Work-event cost | 6 (number of train work-events) \times 350 = 2,100 |
| Blocks swap cost | 60 (see Table 3) |
| Crew imbalance cost | 2 \times 600 = 1,200 (see Table 4) |
| Train imbalance cost | 2 \times 1,000 = 2,000 |
| Missed cars cost | 0 |

For $b \in B$, let $cb(b)$ denote the block-path cost for block b , namely, in case of a dummy block-path; then $cb(b) = C_7 \times r(b)$; otherwise

$$cb(b) = (C_4 \times r(b) \times L_{P(b)}) + \sum_{v \in \beta(b)} s(v), \quad (2)$$

whose two terms correspond to block travel cost and block swap cost, respectively. Recall that $r(b)$ is the number of cars of block b , each block swap at node $v \in V$ costs $s(v)$, and $\beta(b)$ is the set of stations where b is block swapped.

Let Ω_e be the set of block-paths that use arc $e \in A$ of multigraph MG . Note that there is one train corresponding to Ω_e . Binary variable $x_{P(b)}$ states whether block-path $P(b) \in \Omega$ is used or not in the solution.

Therefore, our mathematical model of the TDOP is

$$\begin{aligned} \min \quad & z \\ & = \sum_{t \in \Gamma} ct(t) \lambda_t + \sum_{P(b) \in \Omega} cb(b) x_{P(b)} \end{aligned} \quad (3)$$

$$+ C_5 \sum_{c \in C} \Delta_c + C_6 \sum_{v \in V} \Delta_v$$

subject to $\sum_{P(b) \in \Omega_b} x_{P(b)} \geq 1, \quad \forall b \in B$ (4)

$$\sum_{P(b) \in \Omega_b^i} x_{P(b)} \leq \lambda_t, \quad \forall t \in \Gamma \quad (5)$$

$$\sum_{b \in B} \sum_{P(b) \in \Omega_b^i} x_{P(b)} \leq M_B, \quad \forall t \in \Gamma \quad (6)$$

$$\sum_{P(b) \in \Omega_e} \ell(b) x_{P(b)} \leq \bar{L}(e), \quad \forall e \in E \quad (7)$$

$$\sum_{P(b) \in \Omega_e} w(b) x_{P(b)} \leq \bar{W}(e), \quad \forall e \in E \quad (8)$$

$$\sum_{t \in T_e} K_t^e \lambda_t \leq \bar{T}(e), \quad \forall e \in E \quad (9)$$

$$WE(t) \leq M_W, \quad \forall t \in \Gamma \quad (10)$$

$$x_{P(b)}, \lambda_t \in \{0, 1\}, \quad \forall P(b) \in \Omega, t \in \Gamma. \quad (11)$$

The terms in the objective function (3) correspond to the costs of locomotives, blocks, crew imbalances, and

train imbalances, in that order. Constraint (4) imposes the existence of a block-path—real or dummy—for each block $b \in B$. Constraint (5) forces selecting train t if any of the selected block-paths uses it. Constraints (6)–(10) stand for upper bounds on number of blocks that a train can transport, length and weight of trains passing through arcs, number of trains passing through arcs, and number of work-events per train, respectively. We emphasize that every sequence of train $t \in \Gamma$ must be decomposed in a sequence of crew segments.

4. Algorithm

This section presents the algorithm developed by us to tackle the TDOP. It consists of two main steps.

Step 1 (initial feasible solution). Composed by substeps (1.1)–(1.3), this step is aimed to produce from scratch a feasible solution of the TDOP, namely, where constraints (6)–(11) are satisfied.

(1.1) *Crew Selection*. For every block $b \in B$ determine a minimum length path ξ_b from $o(b)$ to $d(b)$, where each arc can be operated by at least one crew. The set of crews $\pi(b)$ to operate along path ξ_b is chosen following substeps (1.1.1)–(1.1.5) below. For every $\pi(b)$ a minimum set of trains is constructed, so as only one train corresponds to every pair of crews such that one ends where the other starts. Note that with this rule the corresponding trains to carry b can be easily deduced. Let $\mathcal{Q}(b) := \{Q \mid Q \text{ be a set of crew segments covering the arcs of } \xi_b\}$.

In substep (1.1.4) every possible effort is made to avoid the overlapping of crews in $\mathcal{Q}(b)$; therefore, even though $\mathcal{Q}(b)$ contained overlapping crews, this is not an issue since the probability of this happening is truly negligible. In the sequel $\mathcal{Q}(b)$ will be simply written \mathcal{Q} . Although \mathcal{Q} is theoretically of exponential size, in all our experiments we were able to enumerate it completely.

(1.1.1) For $Q \in \mathcal{Q}$, let $\gamma(Q)$ be the number of swaps of block b when transported through crew segments Q . Determine $\gamma^* := \min\{\gamma(Q) : Q \in \mathcal{Q}\}$ and reduce \mathcal{Q} by eliminating from it every Q such that $\gamma(Q) > \gamma^*$. If \mathcal{Q} has a unique element, say, \bar{Q} , make $\pi(b) := \bar{Q}$; otherwise go to (1.1.2).

(1.1.2) If in \mathcal{Q} there is a crew set starting in $o(b)$, then reduce \mathcal{Q} by eliminating from it every Q not starting in $o(b)$. If \mathcal{Q} has a unique element, say, \bar{Q} , make $\pi(b) := \bar{Q}$; otherwise go to (1.1.3).

(1.1.3) If in \mathcal{Q} there is a crew set ending in $d(b)$, then reduce \mathcal{Q} by eliminating from it every Q not ending in $d(b)$. If \mathcal{Q} has a unique element, say, \bar{Q} , make $\pi(b) := \bar{Q}$; otherwise go to (1.1.4).

(1.1.4) Let $\rho(Q)$ be the total distance traveled by the trains corresponding to the crew segments in $Q \in \mathcal{Q}$, and let $\rho^* := \min\{\rho(Q) : Q \in \mathcal{Q}\}$. Reduce \mathcal{Q} by eliminating from it every Q such that $\rho(Q) > \rho^*$. If \mathcal{Q} has a unique element, say, \bar{Q} , then make $\pi(b) := \bar{Q}$; otherwise go to (1.1.5). For example, assume we have a block with origin A and destination D, together with crews A-B-C, B-C-D, and A-B. One solution is to use crews A-B-C and B-C-D, while another uses crews B-C-D and A-B; with this criterion the second solution is chosen.

(1.1.5) Let $q^* = \min\{|Q| : Q \in \mathcal{Q}\}$. Reduce \mathcal{Q} by eliminating from it every Q such that $|Q| > q^*$. If \mathcal{Q} has a unique element, say, \bar{Q} , make $\pi(b) := \bar{Q}$; otherwise make $\pi(b)$ equal to an arbitrarily selected $Q \in \mathcal{Q}$.

(1.2) *Block Ordering.* An order \bar{B} on the set of blocks B is constructed as follows. Let $L(\xi_b)$ be the length of path ξ_b . If $L(\xi_{b'}) > L(\xi_{b''})$ then block b' precedes block b'' in \bar{B} . However, if $L(\xi_{b'}) = L(\xi_{b''})$, then b' precedes b'' in \bar{B} whenever $\nu(b') > \nu(b'')$, where $\nu(\cdot)$ denotes the number of blocks with origin $o(\cdot)$ and destination $d(\cdot)$. In case $L(\xi_{b'}) = L(\xi_{b''})$ and $\nu(b') = \nu(b'')$ hold together, block b' precedes b'' whenever $r(b') > r(b'')$. Finally, if $L(\xi_{b'}) = L(\xi_{b''})$, $\nu(b') = \nu(b'')$, and $r(b') = r(b'')$, this tie between b' and b'' is arbitrarily broken to produce \bar{B} .

(1.3) *Train Assignment.* Blocks are considered one after the other, according to order $\bar{B} = (b_1, \dots, b_m)$. First, assign block b_1 to the crews selected in Step (1.1), together with the required train(s). For $j = 2, \dots, m$, if path ξ_{b_j} is contained in path ξ_{b_i} for some $i \in [1, j-1]$, then assign block b_j to path ξ_{b_i} whenever upper bounds on feet, tons, and max blocks per train allow. Otherwise, assign block b_j to the crews selected in Step (1.1), together with the required train(s). This step produces a feasible solution S .

Step 2 (simulated annealing). Proposed more than three decades ago [24, 25] Simulated Annealing (SA, for short) is one of the most successful metaheuristics to find good solutions of many combinatorial optimization problems (see [26] and the references therein), including the railroad freight transportation design problem [27]. In its theoretical formulation, SA converges with probability 1 to a global minimum under certain assumptions on the control parameters. In practice, these assumptions are impossible to be implemented, but adequate cooling schemes increase the likelihood of obtaining a near optimal solution [28].

Central to SA is the neighborhood concept. For our proposal we hand tailored it as follows. Let $\Phi(\mathcal{F})$ denote the set of feasible solutions of an instance \mathcal{F} of the TDOP. The *neighborhood* of a solution $S \in \Phi(\mathcal{F})$ is

$$N(S) = N_1(S) \cup N_2(S) \cup N_3(S), \quad (12)$$

```

 $\tau \leftarrow T_o; S^* \leftarrow S;$ 
While  $\tau \geq T_f$  do
   $k \leftarrow 0;$ 
  While  $k \leq \theta$  do
     $S' \leftarrow \text{NEIGHBOR}(S);$ 
     $\delta \leftarrow z(S') - z(S);$ 
    If  $\text{RAND}(0, 1) < e^{-\delta/\tau}$  or  $(\delta < 0)$  then
       $S \leftarrow S';$ 
    EndIf
     $k \leftarrow k + 1;$ 
    If  $z(S) < z(S^*)$  then
       $S^* \leftarrow S; k \leftarrow 0;$ 
    EndIf
  EndWhile
   $\tau \leftarrow \alpha\tau;$ 
EndWhile
call FUSION.

```

ALGORITHM 1: Algorithm simulated annealing.

where (i) $N_1(S) \subseteq \Phi(\mathcal{F})$ contains every solution obtainable from S by adding one or more trains to deliver one block alone, say b , through ξ_b . (ii) $N_2(S) \subseteq \Phi(\mathcal{F})$ contains the solutions obtainable from S by removing one block, say b , from the set Ψ of trains carrying it and delivering it through a minimal length route from $o(b)$ to $d(b)$ without using any train of Ψ , using another trains in S . (iii) $N_3(S) \subseteq \Phi(\mathcal{F})$ contains every solution obtainable from S by the removal of one block, say b , from the set Ψ of trains carrying it, and delivering it through a minimal length route from $o(b)$ to $d(b)$ without using any train of Ψ , with at least one added train, and using at least one existing train.

We implemented a simulated annealing procedure with geometrical cooling scheme as shown below, where the variable τ stands for the system temperature, and the parameters T_o , T_f , α , θ , denote, respectively, the initial and the final temperature, the cooling factor, and the internal cycle length.

Procedure NEIGHBOR(S) randomly produces (with uniform distribution) a feasible solution in $N(S)$, and RAND delivers a uniformly distributed random number in the interval $(0, 1)$. SA starts with the solution S delivered by Step 1 above (see Algorithm 1).

Throughout the process, the variable S^* holds the best solution so far, and the counter k helps to control the number of iterations of the internal cycle. Thus, the system temperature τ drops if and only if θ iterations occur without any improvement on S^* .

Added at the end of the external cycle, procedure FUSION attempts to improve solution S^* by successively applying the following four operations in the order shown. Before passing to the next operation, for each operation all possibilities are considered, repeating it exhaustively as long as the cost is lowered and feasibility is preserved.

- (1) When two trains have identical routes, one locomotive is eliminated once its blocks are passed to the other.

TABLE 6: Average results—objective function value rounded to nearest integer—of ten runs for instance *Data Set 2*. Time shown in seconds and italics.

| $\theta \setminus \alpha$ | 0.70 | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 | 0.99 |
|---------------------------|--------------|--------------|---------------|---------------|---------------|---------------|----------------|
| 500 | 3,209,687 | 3,210,058 | 3,212,176 | 3,209,065 | 3,205,954 | 3,195,864 | 3,185,775 |
| | <i>2,790</i> | <i>2,226</i> | <i>2,446</i> | <i>2,666</i> | <i>2,783</i> | <i>3,934</i> | <i>10,877</i> |
| 1,000 | 3,205,775 | 3,204,303 | 3,203,858 | 3,200,948 | 3,201,772 | 3,191,151 | 3,184,551 |
| | <i>2,459</i> | <i>2,593</i> | <i>2,771</i> | <i>3,351</i> | <i>4,259</i> | <i>5,181</i> | <i>19,899</i> |
| 1,500 | 3,200,260 | 3,201,520 | 3,192,792 | 3,190,837 | 3,187,281 | 3,190,916 | 3,173,312 |
| | <i>2,605</i> | <i>2,634</i> | <i>2,901</i> | <i>3,450</i> | <i>4,355</i> | <i>7,264</i> | <i>29,082</i> |
| 2,000 | 3,201,157 | 3,196,194 | 3,192,166 | 3,195,499 | 3,188,878 | 3,188,311 | 3,175,760 |
| | <i>3,070</i> | <i>2,890</i> | <i>2,540</i> | <i>4,332</i> | <i>4,997</i> | <i>9,066</i> | <i>36,945</i> |
| 4,000 | 3,195,857 | 3,187,514 | 3,187,182 | 3,185,158 | 3,184,931 | 3,176,745 | 3,173,650 |
| | <i>3,808</i> | <i>4,771</i> | <i>5,339</i> | <i>6,844</i> | <i>8,904</i> | <i>16,043</i> | <i>76,952</i> |
| 6,000 | 3,192,045 | 3,189,039 | 3,185,947 | 3,180,655 | 3,181,356 | 3,175,733 | 3,169,176 |
| | <i>5,857</i> | <i>5,976</i> | <i>7,398</i> | <i>8,711</i> | <i>16,026</i> | <i>25,530</i> | <i>116,304</i> |
| 8,000 | 3,188,106 | 3,185,158 | 3,184,032 | 3,181,325 | 3,179,988 | 3,172,099 | 3,167,935 |
| | <i>6,741</i> | <i>6,844</i> | <i>9,216</i> | <i>12,206</i> | <i>18,319</i> | <i>37,695</i> | <i>153,931</i> |
| 10,000 | 3,186,663 | 3,184,824 | 3,182,085 | 3,178,864 | 3,176,411 | 3,168,984 | 3,164,844 |
| | <i>7,466</i> | <i>8,622</i> | <i>10,846</i> | <i>14,772</i> | <i>21,081</i> | <i>42,862</i> | <i>218,506</i> |

- (2) If there are stations u, v , such that $\sum_{t \in \Gamma} \Delta_u(t) \lambda_t < 0$ and $\sum_{t \in \Gamma} \Delta_v(t) \lambda_t > 0$, then a train with no blocks is added starting in v and ending in u .
- (3) If a train starts its route in a station where another train terminates its travel, one locomotive is eliminated once its blocks are passed to the other.
- (4) If there are trains t', t'' , such that the t'' route is contained in the t' route, then the t'' locomotive is eliminated once its blocks are passed to t' .

Example 2. For clarity sake, consider a railroad network as schematically depicted in Figure 2. Assume a block b with $o(b) = A$ and $d(b) = D$, and let the crew segments be AF, AD, AI, AP, FD, IL, LD, QR, and DP. Also, let S be a feasible solution with six trains in correspondence to routes: $A \rightarrow E \rightarrow F, F \rightarrow G \rightarrow D, A \rightarrow H \rightarrow I, I \rightarrow J \rightarrow F \rightarrow K \rightarrow L, A \rightarrow N \rightarrow O \rightarrow P$, and $L \rightarrow M \rightarrow D$, where block b is delivered by the trains assigned to routes $A \rightarrow E \rightarrow F$ and $F \rightarrow G \rightarrow D$; hence 63 is their total traveled distance. Thus, solution $S' \in N_1(S)$ is identical to S with the exception that block b is delivered by a new train through route $A \rightarrow B \rightarrow C \rightarrow D$, with length 42.

Solution $S'' \in N_2(S)$ is identical to S with the exception that block b is delivered by the trains with routes $A \rightarrow H \rightarrow I, I \rightarrow J \rightarrow F \rightarrow K \rightarrow L$, and $L \rightarrow M \rightarrow D$, with total length 91, and without adding a new train.

Solution $S''' \in N_3(S)$ is identical to S with the exception that block b is delivered through route $A \rightarrow N \rightarrow O \rightarrow P$, adding a new train for segment PD; the total length of this route is 47.

In regard to simulated annealing parameters, we tested diverse settings, modifying one parameter at a time. Among the possible combinations of the cooling factor $\alpha = 0.70, 0.75, 0.80, 0.85, 0.90, 0.95$, and 0.99 , and the internal cycle lengths $\theta = 500, 1,000, 1,500, 2,000, 4,000, 6,000, 8,000$, and $10,000$,

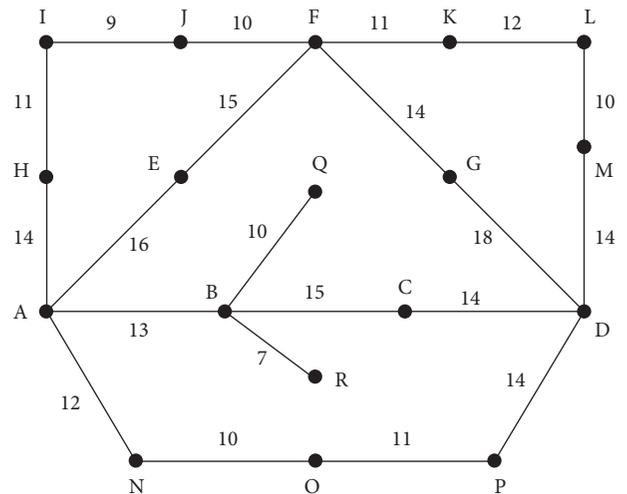


FIGURE 2: Diagram of railroad networks. Example 2.

for every pair $\{\alpha, \theta\}$ we run ten times the algorithm on a PC with Windows 8.

Table 6 displays the average results obtained for instance *Data Set 2*, where computer times are shown in italics. Thus, we chose $\alpha = 0.90$ and $\theta = 1,000$ since together they yielded a good trade-off between computer time and solution quality. As initial temperature we chose $T_o = 30,000$, because on average it yielded an acceptance rate of around 95%. For the final temperature $T_f = 1$ was selected, yielding an acceptance rate of around 1% on average.

5. Numerical Results

Our simulated annealing approach, as described in Section 4 to deal with the TDOP, was implemented on a computer with Xeon E5-2643 v2 3.5 GHz processor (we run in a single

TABLE 7: Performance comparison of our SA approach and other methods; computing times are given in seconds. The bottom row shows a lower bound on the optimal cost, computed as explained in Section 5.3. (all figures rounded to integers).

| Number of | <i>Example</i> | | <i>Data Set 1</i> | | <i>Data Set 2</i> | |
|----------------------|----------------|------|-------------------|--------|-------------------|--------|
| Nodes | 4 | | 94 | | 221 | |
| Blocks | 5 | | 239 | | 369 | |
| Arcs | 5 | | 134 | | 294 | |
| Crew Segments | 5 | | 154 | | 154 | |
| | cost | time | cost | time | cost | time |
| [21] | — | — | 2,043,471 | 45 | 3,187,999 | 225 |
| [20] | — | — | 2,032,516 | 2,201 | 3,188,500 | 2,067 |
| [18] | — | — | 2,018,644 | 80,000 | 3,152,019 | 75,000 |
| [23] | 14,916 | 0 | 2,069,752 | 342 | 3,240,177 | 2,700 |
| SA | 14,916 | 1 | 2,005,514 | 181 | 3,170,815 | 1,991 |
| Lower bound σ | 10,956 | — | 1,660,361 | — | 2,751,213 | — |

processor), 64 GB RAM, and g++ compiler. Two experiments were conducted to investigate its efficiency.

In the first experiment—see Section 5.1—we tested SA on three instances available to us: *Example* (a toy, synthetic instance), *Data Set 1* (real), and *Data Set 2* (real), from [13].

The second experiment was designed to evaluate the performance of SA from the point of view of the quality of the results and the required computer time. We did extensive testing on randomly generated instances of various sizes; they are dealt with in Section 5.2.

5.1. Testing SA on Specific Instances. Employing parameters $\theta = 1000$ and $\alpha = 0.9$, we compared SA results on instances *Example*, *Data Set 1*, and *Data Set 2*, as can be seen in Table 7, where computation times are also shown to give an idea of the performance of the implementation, although the platforms used were not the same. So far [18] had provided—using a mixed integer programming model—the best results for the two real instances but at the expense of very high computing time. On the other hand, the network-oriented formulation in [21] yielded the fastest algorithm to date.

Other approaches include column-generation [23], with better results than those found in [22], and column-generation combined with tabu search [20] improving on [19]. In the case of instance *Example* the only published results come from [23]. The best results found with SA were 1,999,315 with $\theta = 8000$ and $\alpha = 0.99$, in 19,819 seconds for instance *Data Set 1*, and 3,155,267 with $\theta = 10,000$ and $\alpha = 0.95$ in 41,233 seconds for instance *Data Set 2*, with the more relevant results found in the literature; our results yield lower cost for *Data Set 1* (−0.95%), same cost for *Example*, and higher cost for *Data Set 2* (+0.10%).

5.2. Testing SA on Random Instances. A set of 20 random instances was generated as follows with the number n of stations and number m of blocks shown in column II of Table 8. All random choices were made with uniform distribution.

For each pair $\{n, m\}$ we first construct a complete graph $K_n = (V, E)$, whose vertex set V corresponds to a set of n

integer points randomly generated in a square of side 2^{15} ; the length of each edge $(u, v) \in E$ is equal to the Euclidean distance between vertices u and v . Also, we determine the set $E_t \subset E$ of edges in a minimum length spanning tree on K_n , as well as the set $E_h \subset E$ of edges in the convex hull of V . Finally, we make the railroad network of the instance correspond to graph $G = (V, E_h \cup E_t)$.

Then, crew segments are randomly created so that every edge of the network belongs to one crew segment. The swap cost $s(v)$ in every station $v \in V$ is also randomly assigned in the integer range [10, 100].

We create next a set of m blocks, one after the other, verifying that each does not exceed the maximum number of swaps allowed to arrive to destination when traveling along a minimal length path. The number of cars $r(b)$, length $\ell(b)$, and weight $w(b)$ of every block b are randomly chosen in the integer ranges [1, 100], [56, 65], and [74, 86], respectively, as these ranges are similar to those found in real instances. Each time we tentatively form a block, Step 1 of Section 4 is performed for feasibility verification.

Also, for every edge $e \in E$, the values of $\bar{L}(e)$, $\bar{W}(e)$, and $\bar{T}(e)$ are randomly chosen in the ranges [7,000, 14,000] and [9,000, 18,000], [6, 12], respectively.

Once random instances were generated, SA run 50 times for each. Results are displayed in columns III–V of Table 8 (indeed, instance p5_7D was taken as Example 1 in Section 2).

5.3. Lower Bounds. Establishing good lower bounds for the optimal solution of the TDOP seems a very difficult task. However, to assess the SA performance, we propose here a lower bound σ for each solved instance; see Tables 7 and 8. To this aim we computed as explained below lower bounds for total travel cost (σ_1), train start cost (σ_2), train travel cost (σ_3), work-event cost (σ_4), and missed car cost (σ_5). Thus, $\sigma = \sum_{j=1}^5 \sigma_j$.

In regard to travel cost let $\sigma_1 = C_4 \sum_{b \in B} (\psi(b) \times r(b))$, where $\psi(b)$ is the length of the shortest route from $o(b)$ to $d(b)$. In the case of the two real instances σ_1 is identical to the bound computed in [23].

TABLE 8: Relevant data (columns I, II) and results (columns III, IV, and V) of 50 runs of SA on 20 random instances. A lower bound—explained in Section 5.3—on the optimal solution cost is shown in column VI. The last column is an indicator of the solution quality.

| I | II | III | IV | V | VI | VII |
|-----------|---|------------------------|--------------|----------------|--------------------------|--|
| Instance | Number of stations, blocks, arcs, crew segments | Minimum cost (μ) | Average cost | Avg time (sec) | Lower bound (σ) | Difference in % $100 \times \frac{\mu - \sigma}{\sigma}$ |
| p5_7D | 5 7 5 6 | 47,193 | 47,193 | 1 | 32,957 | 43.18 |
| p5_10D | 5 10 4 6 | 77,691 | 77,726 | 1 | 60,336 | 28.76 |
| p10_15D | 10 15 11 14 | 224,186 | 226,845 | 4 | 187,862 | 19.34 |
| p10_20D | 10 20 8 13 | 294,595 | 299,021 | 4 | 248,649 | 18.48 |
| p20_30D | 20 30 13 25 | 297,952 | 301,013 | 13 | 243,932 | 22.15 |
| p20_40D | 20 40 15 25 | 336,651 | 339,876 | 13 | 267,840 | 25.69 |
| p40_60D | 40 60 22 45 | 565,954 | 576,906 | 49 | 446,538 | 26.74 |
| p40_80D | 40 80 21 45 | 568,089 | 573,991 | 46 | 446,411 | 27.26 |
| p80_120D | 80 120 38 86 | 791,072 | 800,660 | 124 | 614,225 | 28.79 |
| p80_160D | 80 160 38 89 | 987,646 | 997,005 | 168 | 791,041 | 24.85 |
| p160_240D | 160 240 62 169 | 2,214,593 | 2,240,989 | 376 | 1,791,395 | 23.62 |
| p160_320D | 160 320 72 169 | 2,201,347 | 2,221,606 | 320 | 1,765,637 | 24.68 |
| p200_300D | 200 300 82 216 | 4,043,845 | 4,071,617 | 409 | 3,306,843 | 22.29 |
| p200_400D | 200 400 81 210 | 1,591,058 | 1,607,980 | 463 | 1,203,586 | 32.19 |
| p250_375D | 250 375 96 261 | 2,518,148 | 2,548,918 | 856 | 1,958,093 | 28.60 |
| p250_500D | 250 500 104 260 | 1,881,881 | 1,905,522 | 723 | 1,445,175 | 30.22 |
| p300_450D | 300 450 117 311 | 8,919,042 | 8,969,957 | 606 | 7,480,671 | 19.23 |
| p300_600D | 300 600 115 313 | 10,419,523 | 10,453,791 | 744 | 8,782,219 | 18.64 |
| p320_480D | 320 480 138 333 | 8,172,082 | 8,218,034 | 662 | 6,676,505 | 22.40 |
| p320_640D | 320 640 138 332 | 2,053,295 | 2,077,627 | 900 | 1,516,815 | 35.37 |

For the train start cost a trivial lower bound is $\sigma_2 = C_1 \times \lceil m/M_B \rceil$, where M_B stands for the maximum number of blocks per train.

Let (ψ_1, \dots, ψ_m) be an order on $\Psi = \{\psi(b) : b \in B\}$, with $\psi_j \geq \psi_{j+1}$, for $j = 1, \dots, m-1$. Thus, for the train travel cost our proposed lower bound is $\sigma_3 = C_2 \sum_{j \in J} \psi_j$, where $J = \{j : (1 = j \bmod M_B) \text{ and } 1 \leq j \leq m\}$. Bound σ_3 slightly improves on that from [20].

For every station $v \in V$, let $\rho(v)$ be the number of blocks b such that $o(b) = v$ or $d(b) = v$. Then $\sigma_4 = C_3 \sum_{v \in X} \lceil \rho(v)/M_B \rceil$, where X is the set of stations belonging to a crew segment with the exclusion of its end points. As far as we know this lower bound for the work-event cost is proposed here for the first time.

Let Y denote the set of blocks b such that $o(b)$ or $d(b)$ does not belong to H (refer to Section 3 for a definition of vertex set H). Thus, it is impossible to deliver any block in Y . Our lower bound for the missed car cost is then $\sigma_5 = C_7 \sum_{b \in Y} r(b)$. Although in practice it is not likely to get $\sigma_5 > 0$, instance *Data Set 02* contains some blocks belonging to Y .

Thus, in regard to the best solutions found for instances *Example*, *Data Set 1*, and *Data Set 2* (see Table 7) our lower bound σ yields differences of 36.14%, 20.79%, and 14.57%, respectively, namely, 23.83% on average. Column VII of Table 8 shows a similar behavior of σ for the synthetic

instances. These facts lead us to believe that σ , although the highest lower bound known to us is far from the true optimum.

6. Conclusion

In this paper we proposed a simulated annealing approach for the Train Design Optimization Problem. This approach was computationally tested with three instances well-known in the specialized literature, and with 20 instances randomly generated by us, of sizes up to 320 stations and 640 car blocks. Its results show superiority—in regard to the objective function value—over those obtained elsewhere for instance *Data Set 01*, and competitiveness (in fact the runner-up) for instance *Data Set 02*, both results using reasonable computing resources.

We think however that much research, be it theoretical or empirical, must still be carried out to successfully deal with this most challenging, combinatorial optimization problem. The TDOP being so fundamental in the railroad industry, there is a need to develop new heuristics or metaheuristics, which, working alone or hybridized, improve the best results obtained so far. Also, a pending and difficult task is to produce efficient procedures that yield good lower bounds for the TDOP objective function, so as to help assess the performance of proposed approaches.

Disclosure

David Romero is on sabbatical leave at Laboratorio Nacional de Informática Avanzada, Xalapa, Veracruz, Mexico.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

References

- [1] A. A. Khaled, M. Jin, D. B. Clarke, and M. A. Hoque, "Train design and routing optimization for evaluating criticality of freight railroad infrastructures," *Transportation Research Part B: Methodological*, vol. 71, pp. 71–84, 2015.
- [2] V. Cacchiani and P. Toth, "Nominal and robust train timetabling problems," *European Journal of Operational Research*, vol. 219, no. 3, pp. 727–737, 2012.
- [3] M. F. Gorman, "An application of genetic and tabu searches to the freight railroad operating plan problem," *Annals of Operations Research*, vol. 78, pp. 51–69, 1998.
- [4] X. Zhou and M. Zhong, "Single-track train timetabling with guaranteed optimality: branch-and-bound algorithms with enhanced lower bounds," *Transportation Research Part B: Methodological*, vol. 41, no. 3, pp. 320–341, 2007.
- [5] A. Balakrishnan, A. Kuo, and X. Si, "Real-Time decision support for crew assignment in double-ended districts for U.S. freight railways," *Transportation Science*, vol. 50, no. 4, pp. 1337–1359, 2016.
- [6] B. Vaidyanathan and R. K. Ahuja, "Crew Scheduling Problem," in *Handbook of Operations Research Applications at Railroads*, pp. 163–175, Springer Verlag, 2015.
- [7] R. K. Ahuja, K. C. Jha, and J. Liu, "Solving real-life railroad blocking problems," *Interfaces*, vol. 37, no. 5, pp. 404–419, 2007.
- [8] C. Van Dyke and M. Meketon, "Railway blocking process," in *Handbook of Operations Research Applications at Railroads*, B. W. Patty, Ed., pp. 119–162, Springer Verlag, 2015.
- [9] A. A. Assad, "Models for rail transportation," *Transportation Research A: General*, vol. 14, no. 3, pp. 205–220, 1980.
- [10] J. Cordeau, P. Toth, and D. Vigo, "A survey of optimization models for train routing and scheduling," *Transportation Science*, vol. 32, no. 4, pp. 380–404, 1998.
- [11] S. Harrod and M. F. Gorman, "Operations research for freight train routing and scheduling," in *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley and Sons, 2010.
- [12] A. K. Nemani and R. K. Ahuja, "OR models in freight railroad industry," in *Wiley Encyclopedia of Operations Research and Management Science*, John Wiley and Sons, Wiley Encyclopedia of Operations Research and Management Science, 2010.
- [13] INFORMS, "RAS Problem Solving Competition," 2011, <https://www.informs.org/Community/RAS/Problem-Solving-Competition/2011-RAS-Problem-Solving-Competition>.
- [14] M. F. Gorman, "Operations research in rail pricing and revenue management," *International Series in Operations Research and Management Science*, vol. 222, pp. 243–254, 2015.
- [15] C. Van Dyke and M. Meketon, "Train Scheduling," in *Handbook of Operations Research Applications at Railroads*, B. W. Patty, Ed., pp. 1–42, Springer Verlag, 2015.
- [16] C. Van Dyke and M. Meketon, "Car scheduling/trip planning," in *Handbook of Operations Research Applications at Railroads*, B. W. Patty, Ed., pp. 79–118, Springer Verlag, 2015.
- [17] N. Azad, E. Hassini, and M. Verma, "Disruption risk management in railroad networks: an optimization-based methodology and a case study," *Transportation Research Part B: Methodological*, vol. 85, pp. 70–88, 2016.
- [18] I.-L. Wang, H.-Y. Lee, and Y.-T. Liang, "Train design optimization," in *INFORMS 2011 Annual Meeting - RAS 2011, Problem Solving Competition*, Charlotte, NC, USA, 2011.
- [19] F. Colombo, R. Cordone, and M. Trubian, "A column-row generation heuristic for the train design optimization problem," in *Proceedings of the INFORMS 2011 Annual Meeting - RAS 2011 Problem Solving Competition*, vol. 10, Charlotte, NC, USA, 2011, https://www.informs.org/content/download/254681/2405878/file/AllReportsFrom-RAS-2011_Problem_Competition.pdf.
- [20] F. Colombo, *Mathematical Programming Algorithms for Network Optimization Problems. Dissertation, [Dissertation, thesis]*, Università degli Studi di Milano, Italy, 2014.
- [21] L. Lozano, J. E. González, and A. L. Medaglia, "A network-oriented formulation and decomposition approach to solve the 2011 RAS problem," in *Proceedings of the INFORMS 2011 Annual Meeting - RAS 2011 Problem Solving Competition*, vol. 10, Charlotte, NC, USA, https://www.informs.org/content/download/254681/2405878/file/AllReportsFrom-RAS-2011_Problem_Competition.
- [22] J. G. Jin, J. Zhao, and D. H. Lee, "INFORMS RAS Problem Solving Competition 2011," in *Proceedings of the INFORMS 2011 Annual Meeting*, vol. 10, Charlotte, NC, USA, https://www.informs.org/content/download/254681/2405878/file/AllReportsFrom-RAS-2011_Problem_Competition.pdf.
- [23] J. G. Jin, J. Zhao, and D. H. Lee, "A column generation based approach for the train network design optimization problem," *Transportation Research Part E*, vol. 50, p. 17, 2013.
- [24] V. Cerný, "Minimization of Continuous Functions by Simulated Annealing," Research Institute for Theoretical Physics HU-TFT-84–51, University of Helsinki, 1984.
- [25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [26] E. Aarts, J. Korst, and W. Michelis, "Simulated annealing," in *Search Methodologies*, E.K. Burke and G. Kendall, Eds., 285, p. 265, Springer Science and Business Media, New York, NY, USA, 2014.
- [27] Á. Marín and J. Salmerón, "A simulated annealing approach to the railroad freight transportation design problem," *International Transactions in Operational Research*, vol. 3, no. 2, pp. 139–149, 1996.
- [28] P. J. Van Laarhoven and E. H. Aarts, "Simulated annealing," in *Simulated Annealing: Theory and Applications*, pp. 7–15, Springer, Netherlands, 1987.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

