Quantum Information Processing (2022) 21: 337
https://doi.org/10.1007/s11128-022-03670-y

Quadratic and higher-order unconstrained binary optimization of railway rescheduling for quantum computing

Krzysztof Domino[1] · Akash Kundu[1,3] · Özlem Salehi[1] · Krzysztof Krawiec[2]

Received: 25 March 2022 / Accepted: 31 August 2022 / Published online: 29 September 2022
© The Author(s) 2022

Abstract

As consequences of disruptions in railway traffic affect passenger experience and satisfaction, appropriate rerouting and/or rescheduling is necessary. These problems are known to be NP-hard, given the numerous restrictions of traffic nature. With the recent advances in quantum technologies, quantum annealing has become an alternative method to solve such optimization problems. To use quantum annealing, the problem needs to be encoded in quadratic unconstrained binary optimization (QUBO) or higher-order binary optimization (HOBO) formulation that can be recast as a QUBO. This paper introduces QUBO and HOBO representations for rescheduling problems of railway traffic management; the latter is a new approach up to our knowledge. This new approach takes into account not only the single-track lines but also the double- and multi-track lines, as well as stations composed of tracks and switches. We consider the conditions of minimal headway between trains, minimal stay on stations, track

Krzysztof Domino, Akash Kundu, Özlem Salehi and Krzysztof Krawiec have contributed equally to this work.

✉ Krzysztof Domino
kdomino@iitis.pl

Akash Kundu
akundu@iitis.pl

Özlem Salehi
osalehi@iitis.pl

Krzysztof Krawiec
krzysztof.krawiec@polsl.pl

1 Institute of Theoretical and Applied Informatics Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland

2 Faculty of Transport and Aviation Engineering, Silesian University of Technology, Akademicka 2A, 44-100 Gliwice, Poland

3 Joint Doctoral School, Silesian University of Technology, Akademicka 2A, 44-100 Gliwice, Poland

occupation, and rolling stock circulation. Furthermore, a hybrid quantum-classical procedure is presented that includes rerouting. We demonstrate the proof of concept implementation on the D-Wave Quantum Processing Unit and D-Wave hybrid solver.

Keywords Higher-order binary optimization · Railway rescheduling · Railway rerouting · Quantum-classical hybrid procedure · D-Wave annealer

1 Introduction

Railway transport is perceived as a more sustainable and ecological alternative to individual mobility [1, 2]. The increasing train traffic and other safety-related issues cause dispatching problems in case of disturbances which may lead to rerouting and rescheduling. Failure to resolve them quickly and efficiently can cause inconvenience for the passengers and increase the costs. No matter what the reason for the disturbance is (technical malfunction of traffic control, collision with car or animal, system—see [3]), the objective is to reduce delay propagation [4–6].

One can harness quantum computing for solving the railway rescheduling and rerouting problem formulated as an optimization problem. A promising heuristic algorithm is quantum annealing, which relies on the quantum adiabatic model of computation [7–9]. Commercially available quantum annealers are provided by the D-Wave company [10]. The problem of interest needs to be formulated as an Ising model, which in turn determines the coupling strength between the pair of qubits of the machine. After the system is evolved slowly enough for a particular duration, it is expected to be found in the minimum energy state, encoding the solution that minimizes the objective function of the problem. Any problem that is formulated as a quadratic unconstrained binary optimization (QUBO) can be easily transformed into an Ising model and solved using quantum annealing in principle. Since it is more natural to express problems using QUBO representation than the Ising model, it is desirable to find QUBO formulations for optimization problems to target quantum annealing [11, 12]. A generalization of QUBO is higher-order binary optimization (HOBO) representation that allows not only quadratic terms but also higher-order terms in the objective function. There has been some recent work on formulating HOBO representations for combinatorial optimization problems in the context of quantum optimization [13–15].

The motivation of this paper is to demonstrate that it is possible to encode typical railway infrastructure and traffic conditions as QUBOs and HOBOs, making the problems quantum computing ready. This paper is a follow-up of [16] which comprises railway rescheduling under disruptions on a single-track railway line encoded using QUBO. Now, we remove the restriction of single-track lines, enabling also double- and multi-track lines on model trains traffic on stations. We use a parallel machine approach improved by rerouting, resulting in a hybrid algorithm. The presented representations for railway rescheduling and rerouting include the conditions of the minimal headway between trains, minimal stay on stations, station/track occupation, and rolling stock circulation. We use a classical procedure that mimics real-life rerouting practices together with quantum annealing to solve the rescheduling problem and end up with a hybrid algorithm. Although the detailed discussion of our approach concerns

the railway rescheduling problem introduced here, similar approaches can be adopted for problems from other branches of operational research such as factory trolleys or electric busses rescheduling/rerouting.

There is a vast amount of research in the scope of resuming the railway system's capacity and proper functioning after a disruption; for a systematic review see [17]. There are also publications in which other techniques like genetic algorithms and deep learning are used. One may find out more in numerous review papers [18–20] in the scope of optimization methods to solve railway conflict management problems. Given the NP-hardness of such rescheduling problems and their complexity, it is very challenging to solve them on current computational devices in a reasonable time. We expect quantum computing to offer novel opportunities to overcome these limitations.

In our research, we chose the parallel machine approach, where trains have a fixed route within the stations [21]. The reason is that passenger trains have fixed platforms within the station, and the platform change is an extraordinary situation that affects passengers. For demonstration reasons, we start with an Integer Linear Programming formulation where we use order variables [22] to determine the order of trains leaving the station. Alternatively, for the QUBO and HOBO approaches, we use discrete-time units [23], in which binary variables describe whether the event happens at a given time.

Our paper follows other research efforts toward solving transportation-related problems using quantum annealers [15, 24, 25] or quantum approximate optimization algorithm [26] (QAOA) [13]. HOBOs are considered in some mentioned papers for various transportation problems. However, up to our knowledge, HOBO formulation is considered to address the railway rescheduling problem for the first time.

The paper is organized as follows. Section 2 gives a brief overview of the railway system model, which consists of infrastructure and traffic. In this section, we present the notions and formalism to describe the problem of railways rescheduling. In Sect. 3, we present a linear programming representation, we set out the QUBO and HOBO formulations, and we describe our approach to rerouting. We demonstrate the formulations in Sect. 4 both theoretically and using numerical calculations. The last section contains conclusions and a discussion on the possibility of further development of QUBO and HOBO representations to address railway rescheduling.

2 Railway system model

Trains run according to a *schedule* along the routes. The route of the train is composed of stations and lines between them. The line consists of one or more parallel tracks, each split into *line blocks*. The latter we understand as a track section between two signaling utilities that can only be occupied by one train at a time. Stations consist of tracks interconnected by railroad switches (referred to as *switches*). Similar to *line blocks*, stations consist of *station blocks*—track sections at stations between two signaling utilities that can be occupied by only one train at a time. Trains are controlled by dispatchers who can reroute or/and reschedule them if necessary. By rerouting, we understand the change of the track used by a train within a line or a station. By rescheduling, we understand the modification of the train departure time in a

way to avoid conflict and maintain the feasibility of the timetable. (Note that we define infrastructure terms from train traffic perspective rather than their physical characteristics, which is not the usual description in transportation research—we keep this description to keep it coherent with our mathematical model aiming to make it more illustrative.)

Two trains meet and pass (M–P) meeting at the same spatial location while following the same route in opposite directions. Similarly, two trains meet and overtake (M–O) when one train overtakes another. Depending on the type of railway line, M–Ps and M–Os may occur at stations and/or on lines. We distinguish single-track, double-track, and multi-track lines. On *single-track* lines, trains can M–P and M–O only at stations. The usual use of *double-track* line is such that trains are heading in one direction on one track and in the other direction on the other track (unidirectional traffic). It implies M–P possibility at stations and lines and M–O possibility only at stations. We also consider another use of double-track lines as two parallel single-track lines (bidirectional traffic). In this mode, trains can M–O on the line between stations while heading in the same direction on both tracks (this is at the cost of M–P possibility). The bidirectional mode may also be used on *multi-track lines*.

Regardless of the type of line, trains need to keep minimal headway—the distance between two trains following the same direction to preserve safety conditions. Such headway can be measured either in space or in time if taking into account trains' speeds. Trains can terminate at a station, set off from a station, have scheduled stop there, or pass it by. As *conflict* [27], we understand the situation that occurs when at least two trains compete for the same resource in terms of blocks (station or line) or switches. In our model, we aim to resolve optimally all conflicts by rescheduling and rerouting while keeping the safety conditions and limiting the schedule modification.

For model's simplicity, let us assume that the schedule is a pre-set sequences of blocks with departure times assigned. We will refer to this as the *default settings*; any change will be considered as rerouting and creation the new model. In real rail traffic, the schedule is assumed to be conflict free, and conflicts appear due to *delays*. We define *delays* as the difference between $t(j, s_{\text{in}})$ or $t(j, s_{\text{out}})$—the actual time of entering or leaving particular station s by train j, and the scheduled time $\sigma(j, s_{\text{in}})$ or $\sigma(j, s_{\text{out}})$. In the rest of this section, we use s^* for either s_{in} or s_{out}. Hence, the delay is:

$$d(j, s^*) = t(j, s^*) - \sigma(j, s^*). \tag{1}$$

Following [27], we split the delay into *unavoidable* $d_{\text{u}}(j, s^*)$ and *additional* $d_{\text{a}}(j, s^*)$ in the following manner:

$$d(j, s^*) = d_{\text{u}}(j, s^*) + d_{\text{a}}(j, s^*). \tag{2}$$

By unavoidable delay, we understand the delay from outside the model that is propagated through the network, not including any delay that may be caused by other trains' traffic and that cannot be controlled in the model. (Unavoidable delay may be caused by accidents, technical failure, delay from outside the analyzed network, or delays of the trains affected by those on subsequent stations.) The additional delay comprises

delays beyond unavoidable caused by solving conflicts due to traffic, which is in control of our model. The latter is of main interest to us as our goal is to minimize the additional delays.

As we intend not to extend the delays, we assume that the additional delays are limited by the parameter $d_{\max}$:

$$0 \leq d_{\mathrm{a}}(j, s^*) \leq d_{\max}. \tag{3}$$

$d_{\max}$ is a parameter of the model and limits the range of the integer variables in the linear model and the number of variables in QUBO or HOBO approaches; as such, it affects the problem size. It should not be set too low, resulting in a situation in which obtaining a feasible solution is not possible. There are a few possibilities for determining $d_{\max}$. Following [16] and Table 1 therein, one can use some simple heuristics such as FCFS (first come first serve) or FLFS (first leave first serve) to get the solution that is not optimal but feasible. (As discussed in [27], these heuristics are often used in real live railway rescheduling.) Such simple heuristics and solutions can be used to determine $d_{\max}$ for the practical problem. (Bear in mind that in an advanced model, $d_{\max}$ may also be train and station dependent.)

A summary of definition of railway terminologies is given in Table 6 in "Appendix A." The comprehensive illustration of railway infrastructure is given in Fig. 1.

3 Problem formulation

In this section, we discuss the conditions that need to be satisfied and the objective function of the problem. The symbols used are summarized in Table 1. Following [16], our goal is to minimize the weighted additional delay

$$f_o = \sum_{j \in \mathcal{J}} w_j \sum_{s \in \mathcal{S}_j} d_{\mathrm{a}}(j, s_{\mathrm{out}}), \tag{4}$$

where $\mathcal{J}$ is the set of trains, $\mathcal{S}_j$ the set of stations passed by j, w_j is the particular weight reflecting the priority of the j'th train. For implementation reasons, it is more convenient to use:

$$f = \sum_{j \in \mathcal{J}} w_j \sum_{s \in \mathcal{S}_j} \frac{d_{\mathrm{a}}(j, s_{\mathrm{out}})}{d_{\max}}. \tag{5}$$

For clarity of presentation, we introduce the minimal time train j is ready to leave s provided the initial conditions and that no other trains are on the route and denote it by $\upsilon(j, s_{\mathrm{out}})$. By definition,

$$\upsilon(j, s_{\mathrm{out}}) = \sigma(j, s_{\mathrm{out}}) + d_{\mathrm{u}}(j, s_{\mathrm{out}}) \tag{6}$$

Table 1 Summary of the notations used in the paper to denote parameters of the model

Symbol	Description
$j \in \mathcal{J}, (j, j')$	Train, pair of trains
$\mathcal{J}^d, \mathcal{J}^o_{\text{single}}$	Set of trains heading in the same direction on the same route, in opposite directions on the same track
$\mathcal{J}_s^{\text{track}}, \mathcal{J}_s^{\text{switch}}$	Set of trains that compete for the same station block, switch at station
$\mathcal{J}_s^{\text{round}}$	The set of all pairs of trains such that j terminates at s turn around and starts from s as j'
$s \in \mathcal{S}, (s, s')$	Station, pair of stations
$\mathcal{S}_j$	Set of all stations in the route of train j
$\mathcal{C}_j, \mathcal{C}_{j,j'}$	Set of all subsequent pairs of stations in the route of j, common route of j, j'
$\sigma(j, s_{\text{out}})$	Scheduled time of entering, leaving station s by train j
$\upsilon(j, s_{\text{out}})$	Minimal time the train j is ready to leave s, provided the initial conditions and that no other trains are on the route
$d(j, s_{\text{out}}), d_{\text{u}}(j, s_{\text{out}}), d_{\text{a}}(j, s_{\text{out}})$	Delay, unavoidable delay, additional delay of train j on leaving station s
$d_{\max}$	Maximum possible (acceptable) additional delay
$N(d_{\max})$	Number of trains each train may be in conflict at each station, track or switch (on average)
$\tau^{(\text{pass})}(j, s, s')$	Minimal passing time of train j between s and s' (the time it takes train j to travel from s to s')
$\tau^{(\text{blocks})}(j, s, s')$	Minimal time required for train j (traveling from s to s') to release blocks to allow another train to follow at a top speed
$\tau^{(\text{stop})}(j, s)$	Minimal stopping time at the station s by train j
$\tau^{(\text{prep.})}(j, j', s)$	Minimal rolling stock preparation time
$\tau^{(\text{res.})}(j, s)$	Time of using the conflicted resource (i.e., set of swishes) by trains j at stations s
w_j	Weight of train j in the objective
$p_{\text{sum}}, p_{\text{pair}}, p_{\text{qubic}}$	Penalty constants for HOBO/QUBO formulation
f	Objective function
M	A large constant for linearization

and

$$\begin{aligned} d_{\text{a}}(j, s_{\text{out}}) &= d(j, s_{\text{out}}) - d_{\text{u}}(j, s_{\text{out}}) \\ &= t(j, s_{\text{out}}) - \sigma(j, s_{\text{out}}) - d_{\text{u}}(j, s_{\text{out}}) \\ &= t(j, s_{\text{out}}) - \upsilon(j, s_{\text{out}}), \end{aligned} \tag{7}$$

where the first line follows by Eq. (2), the second line follows by Eq. (1) and the third line follows by Eq. (6).

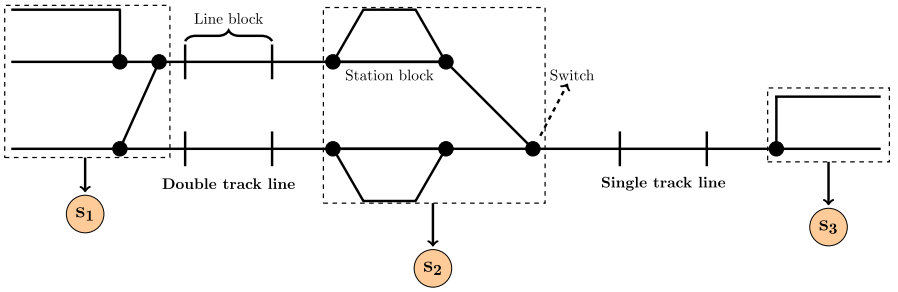


Fig. 1 A comprehensive yet simplified illustration of railway infrastructure

Table 2 List of the variables used in the paper

Symbol	Type	Description
$t(j, s_{\text{out}})$	Integer	Time of train j on leaving station s
$t(j, s_{\text{in}})$	Integer	Time of train j on entering station s, uniquely determined by $t(j, s_{\text{out}})$
$y(j, j, s_{\text{out}})$	Binary 0-1	1 Iff j leaves s before j' (determines the order of trains j and j' while leaving station s)
$z(j, j', s, s')$	Binary 0-1	1 Iff train j enters the single track line between s and s' before j'. (determines the order of trains j and j' while entering the particular track line between station s and s')
$x_{j,t,s}$	Binary 0-1	1 Iff train j leaves station s at time t.
$\tilde{x}_{j,j',t,t',s}$	Binary 0-1	Auxiliary variable for HOBO quadratization $\tilde{x}_{j,j',t,t',s} = x_{j,t,s}x_{j',t',s}$

Now, we can rewrite the objective function defined in Eq. (5) using Eq. (7) as

$$f = \sum_{j \in \mathcal{J}} w_j \sum_{s \in \mathcal{S}_j} \frac{t(j, s_{\text{out}}) - \upsilon(j, s_{\text{out}})}{d_{\max}}. \tag{8}$$

As the objective is defined, we move on to constraints derived from train traffic safety conditions and other technical issues.

We start with the minimal passing time condition which ensures that for any pair of subsequent stations (s, s'), that is on the route of $j \in J$, the entry time to station s' is exactly equal to the leaving time of station s plus the time it takes for train j to move from s to s', which we denote by $\tau^{(\text{pass})}(j, s, s')$, see also Fig. 2. Note that we make an assumption that the train can leave s only if it can proceed at full speed to s'. Given this, the condition can be stated as:

$$t(j, s'_{\text{in}}) = t(j, s_{\text{out}}) + \tau^{(\text{pass})}(j, s, s'). \tag{9}$$

Next, we move to the minimal headway condition. Consider trains j, j' heading in the same direction. To determine their order, we use the precedence variables

$y(j, j', s_{\text{out}}) \in \{0, 1\}$ that is equal to 1 iff j leaves s before j'. (The precedence variable implementation appears to be more efficient than the order variable implementation [28].) Naturally, for any $j, j' \in \mathcal{J}$ and $s \in \mathcal{S}_j \bigcap \mathcal{S}_{j'}$, it follows that

$$y(j, j', s_{\text{out}}) = 1 - y(j', j, s_{\text{out}}). \tag{10}$$

Assume that train j leaves s before train j'. Then, j' needs to wait for at least additional $\tau^{(\text{blocks})}(j, s, s')$ which is the minimal time (headway) required for train j (traveling from s to s') to release blocks to allow j' to follow at full speed, see also simple illustrative presentation in Fig. 2. However, if j is slower than j', then an additional waiting time of $\tau^{(\text{pass})}(j, s, s') - \tau^{(\text{pass})}(j', s, s')$ is needed. For all $j, j' \in \mathcal{J}^d$—the set of pairs of trains heading toward the same direction on the same route—and $(s, s') \in \mathcal{C}_{j,j'}$—the set of subsequent stations in the common route of j and j'—the condition can be expressed as follows:

$$\begin{aligned} &y(j, j', s_{\text{out}}) = 1 \implies \\ &t(j', s_{\text{out}}) \geq t(j, s_{\text{out}}) + \tau^{(\text{blocks})}(j, s, s') + \max\{0, \tau^{(\text{pass})}(j, s, s') - \tau^{(\text{pass})}(j', s, s')\}. \end{aligned} \tag{11}$$

In a single track line, a train can enter the single line only if it is cleared by the train approaching from the opposite direction—we call it the single track line condition. Similar to y, we define the precedence variable $z(j, j', s, s') \in \{0, 1\}$, that determines which train enters first the single track line between s and s'. Note that the following is true for all $j, j' \in \mathcal{J}^o_{\text{single}}$—the set of all trains heading in opposite direction on the same track—and $(s, s') \in \mathcal{C}_{j,j'}$.

$$z(j, j', s, s') + z(j', j, s', s) = 1. \tag{12}$$

By $\tau^{(\text{res.})}(j, s)$, we denote the time of using the conflicted resource (i.e., set of switches) by trains j at station s, see also Fig. 2. For all $j, j' \in \mathcal{J}^o_{\text{single}}$ and $(s, s') \in \mathcal{C}_{j,j'}$, the single track line condition is expressed as:

$$z(j, j', s, s') = 1 \implies t(j', s'_{\text{out}}) \geq t(j, s'_{\text{in}}) + \tau^{(\text{res.})}(j, s'), \tag{13}$$

If the train is due to stop at the station, then it needs to wait at least $\tau^{(\text{stop})}(j, s)$, which is the minimal stopping time at the station s by train j, see Fig. 2. Apart from this, the train must not leave before its scheduled departure time. This is called the minimal stay condition. This results in the following conditions for all $j \in \mathcal{J}$ and $s \in S_j$:

$$t(j, s_{\text{out}}) \geq t(j, s_{\text{in}}) + \tau^{(\text{stop})}(j, s), \tag{14}$$

and

$$t(j, s_{\text{out}}) \geq \sigma(j, s_{\text{out}}). \tag{15}$$

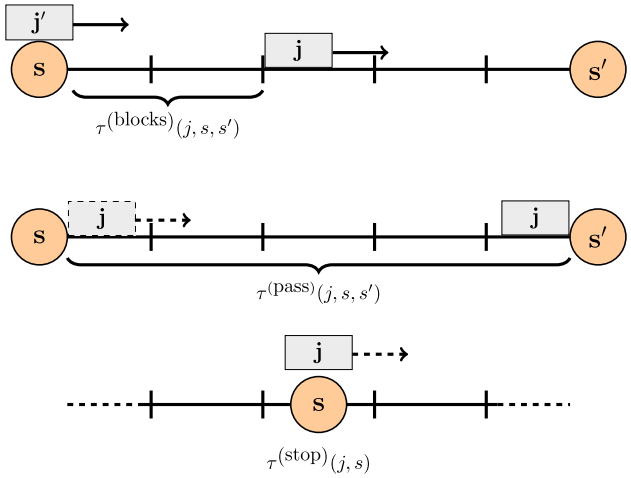


Fig. 2 Illustration of $\tau^{(\text{blocks})}$, $\tau^{(\text{pass})}$ and $\tau^{(\text{stop})}$, in our model they are in time units. In this demonstrative example, $\tau^{(\text{blocks})}$ requires passing two subsequent block sections, which is rather usual for trains traffic management, but not the limitation of the model. (We do not consider here the length of the train)

We also use the rolling stock circulation condition analogous to the one discussed in [16]. By $\tau^{(\text{prep.})}(j, j', s)$, we denote the minimal rolling stock preparation time, if train j terminates at s and then starts as new j', see Fig. 3. For all $s \in S$ and $j, j' \in \mathcal{J}_s^{\text{round}}$—the set of pairs of trains that terminates at s and set off as a new train—we have the condition:

$$t(j', s_{\text{out}}) \geq t(j, s_{\text{in}}) + \tau^{(\text{prep.})}(j, j', s). \tag{16}$$

There are cases where two trains are to use the same set of switches at station s while entering the station, and leaving it. This is called the switch occupancy condition. This condition is (partially) integrated with the single track line condition (a common set of switches where a single line enters/ leaves a station) and track occupancy condition (a common track that can be occupied by one train only). Hence, as $\mathcal{J}_s^{\text{switch}}$ we consider the set of pairs of trains that compete for the same switch or switch set not considered in other conditions. For all $s \in S$ and $j, j' \in \mathcal{J}_s^{\text{switch}}$, this condition can be stated as:

$$y(j, j', s^{***}) = 1 \implies t(j', s^{*}) \geq t(j, s^{**}) + \tau^{(\text{res.})}(j, s), \tag{17}$$

where s^*, s^{**}, s^{***} may be s_{in} or s_{out} depending on the particular situation on the station. Two trains cannot occupy **set of switches** at the station—Eq. (17) In Eq. (17), s^*, s^{**} may be s_{in} or s_{out} depending on the particular trains at a station, similarly $y(j', j, s^{***})$. For example, if j and j' compete for the common switch as j and j' both leave s, we have $s^* = s^{**} = s^{***} = s_{\text{out}}$. There may be also other possibilities, e.g., including z variable instead of y variable; however, we do not discuss them in this simple model.

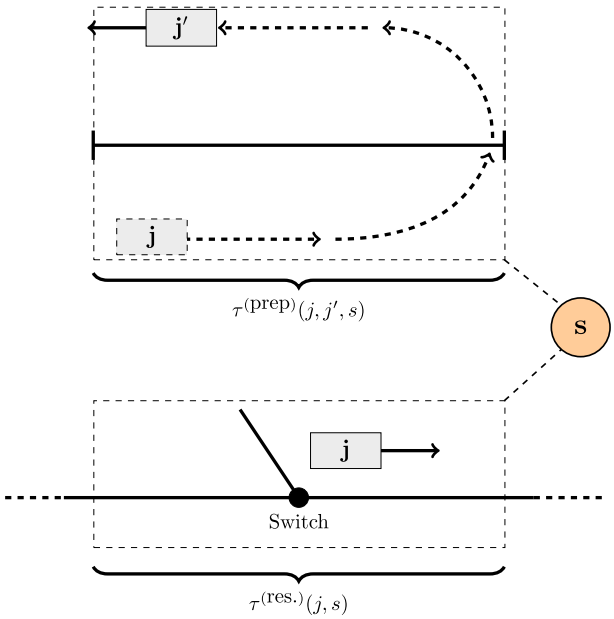


Fig. 3 Illustration of $\tau^{(\text{res})}$ and $\tau^{(\text{prep})}$. Train j terminates at station s and the rolling stock is changed to another train j' (upper panel). Train j occupies switch at station s, and such switch is not available for other train at that time (lower panel)

Now, let us discuss the **track occupancy** condition. As we are using a parallel machine approach, trains are assigned to particular tracks and station blocks that can be occupied only by one train at once. Consider two trains j_1, j_2 that compete for the same track at the station. The subsequent train has to wait until the previous one leaves. This results in

$$y(j, j', s_{\text{out}}) = 1 \implies t(j', s_{in}) \geq t(j, s_{out}) + \tau^{(\text{res.})}(j, s) \tag{18}$$

for all $s \in S$ and $j, j' \in \mathcal{J}_s^{\text{track}}$. Here, $\mathcal{J}_s^{\text{track}}$ is the set of trains that compete for the same track at station s. The additional term $\tau^{(\text{res.})}$ can be used if the two above-mentioned trains use the same set of switches (then, the pair is excluded from $\mathcal{J}_s^{\text{switch}}$).

3.1 Integer linear programming representation

Based on the problem formulation presented above, we construct an integer linear programming (ILP) formulation. To linearize the implications, of the form $a = 1 \implies b \geq c$, we use the transformation $b + M(1 - a) \geq c$ where M is a large constant. Furthermore, we use Eqs. (10) and (12) for the simplification of the equations with

precedence variables. We use the variables $t(j, s_{\text{out}})$, $y(j, j', s_{\text{out}})$ and $z(j, j', s, s')$ as defined previously. ILP takes the following form.

$$\text{min.} \sum_{j \in \mathcal{J}} w_j \sum_{s \in S_j} \frac{t(j, s_{\text{out}}) - \upsilon(j, s_{\text{out}})}{d_{\max}} \tag{19}$$

subject to

$$t(j, s'_{\text{in}}) - t(j, s_{\text{out}}) = \tau^{(\text{pass})}(j, s, s') \qquad \forall_{j \in \mathcal{J}} \forall_{(s,s') \in \mathcal{C}_j} \tag{20}$$

$$t(j', s_{\text{out}}) + M \cdot (1 - y(j, j', s_{\text{out}})) - t(j, s_{\text{out}}) \geq \tau^{(\text{blocks})}(j, s, s') + \max\{0, \tau^{(\text{pass})}(j, s, s') - \tau^{(\text{pass})}(j', s, s')\} \qquad \forall_{j,j' \in \mathcal{J}^d} \forall_{(s,s') \in \mathcal{C}_{j,j'}} \tag{21}$$

$$t(j', s'_{\text{out}}) + M \cdot z(j, j', s, s') - t(j, s'_{\text{in}}) \geq \tau^{(\text{res.})}(j, s') \qquad \forall_{j,j' \in \mathcal{J}^o_{\text{single}}} \forall_{(s,s') \in \mathcal{C}_{j,j'}} \tag{22}$$

$$t(j, s_{\text{out}}) - t(j, s_{\text{in}}) \geq \tau^{(\text{stop})}(j, s) \qquad \forall_{j \in \mathcal{J}} \forall_{s \in S_j} \tag{23}$$

$$t(j, s_{\text{out}}) \geq \sigma(j, s_{\text{out}}) \qquad \forall_{j \in \mathcal{J}} \forall_{s \in S_j} \tag{24}$$

$$t(j', s_{\text{out}}) - t(j, s_{\text{in}}) \geq \tau^{(\text{prep.})}(j, j', s) \qquad \forall_{s \in S} \forall_{(j,j') \in \mathcal{J}^{\text{round}}_s} \tag{25}$$

$$t(j', s_{\text{in}}) + M \cdot y(j, j', s_{\text{out}}) - t(j, s_{\text{out}}) \geq \tau^{(\text{res.})}(j, s) \qquad \forall_{s \in S} \forall_{j,j' \in \mathcal{J}^{\text{track}}_s} \tag{26}$$

$$t(j', s^*) + M \cdot y(j, j', s^{***}) - t(j, s^{**}) \geq \tau^{(\text{res.})}(j, s) \qquad \forall_{s \in S} \forall_{j,j' \in \mathcal{J}^{\text{switch}}_s} \tag{27}$$

$$y(j', j, s_{\text{out}}) \in \{0, 1\},\ z(j, j', s, s') \in \{0, 1\} \qquad \forall_{(s,s') \in \mathcal{C}_{j,j'}} \tag{28}$$

$$t(j, s_{\text{out}}) \in \{\upsilon(j, s_{\text{out}}), \ldots, \upsilon(j, s_{\text{out}}) + d_{\max}\} \qquad \forall_{j \in \mathcal{J}} \forall_{s \in S_j} \tag{29}$$

The range for the integer variables $t(j, s_{\text{out}})$ follows since the following is true by Eqs. (1) and (3), and the definition of $\upsilon(j, s_{\text{out}})$.

$$t(j, s_{\text{out}}) = d_{\text{u}}(j, s_{\text{out}}) + d_{\text{a}}(j, s_{\text{out}}) + \sigma(j, m) = \upsilon(j, s_{\text{out}}) + d_{\text{a}}(j, s_{\text{out}}) \tag{30}$$

$$\upsilon(j, s_{\text{out}}) \leq \upsilon(j, s_{\text{out}}) + d_{\text{a}}(j, s_{\text{out}}) \leq \upsilon(j, s_{\text{out}}) + d_{\max} \tag{31}$$

Although we use the variables $t(j, s_{\text{in}})$ for the clarity of the formulation, thanks to the first constraint, they are defined uniquely and not used when formulating the program. Given this, we have roughly a single time variable per station and train (but some trains may not serve all stations) and overall :

$$\#(t) \leq |\mathcal{J}||\mathcal{S}|. \tag{32}$$

Similarly, we define the precedence variables only for an ordered pair (j, j') as the corresponding variable can be replaced using Eqs. (10) and (12). This results in a single precedence variable y per station and train pair.

$$\#(y) \leq \frac{|\mathcal{J}|\,(|\mathcal{J}| - 1)}{2} |\mathcal{S}|. \tag{33}$$

However, we do not need to compare all pairs of trains in case of dense train traffic, and the number of trains to be compared is somehow limited by $d_{\max}$. (There will be pairs that would never meet for given $d_{\max}$). Let assume each train can be in conflict with $N(d_{\max}) \leq \frac{|\mathcal{J}|-1}{2}$ trains at each station, track or switch (on average). $N(d_{\max})$ is

non-decreasing in $d_{\max}$. We have then the approximation:

$$\#(y) \approx |\mathcal{J}||\mathcal{S}|N(d_{\max}) \tag{34}$$

We also have some additional precedence variables, e.g., for the single-track lines. Using similar approximation:

$$\#(z) \approx |\mathcal{J}||\mathcal{S}|N(d_{\max}) \tag{35}$$

This is, however, adequate if all trains use the single track line; otherwise, we can treat it as the limit.

The number of minimal headway Eq. (21), and track occupancy Eq. (26) constraints are both roughly equal to number of y variables, as each such variable concerns the conflict on these conditions. The number of single track line Eq. (22) conditions is roughly proportional to number of z variables from the same reason. The number of minimal stay constraints Eqs. (23) and (24) are both limited (or can be approximated) by $|\mathcal{S}|\ |\mathcal{J}|$ (limit comes from the fact that not all trains serve all stations).

The number of rolling stock circulation constraints Eq. (25) is not large in comparison with others, for sure it is limited by $\frac{|\mathcal{J}|}{2}$ (this would be a situation that one-half of the trains turn to another half). The number of variables in switch conditions, Eq. (27) is not straightforward, as there are many possibilities and approaches. We can again approximate them by the number of y variables. The number of constraints can be approximated/ limited by:

$$\begin{aligned} \#(\text{constraints}) &\approx 3\#(y) + \#(z) + 2|\mathcal{S}||\mathcal{J}| \\ &\approx (4N(d_{\max}) + 2)|\mathcal{S}||\mathcal{J}| \geq \left(4\frac{|\mathcal{J}| - 1}{2} + 2\right)|\mathcal{S}||\mathcal{J}|. \end{aligned} \tag{36}$$

Hence, one can conclude that if $d_{\max}$ is set properly, the problem size should be linear in the number of trains and stations.

It is broadly accepted that railway problems are equivalent to job-shop models with blocking constraints, see, e.g., [29] (such job-shop is equivalent in principle to the set partition problem, see, e.g., [30]). In detail, in such an NP-hard problem, we have the release t_i and due dates υ_i of jobs, requirements of the model (blocking constraints), and there may be also some additional constraints such as no-waiting, and recirculation (rcrc). In our analogy, trains are jobs, and selected block sections are machines. With the standard notation of scheduling theory [31], our problem falls into the class $\mathcal{J}|t_i, \upsilon_i$,block, no wait, rcrc$|\sum_j w_j T_j$. Above-mentioned conditions comply with ours in the following way:

1. Equation (19) is the objective, weighted tardiness with incorporated due time υ,
2. Equation (20) is the no-waiting constraint on the line (on the station waiting is allowed),
3. Equations (21), (22), (26) and (27) are blocking constraints,
4. Equation (25) is the recirculation constraint;
5. Equations (23) and (24) concern the release time.

The presented linear programming approach is a standalone model. However, it fails in rapid computation for some models with more than a few trains [28]. Hence, it may be beneficial to use another computation paradigm, such as quantum (or quantum-inspired) annealing. As the alternative, in the next subsection we derive the HOBO representation directly form dispatching conditions (i.e., independently on ILP).

3.2 HOBO representation

A higher-order unconstrained binary optimization (HOBO) problem involves the minimization of a multilinear polynomial expression defined over binary variables

$$h(\mathbf{x}) = \sum_{S \subseteq V} c_S \prod_{i \in S} x_i,$$

where $\mathbf{x}$ denotes the vector of all binary variables $x_1, x_2, \ldots, x_n$, $V = \{1, 2, \ldots, n\}$ and c_S are the real coefficients. It is also equivalently expressed as Pseudo-Boolean optimization [32] and polynomial unconstrained binary optimization [33].

The degree or order of a HOBO is the size of the largest set S. The problem is called quadratic unconstrained binary optimization (QUBO) when the degree is equal to 2, and the term HOBO is often used for higher-order problems. For the parallel machine approach adopted in this paper, we have the third order of HOBO.

To formulate the problem, we use the time indexing variable

$$x_{j,t,s} \in \{0, 1\}, \tag{37}$$

that is 1 if train j leaves station s at time t, and 0 otherwise (recall that each time index t can be represented uniquely by delay via Eq. (1)). Note that, we use Eq. (9) to compute the arrival time from the departure time from the previous station. We use the discretized t that is limited from both sides by Eqs. (30) and (31). We denote this limit by

$$t \in \mathcal{T}_{j,s}, \text{ where } \mathcal{T}_{j,s} \equiv \{\upsilon(j, s_{\text{out}}), \upsilon(j, s_{\text{out}}) + 1, \ldots, \upsilon(j, s_{\text{out}}) + d_{\max}\}, \tag{38}$$

here, we consider 1-min resolution. This limitation ensures the timetable condition in Eq. (15).

We have the linear objective function defined as in Eq. (8):

$$f(\mathbf{x}) = \sum_{j \in \mathcal{J}} w_j \sum_{s \in S_j} \sum_{t \in \mathcal{T}_{j,s}} \frac{t(j, s_{\text{out}}) - \upsilon(j, s_{\text{out}})}{d_{\max}} x_{j,t,s}. \tag{39}$$

In our approach, we do not take into account recirculation, i.e., each train leaves each station $s \in S_j$ once and only once:

$$\forall_{j \in \mathcal{J}} \forall_{s \in S_j} \sum_{t} x_{j,t,s} = 1. \tag{40}$$

To convert the constrained problem into an unconstrained one, we use the well-established penalty method [34]. Constraints are incorporated into the objective function so that violation of the constraints adds a positive penalty to the objective function. For instance, to include the constraint in Eq. (40) in the objective function, we set a large enough penalty constant p_{sum} and use the following penalty term:

$$P_{\text{sum}}(\mathbf{x}) = p_{\text{sum}} \sum_{j\in\mathcal{J},s\in S_j} \left(\sum_{t,t'\in\mathcal{T}_{j,s}^{\times 2}\ t\neq t'} x_{j,t,s}x_{j,t',s} - \sum_{t\in\mathcal{T}_{j,s}} x_{j,t,s} \right). \tag{41}$$

Following [16], the conditions described in Eqs. (9)–(18) can be expressed using binary variables so that the quadratic terms yield 0 if the solution is feasible, and produces a penalty otherwise. For this reason, we use a sufficiently large penalty constant p_{pair}. Note that we have symmetric terms $(x_1x_2+x_2x_1)$ to follow the convention of symmetric QUBO formulation.

The minimal headway condition given by Eqs. (10) and (11) can be expressed in the following form:

$$\begin{aligned} &P_{\text{pair}}^{\text{headway}}(\mathbf{x}) \\ &= p_{\text{pair}} \sum_{j,j'\in\mathcal{J}^d} \sum_{(s,s')\in\mathcal{C}_{j,j'}} \sum_{t\in\mathcal{T}_{j,s},t'\in\mathcal{T}_{j',s},A<t'-t<B} (x_{j,t,s}x_{j',t',s} + x_{j',t',s}x_{j,t,s}), \end{aligned}$$

where

$$\begin{aligned} A &= -\tau^{(\text{blocks})}(j',s,s') - \max\{0, \tau^{(\text{pass})}(j',s,s') - \tau^{(\text{pass})}(j,s,s')\}, \\ B &= \tau^{(\text{blocks})}(j,s,s') + \max\{0, \tau^{(\text{pass})}(j,s,s') - \tau^{(\text{pass})}(j',s,s')\}. \end{aligned} \tag{42}$$

The single track condition defined in Eqs. (12) and (13) yields:

$$\begin{aligned} &P_{\text{pair}}^{\text{1track}}(\mathbf{x}) \\ &= p_{\text{pair}} \sum_{j,j'\in\mathcal{J}^o_{\text{single}}} \sum_{(s,s')\in\mathcal{C}_{j,j'}} \sum_{\substack{t\in\mathcal{T}_{j,s},t'\in\mathcal{T}_{j',s'} \\ A<t'-t<B}} (x_{j,t,s}x_{j',t',s'} + x_{j',t',s'}x_{j,t,s}), \end{aligned}$$

where

$$\begin{aligned} A &= -\tau^{(\text{res})}(j,j',s') - \tau^{(\text{pass})}(j',s',s'), \\ B &= \tau^{(\text{pass})}(j,s,s') + \tau^{(\text{res})}(j,j',s'). \end{aligned} \tag{43}$$

The minimal stay condition given in Eq. (14) (incorporated if necessary with Eq. (15)) yields:

$$P_{\text{pair}}^{\text{stay}}(\mathbf{x}) = p_{\text{pair}} \sum_{j\in\mathcal{J}} \sum_{(s,s')\in\mathcal{C}_j} \sum_{\substack{t\in\mathcal{T}_{j,s},t'\in\mathcal{T}_{j,s'} \\ t'<t+\tau^{(\text{pass})}(j,s,s')+\tau^{(\text{stop})}(j,s)}} (x_{j,t,s}x_{j,t',s'} + x_{j,t',s'}x_{j,t,s}). \tag{44}$$

The rolling stock circulation condition in Eq. (16) yields:

$$P_{\text{pair}}^{\text{circ}}(\mathbf{x}) = p_{\text{pair}} \sum_{s\in S} \sum_{(j,j')\in\mathcal{J}_s^{\text{round}}} \sum_{\substack{t\in\mathcal{T}_{j,s},t'\in\mathcal{T}_{j',s'} \\ t'<t+\tau^{(\text{pass})}(j,s,s')+\tau^{(\text{prep.})}(j,j',s)}} (x_{j,t,s}x_{j,t',s'} + x_{j,t',s'}x_{j,t,s}). \tag{45}$$

The switch occupation condition in Eq. (17) yields:

$$P_{\text{pair}}^{\text{switch}}(\mathbf{x}) = p_{\text{pair}} \sum_{s\in S} \sum_{j,j'\in\mathcal{J}_s^{\text{switch}}} \sum_{\substack{t\in\mathcal{T}_{j,s},t'\in\mathcal{T}_{j',s} \\ -\tau^{(\text{res.})}(j',s)<t'-t<\tau^{(\text{res.})}(j,s)}} (x_{j,t,s}x_{j',t',s} + x_{j',t',s}x_{j,t,s}). \tag{46}$$

The above can be checked alone or integrated with other conditions such as track occupation condition in Eq. (18) and single track condition in Eq. (43). The order of trains can be changed at the station only if these trains use different tracks at the station. Suppose that j and j' are on the same track at the station, hence, they cannot change order. To express this condition, we need a higher order term, which yields a HOBO formulation. Let $t'' = t(j', s'_{\text{out}})$, $t' = t(j', s_{\text{out}})$ and $t = t(j, s_{\text{out}})$, where s' is a station prior to s in the route of j'. If j leaves before j', i.e., $t < t'$ ($t \neq t'$ to prevent trains leaving the same track at the same time), then j' must enter after j leaves, i.e., $t'' + \tau^{(\text{pass})}(j', s', s) \geq t + \tau^{(\text{res})}(j, j', s)$. The following term needs to be 0:

$$P_{\text{qubic}}^{\text{occ.}}(\mathbf{x}) = 2p_{\text{pair}} \sum_{s\in S} \sum_{j,j'\in\mathcal{J}_s^{\text{track}}} \sum_{\substack{t\in\mathcal{T}_{j,s},\ t'\in\mathcal{T}_{j',s}\ t''\in\mathcal{T}_{j',s'} \\ t''+\tau^{(\text{pass})}(j',s',s)-\tau^{(\text{res})}(j,j',s)<t\leq t'}} x_{j,t,s}x_{j',t',s}x_{j',t'',s'}. \tag{47}$$

We use the penalty value $2p_{\text{pair}}$ to be consistent with the symmetrization.

The resulting HOBO representation is expressed as:

$$\begin{aligned}\text{min. } h(\mathbf{x}) = f(\mathbf{x}) + P_{\text{sum}}(\mathbf{x}) + P_{\text{pair}}^{\text{headway}}(\mathbf{x}) + P_{\text{pair}}^{\text{1track}}(\mathbf{x}) + P_{\text{pair}}^{\text{stay}}(\mathbf{x}) + P_{\text{pair}}^{\text{circ}}(\mathbf{x}) \\ + P_{\text{pair}}^{\text{switch}}(\mathbf{x}) + P_{\text{qubic}}^{\text{occ.}}(\mathbf{x}),\end{aligned} \tag{48}$$

where $f(\mathbf{x})$ is the objective function and the rest are the penalty terms that need to be minimized.

The penalty constants p_{sum} and p_{pair} have to be large enough to ensure the constraints to be always fulfilled, regardless the penalty value in the objective. However, these constants cannot be too high; because, in that case, they may affect the performance of the quantum annealer.

The number of variables $x_{j,t,s}$ depends on the time resolution of the system and $d_{\max}$. It can be approximated by:

$$\#(x) \leq |\mathcal{J}||\mathcal{S}|(d_{\max}+1). \tag{49}$$

Here "$\leq$" sign is used as some trains may not serve some stations. The number of variables here (as in ILP case Eq. (32)) should be linear in the number of trains and stations, but it is also proportional to $(d_{\max}+1)$. This can be, however, contrasted with the fact that in the ILP case, we have a broader range of time-indexed variables. As argued before, we assume that each train can be potentially in conflict with $N(d_{\max})$ at each station, line or switch. Then, for each condition there are roughly $|\mathcal{J}||\mathcal{S}|(d_{\max}+1)N(d_{\max})$ nonzero QUBO or HOBO terms. The exception is the rolling stock occupation condition that occurs rather rarely in comparison with others. We expect roughly $3|\mathcal{J}||\mathcal{S}|(d_{\max}+1)N(d_{\max})$ nonzero QUBO terms, and $|\mathcal{J}||\mathcal{S}|(d_{\max}+1)N(d_{\max})$ HOBO terms. In comparison with LIP (36), this still can be an efficient implantation provided $d_{\max}$ is controlled.

3.3 QUBO representation

A quadratic unconstrained binary optimization (QUBO) problem is formally defined as

$$q(\mathbf{x}) = \sum_{i,j=1}^{n} x_i Q_{ij} x_j,$$

where Q is a real matrix of coefficients. To be able to solve a problem using quantum annealing, we must first encode it using QUBO formulation as current quantum annealers allow only two-body interactions and representation through Ising model.

In this section, we will convert the HOBO representation into a QUBO representation. Note that we formulate HOBO directly from the dispatching conditions. The advantage of such a take is that in HOBO, we have one-to-one relation between real dispatching constraints and penalties of the mathematical formulation of the problem. (Latter auxiliary variables are only used in quadratization of HOBO). Alternatively, to obtain a QUBO formulation for the problem, one can transform the ILP presented in Sect. 3.1 by first converting inequalities into equalities using slack variables and then moving equality constraints to the objective using the penalty method. The ILP formulation requires binary variables quadratic in the number of trains. Furthermore, for the transformation, additional slack variables are needed as many as the number of inequality constraints which is quadratic in the number of trains, and they need to be optimized within the model as well. Since our HOBO approach is linear in the number of trains, we think that using it as the basis of the QUBO formulation may be more

adequate for dense railroad traffic, with rather small delays; for instance for metro, trams, and urban rapid transport.

The qubic terms in the HOBO representation need to be converted to obtain a QUBO representation. The cubic terms can be expressed using quadratic terms at the cost of introducing new binary variables, see [35]. For the decomposition, we use the auxiliary variable $\tilde{x}_{j,j',t,t',s} = x_{j,t,s}x_{j',t',s}$. The simplest approach here is to use the Rosenberg polynomial approach [36]. The constraint:

$$x_{i_1}x_{i_2}x_{i_3} = 0, \tag{50}$$

is equivalent to:

$$\tilde{x}_k x_{i_3} = 0, \text{ where } \tilde{x}_k = x_{i_1}x_{i_2}, \tag{51}$$

i.e., $k = k(i_1, i_2)$. Then, one can use the polynomial:

$$h(x_{i_1}, x_{i_2}, \tilde{x}_k) = 3\tilde{x}_k^2 + x_{i_1}x_{i_2} - 2x_{i_1}\tilde{x}_k - 2x_{i_2}\tilde{x}_k, \tag{52}$$

that is 0 if $\tilde{x}_k = x_{i_1}x_{i_2}$, and positive (equal to 1 or 3) otherwise. Using the auxiliary vector of variables $\tilde{\mathbf{x}}$, the penalty terms will be as follows:

$$\begin{aligned} &P_{\text{qubic}}(\mathbf{x}, \tilde{\mathbf{x}}) \\ &= p_{\text{pair}} \sum_{(i_1,i_2,i_3)\in\Gamma} (\tilde{x}_{k(i_1,i_2)}x_{i_3} + x_{i_3}\tilde{x}_{k(i_1,i_2)}) + p_{\text{qubic}} \sum_{(i_1,i_2)\in\Gamma'} h(x_{i_1}, x_{i_2}, \tilde{x}_{k(i_1,i_2)}), \end{aligned} \tag{53}$$

where Γ is a set of particular indices of the cubic term (in Eq. (47)), and Γ' a set of indices, where we require Eq. (51) to hold. Observe that for each pair of trains and for each station where the track occupation condition is to be checked; we have roughly $(d_{\max}+1)^2$ auxiliary variables. Hence, this condition needs to be used with caution while modeling railway systems of considerable size.

The resulting QUBO representation is expressed as:

$$\begin{aligned} \text{min. } q(\mathbf{x}, \tilde{\mathbf{x}}) = f(\mathbf{x}) + P_{\text{sum}}(\mathbf{x}) + P_{\text{pair}}^{\text{headway}}(\mathbf{x}) + P_{\text{pair}}^{\text{1track}}(\mathbf{x}) + P_{\text{pair}}^{\text{stay}}(\mathbf{x}) \\ + P_{\text{pair}}^{\text{circ}}(\mathbf{x}) + P_{\text{pair}}^{\text{switch}}(\mathbf{x}) + P_{\text{qubic}}(\mathbf{x}, \tilde{\mathbf{x}}), \end{aligned} \tag{54}$$

where $f(\mathbf{x})$ is the objective function and the rest are the penalty terms that need to be minimized.

The number of $\tilde{x}$ variables $x_{j,t,s}$ depends on $d_{\max}$. It can be approximated by:

$$\#(\tilde{x}) \approx |\mathcal{J}||\mathcal{S}|N(d_{\max})(d_{\max}+1)^2, \tag{55}$$

as we use the same approximation as in (34). We also have in mind that some trains may not serve some stations. When compared with Eq. (33), we can conclude that for

the QUBO approach we need to control $d_{\max}$ more strictly. QUBO implementation may still be efficient but for small $d_{\max}$ determined, e.g., by some simple heuristics.

3.4 Rerouting formulation

We aim to solve the problem of setting the order of already delayed trains having limited resources in terms of infrastructure and traffic regulations. We follow the general idea set out in [37] where the widespread optimization problem needs to be decomposed into smaller components to demonstrate the supremacy of quantum or quantum-inspired approaches. In our case, we propose a decomposition that mimics some real-life rerouting practices. Namely, trains follow their default route as long as it does not cause distortion. Here, we have the subproblem to be optimized by classical, quantum, or hybrid quantum-classical resources. If a solution is not satisfactory, we can change the path of the selected trains (aka reroute them) using the classical approach and then solve the new subproblem. We propose the following algorithm summarized in Fig. 4. The red region indicates the part that can be performed using the quantum (or quantum-inspired) resource at the current state of the art. As quantum computing becomes more and more advanced in the future, we will be moving the quantum border wider and wider to cover the whole algorithm finally.

We start from the given infrastructure, schedule, maximal possible additional delay parameters, priorities of the individual trains, and the default train routes (aka default setting). Then, we perform the optimization and check both feasibility of the solution as well as the objective value. If the solution is infeasible, we pick the nonfeasible conflict. Similarly, if we find the objective value too high, we pick the conflict, increasing the objective value the most. From this conflict, we pick one train (the one with lower priority) and reroute it by:

1. Changing the track to the parallel one,
2. Changing the platform at the station,
3. Changing the path within the station.

We repeat the procedure until we get a satisfactory objective value or we achieve some stopping condition. The optimization subproblem (red) can be encoded either as a linear program, or following the QUBO or HOBO approaches.

4 Demonstration of the model

We consider a railway model which we depict it in Fig. 5. There are 2 stations s_1 and s_2, a double-track line between them, and a depot; the switched are represented with c_i. We have 3 trains:

- Inter-City (the faster one): $j_1, s_1 \to s_2$,
- Regional: $j_2, s_1 \to s_2$,
- Regional: $j_3, s_2 \to s_1$.

We will use the following parameters:

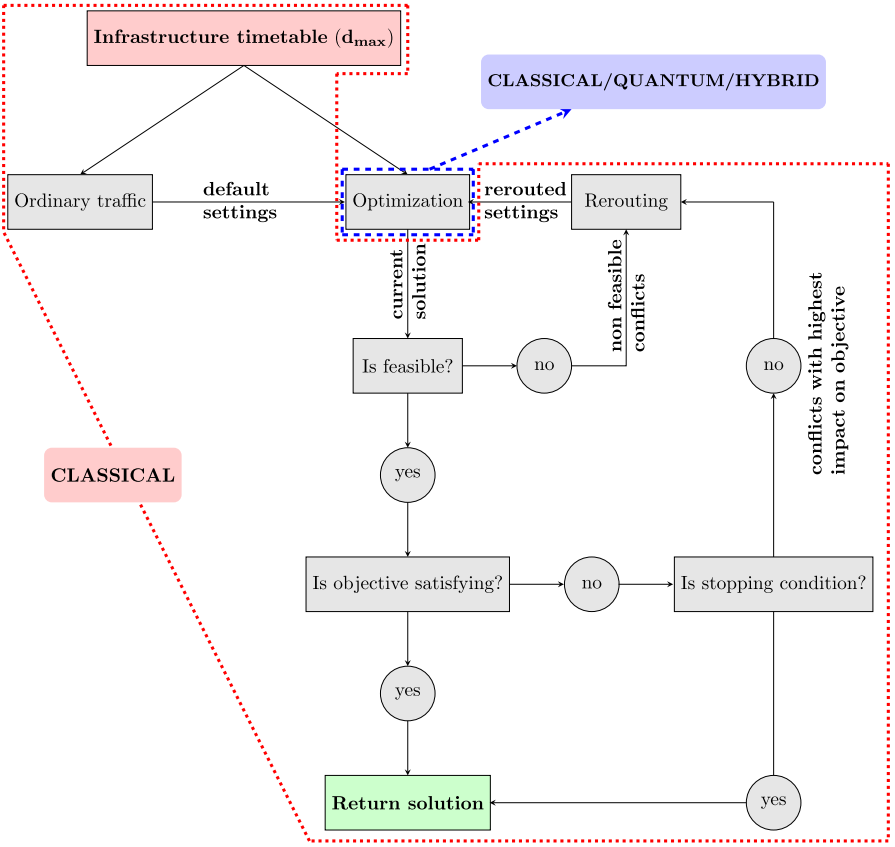


Fig. 4 The algorithm of quantum, classical, and hybrid quantum-classical for railway rescheduling and rerouting

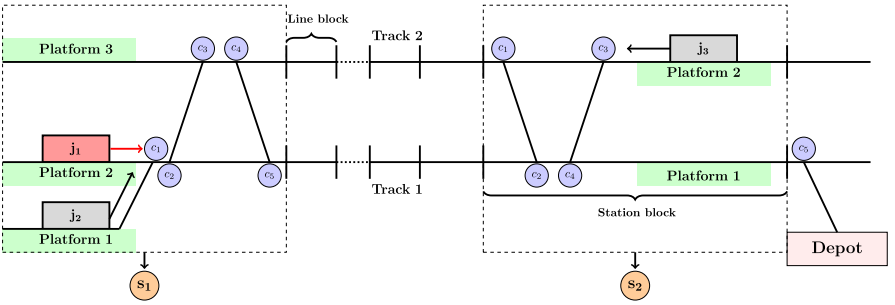


Fig. 5 The demonstrative model

1. Minimal passing time: $\tau^{(\text{pass})}(j_1, s_1, s_2) = 4$, $\tau^{(\text{pass})}(j_2, s_1, s_2) = 8$, and $\tau^{(\text{pass})}(j_3, s_2, s_1) = 8$.
2. Minimal headway: $\tau^{(\text{blocks})}(j_1, s_1, s_2) = 2$ and $\tau^{(\text{blocks})}(j_2, s_1, s_2) = 2$.
3. Minimal stay: $\tau^{(\text{stop})}(j_1, s_2) = \tau^{(\text{stop})}(j_2, s_2) = 1$.
4. For all common resources, $\tau^{(res.)} = 1$.
5. After entering s_2, both j_1 and j_2 depart to the depot after the minimal stay. We only count delays of j_1 and j_2 at s_1 and delay of j_3 at s_2.

Assume all trains are already delayed. Hence, they can leave the stations as soon as the resources (free rail track ahead) are available. We consider the objective as denoted in Eq. (8), with the following weights $w_{j_1} = 2.0$, $w_{j_2} = w_{j_3} = 1.0$ (Inter-City train has higher priority). We set $d_{\max} = 10$ for all trains, and use 1 min resolution. The initial conditions are as follows: $\nu(j_1, s_{1\text{out}}) = 4$, $\nu(j_2, s_{1\text{out}}) = 1$, and $\nu(j_3, s_{2\text{out}}) = 8$. We compute unavoidable delays and νs prior to the optimization. Particular departure times of the trains are in the following range:

$$\begin{aligned} t_1 &= t(j_1, s_{1\text{out}}) \in \{4, 5, \ldots, 14\} \equiv \mathcal{T}_1 \\ t_2 &= t(j_2, s_{1\text{out}}) \in \{1, 2, \ldots, 11\} \equiv \mathcal{T}_2 \\ t_1^* &= t(j_1, s_{2\text{out}}) \in \{9, 10, \ldots, 19\} \equiv \mathcal{T}_1^* \\ t_2^* &= t(j_2, s_{2\text{out}}) \in \{10, 11, \ldots, 19\} \equiv \mathcal{T}_2^* \\ t_3 &= t(j_3, s_{2\text{out}}) \in \{8, 9, \ldots, 18\} \equiv \mathcal{T}_3. \end{aligned} \tag{56}$$

Now, we will investigate the linear programming approach and the time-indexed representation which leads to the QUBO formulation. For QUBO and HOBO representations, we use the following time indexed variables $x_{j_1,t_1,s_1} : t_1 \in \mathcal{T}_1$, $x_{j_1,t_1^*,s_2} : t_1^* \in \mathcal{T}_1^*$, $x_{j_2,t_2,s_1} : t_2 \in \mathcal{T}_2$, $x_{j_2,t_2^*,j_2} : t_2^* \in \mathcal{T}_2^*$, and $x_{j_3,t_3,s_2} : t_3 \in \mathcal{T}_3$.

From Eq. (20),

$$\begin{aligned} t(j_1, s_{2\text{in}}) &= t_1 + \tau^{(\text{pass})}(j_1, s_1, s_2) = t_1 + 4, \\ t(j_2, s_{2\text{in}}) &= t_2 + \tau^{(\text{pass})}(j_2, s_1, s_2) = t_2 + 8, \\ t(j_3, s_{1\text{in}}) &= t_3 + \tau^{(\text{pass})}(j_3, s_2, s_1) = t_3 + 8, \end{aligned} \tag{57}$$

and we will replace the occurrences of the variables on the left hand side using Eq. (57) in the ILP formulation. Note that we use t_1, t_2, t_3 only to compute the penalty for the delays.

In the QUBO formulation, we have the following penalty term from Eq. (41), ensuring that each train leaves each station only once:

$$\begin{aligned} &P_{\text{sum}}(\mathbf{x}) \\ &= p_{\text{sum}} \sum_{\substack{\mathcal{T} \in \{\mathcal{T}_1, \mathcal{T}_1^*, \mathcal{T}_2, \mathcal{T}_2^*, \mathcal{T}_3\} \\ (s,j) \in \{(s_1,j_1),(s_1,j_2),(s_2,j_3),(s_2,j_1),(s_2,j_2)\}}} \left(\sum_{(t,t') \in \mathcal{T}^{\times 2}, t \neq t'} x_{j,t,s} x_{j,t',s} - \sum_{t \in \mathcal{T}} x_{j,t,s} \right). \end{aligned} \tag{58}$$

As the default setting, we consider a double-track line, where each track has its own direction (unidirectional traffic). There is a conflict between j_1 and j_2 on the line from s_1 to s_2. If j_1 goes first at $t = 4$, then j_2 can start earliest at $t = 6$ (with an additional delay of 5) to proceed at full speed. If j_2 goes first at $t = 1$, then j_1 can start earliest at $t = 7$ (with an additional delay of 3) to proceed at a full speed. In both cases, j_3 can proceed undisturbed.

In the case of linear programming, the conflict can be resolved by setting the order variable $y(j_1, j_2, s_{1\text{out}}) \in \{0, 1\}$ to 1 if j_1 goes first and 0 if j_2 goes first. Recall that $y(j_2, j_1, s_{1\text{out}}) = 1 - y(j_1, j_2, s_{1\text{out}})$. Referring to Eq. (21),

$$\begin{cases} t_2 + M \cdot (1 - y(j_1, j_2, s_{1\text{out}})) - t_1 & \geq 2 + 0, \\ t_1 + M \cdot y(j_1, j_2, s_{1\text{out}}) - t_2 & \geq 2 + 4, \end{cases} \tag{59}$$

where M is a large number. Equivalently, $t_2 - 2 < t_1 < t_2 + 6$ is not allowed in the time-indexed variable approach. Hence, we have the following QUBO penalty term:

$$P_{\text{pair}}^{\text{headway}}(\mathbf{x}) = p_{\text{pair}} \sum_{t_2-2<t_1<t_2+6 \;\; t_1\in\mathcal{T}_1,\; t_2\in\mathcal{T}_2} x_{j_1,t_1,s_1} x_{j_2,t_2,s_1} + x_{j_2,t_2,s_1} x_{j_1,t_1,s_1}. \tag{60}$$

We can express the minimal stay condition in Eq. (23) as

$$t_1^* - (t_1 + 4) \geq 1 \text{ and } t_2^* - (t_2 + 8) \geq 1, \tag{61}$$

and the corresponding QUBO term would be

$$P_{\text{pair}}^{\text{stay}}(\mathbf{x}) = p_{\text{pair}} \left(\sum_{\substack{t_1'<t_1+5 \\ t_1'\in\mathcal{T}_1', t_1\in\mathcal{T}_1}} x_{j_1,t_1,s_1} x_{j_1,t_1',s_2} + \sum_{\substack{t_2'<t_2+9 \\ t_2'\in\mathcal{T}_2', t_2\in\mathcal{T}_2}} x_{j_2,t_2,s_1} x_{j_2,t_2',s_2} \right). \tag{62}$$

The track occupancy condition as defined in Eq. (26) for the track at platform 1 on station s_2, see Fig. 5 (both j_1 and j_2 are scheduled on this track), is expressed as

$$\begin{aligned} t_2 + 8 + M \cdot (1 - y(j_1, j_2, s_{2\text{out}})) - t_1^* &\geq 1, \\ t_1 + 4 + M \cdot y(j_1, j_2, s_{2\text{out}}) - t_2^* &\geq 1, \end{aligned} \tag{63}$$

and we have

$$y(j_1, j_2, s_{1\text{out}}) = y(j_1, j_2, s_{2\text{out}}) \tag{64}$$

as the M–P is not possible on this route (note that this last condition will be lifted while rerouting). In either case, the QUBO (HOBO) representation would be:

$$P^{\text{occ.}}_{\text{qubic}}(\mathbf{x}) = 2p_{\text{pair}}\left(\sum_{\substack{t_1^*\in\mathcal{T}_1^*,\ t_2^*\in\mathcal{T}_2^*\ t_1\in\mathcal{T}_1\\ t_1+4-1<t_2^*\le t_1^*}} x_{j_1,t_1,s_1}x_{j_1,t_1^*,s_2}x_{j_2,t_2^*,s_2} + \sum_{\substack{t_1^*\in\mathcal{T}_1^*,\ t_2\in\mathcal{T}_2\ t_2^*\in\mathcal{T}_2^*\\ t_2+8-1<t_1^*\le t_2^*}} x_{j_2,t_2,s_1}x_{j_1,t_1^*,s_2}x_{j_2,t_2^*,s_2}\right), \tag{65}$$

and for the decomposition we use: $\tilde{x}_{t_1^*,t_2^*} = x_{j_1,t_1^*,s_2}\cdot x_{j_2,t_2^*,s_2}$ (where we use abbreviation $\tilde{x}_{t_1^*,t_2^*}$ for $\tilde{x}_{j_1,j_2,t_1^*,t_2^*,s_2}$). The first part of the qubic penalty function is given by:

$$\begin{aligned} P^1_{\text{qubic}}(\mathbf{x},\tilde{\mathbf{x}}) = {} & p_{\text{pair}}\sum_{\substack{t_1^*\in\mathcal{T}_1^*,\ t_2^*\in\mathcal{T}_2^*\ t_1\in\mathcal{T}_1\\ t_1+4-1<t_2^*\le t_1^*}} (x_{j_1,t_1,s_1}\tilde{x}_{t_1^*,t_2^*} + \tilde{x}_{t_1^*,t_2^*}x_{j_1,t_1,s_1}) \\ & + p_{\text{pair}}\sum_{\substack{t_1^*\in\mathcal{T}_1^*,\ t_2\in\mathcal{T}_2\ t_2^*\in\mathcal{T}_2^*\\ t_2+8-1<t_1^*\le t_2^*}} (x_{j_2,t_2,s_1}\tilde{x}_{t_1^*,t_2^*} + \tilde{x}_{t_1^*,t_2^*}x_{j_2,t_2,s_1}). \end{aligned} \tag{66}$$

and

$$P^2_{\text{qubic}}(\mathbf{x},\tilde{\mathbf{x}}) = p_{\text{qubic}}\sum_{t_1^*\in\mathcal{T}_1^*,t_2^*\in\mathcal{T}_2^*} h(\tilde{x}_{t_1^*,t_2^*},\ x_{j_1,t_1^*,s_2},\ x_{j_2,t_2^*,s_2}), \tag{67}$$

where h is the polynomial from Eq. (52).

We use Eq. (8) for the objective. The ILP takes the following form:

$$\begin{aligned} \min.\ \ & w_{j_1}\frac{t(j_1,s_{1\text{out}}) - v(j_1,s_{1\text{out}})}{d_{\max}} + w_{j_2}\frac{t(j_2,s_{1\text{out}}) - v(j_2,s_{1\text{out}})}{d_{\max}} \\ & + w_{j_3}\frac{t(j_3,s_{2\text{out}}) - v(j_3,s_{2\text{out}})}{d_{\max}} \end{aligned} \tag{68}$$

subject to:

$$\begin{aligned} t_2 + M \cdot (1 - y(j_1, j_2, s_{1\text{out}})) - t_1 &\geq 2, \\ t_1 + M \cdot y(j_1, j_2, s_{1\text{out}}) - t_2 &\geq 6, \\ t_1^* - t_1 &\geq 5, \\ t_2^* - t_2 &\geq 9, \\ t_2 + M \cdot (1 - y(j_1, j_2, s_{1\text{out}})) - t_1^* &\geq -7, \\ t_1 + M \cdot y(j_1, j_2, s_{1\text{out}}) - t_2^* &\geq -3, \\ y(j_1, j_2, s_{1\text{out}}) &\in \{0, 1\}, \end{aligned}$$

and the range of the integer variables t_1, t_2, t_1^*, t_2^* are determined by Eq. (56). We use Eq. (64) for the simplification of the precedence variables.

In QUBO formulation, we have the following objective function:

$$\begin{aligned} f(\mathbf{x}) = w_{j_1} \sum_{t \in \mathcal{T}_1} x_{j_1,t,s_1} \frac{t - v(j_1, s_1)}{d_{\max}} + w_{j_2} \sum_{t \in \mathcal{T}_2} x_{j_2,t,s_1} \frac{t - v(j_2, s_1)}{d_{\max}} \\ + w_{j_3} \sum_{t \in \mathcal{T}_3} x_{j_3,t,s_2} \frac{t - v(j_3, s_2)}{d_{\max}}. \end{aligned} \tag{69}$$

The overall QUBO representation is expressed as

$$\begin{aligned} \min_{\mathbf{x},\tilde{\mathbf{x}}} q(\mathbf{x}, \tilde{\mathbf{x}}) = f(\mathbf{x}) + P_{\text{sum}}(\mathbf{x}) + P_{\text{pair}}^{\text{headway}}(\mathbf{x}) + P_{\text{pair}}^{\text{stay}}(\mathbf{x}) + P_{\text{qubic}}^{1}(\mathbf{x}, \tilde{\mathbf{x}}) \\ + P_{\text{qubic}}^{2}(\mathbf{x}, \tilde{\mathbf{x}}). \end{aligned} \tag{70}$$

If j_1 goes first, $y(j_1, j_2, s_{1\text{out}}) = 1$, then we have an additional delay of 5 from j_2, adding $1 \cdot \frac{5}{10} = 0.5$ to the objective. If j_2 goes first, $y(j_1, j_2, s_{1\text{out}}) = 0$, we have an additional delay of 3 from j_1, adding $2 \cdot \frac{3}{10} = 0.6$ to the objective. Therefore, at this stage, the best solution is to let j_1 go first, yielding $t_1 = 4$, $t_2 = 6$ and $t_3 = 8$.

Suppose now that we find the value of the objective not satisfactory. In this case, we need to perform rerouting. In our case, the rerouting will concern changing the double-track line to the bidirectional traffic mode (many railway operators are being involved in such rerouting, e.g., Koleje Śląskie, eng. Silesian Railways). In details, there is a conflict between the trains j_1 and j_2 on the line between s_1 and s_2. Hence, rerouting will be used to solve this conflict: We use the line between s_1 and s_2 as two parallel single-track lines (Track 1 for j_1 and Track 2 for j_2). In this case, we have no conflict between j_1 and j_2 and we lift the conditions in Eqs. (59) and (64) (as M–P is now possible on the line), or remove the corresponding terms from the QUBO in Eq. (60). However, a new conflict arises between j_2 and j_3 on the single track resource (Line 2), so new conditions or terms will appear. Following Eq. (22), the single track

line condition yields:

$$\begin{cases} t_3 + M \cdot (1 - z(j_2, j_3, s_1, s_2)) - (t_2 + 8) \geq 1, \\ t_2 + M \cdot z(j_2, j_3, s_1, s_2) - (t_3 + 8) \geq 1, \end{cases} \tag{71}$$

as $\tau^{(\text{pass})}(j_2, s_1, s_2) = 8$, and $\tau^{(\text{pass})}(j_3, s_2, s_1) = 8$. Equivalently we cannot have $t_3 - 8 < t_2 < t_3 + 8$) and we have the following QUBO penalty term:

$$P^{\text{1track}}_{\text{pair}}(\mathbf{x}) = p_{\text{pair}} \sum_{t_3-8-1<t_2<t_3+8+1 \ t_2 \in \mathcal{T}_2, t_3 \in \mathcal{T}_3} x_{j_2,t_2,s_1} x_{j_3,t_3,s_2} + x_{j_3,t_3,s_2} x_{j_2,t_2,s_1}, \tag{72}$$

The objective would be as in Eq. (68), but subject to altered constraints:

$$\begin{aligned} t_3 + M \cdot (1 - z(j_2, j_3, s_1, s_2)) - t_2 &\geq 9, \\ t_2 + M \cdot z(j_2, j_3, s_1, s_2) - t_3 &\geq 9, \\ t_1^* - t_1 &\geq 5, \\ t_2^* - t_2 &\geq 9, \\ t_2 + M \cdot (1 - y(j_1, j_2, s_{2\text{out}})) - t_1^* &\geq -7, \\ t_1 + M(y(j_1, j_2, s_{2\text{out}})) - t_2^* &\geq -3, \\ y(j_1, j_2, s_{2\text{out}}) \in \{0, 1\}, \ z(j_2, j_3, s_1, s_2) &\in \{0, 1\}, \end{aligned} \tag{73}$$

and the ranges of the integer variables t_1, t_2, t_1^*, t_2^* are determined by Eq. (56).

The effective QUBO representation here is given as

$$\begin{aligned} \min_{\mathbf{x},\tilde{\mathbf{x}}} q^r(\mathbf{x}, \tilde{\mathbf{x}}) = f(\mathbf{x}) + P_{\text{sum}}(\mathbf{x}) + P^{\text{stay}}_{\text{pair}}(\mathbf{x}) + P^{\text{1track}}_{\text{pair}}(\mathbf{x}) \\ + P^{1}_{\text{qubic}}(\mathbf{x}, \tilde{\mathbf{x}}) + P^{2}_{\text{qubic}}(\mathbf{x}, \tilde{\mathbf{x}}). \end{aligned} \tag{74}$$

If j_3 goes first ($z(j_2, j_3, s_1, s_2) = 0$), the additional delay of j_2 would exceed the maximal $d_{\max} = 10$. The optimal solution is $z(j_2, j_3, s_1, s_2) = 1$ and $y(j_1, j_2, s_{2\text{out}}) = 1$, hence $t_1 = 4$, $t_2 = 2$, $t_3 = 11$, and $t_1^* = 9$. The additional delay of j_1 is 0, j_2 is 1, and j_3 is 3 with the objective 0.4, which is better than the objective of the default settings. As there is no possibility to reroute trains further to lift the conflict between j_2 and j_3, we can consider this objective as the optimal one.

4.1 Numerical calculations

In this section, we present a proof-of-concept by solving the small numerical example described above using D-Wave solvers.

We first solved the problem using the ILP formulation to test the validity of the model. We used Python 3.9 programming language and PulP library [38] to implement the ILP formulation and CBC (Coin-or branch and cut) [39] solver to solve the problem,

Table 3 Solutions obtained from the linear solver

(a) Default Settings

	S_1	S_2		S_1
	Dep.	Arr.	Dep.	Arr.
j_1	4	8	9	
j_2	6	14		
j_3			8	16

(b) Rerouting

	S_1	S_2		S_1
	Dep.	Arr.	Dep.	Arr.
j_1	4	8	9	
j_2	2	10		
j_3			11	19

which is the default solver in PulP, to test the validity of the model. We reached $t_1 = 4$, $t_2 = 6$ and $t_3 = 8$ for the default settings (with objective 0.5), see Table 3a, and $t_1 = 4$, $t_2 = 2$ and $t_3 = 11$ for the rerouting (with objective 0.4) as expected, see Table 3b. Note that we are not interested in the run-time comparison between the linear solver and D-Wave, but we would like to demonstrate the potential of quantum annealing for solving train rescheduling problems.

We implemented the QUBO formulation presented in Sect. 3.3 using D-Wave Ocean SDK. For the numerical calculations on the D-Wave machine, we need to pick particular penalty values. The theory of penalty methods is discussed, for example, in [34]. In general, the solution of the unconstrained objective tends to be a feasible optimal solution to the original problem as the penalties of constraints tend to infinity. However, in practice, these penalties have to be chosen so that the constraint terms do not dominate over the objective. If the penalties are too high, the objective can be lost in the noise of the physical quantum annealer. Based on these heuristics, we used the following strategy in the determination of penalties:

1. Penalties for the hard constraints (p_{sum}, $2p_{\text{pair}}$, and p_{qubic}) should be higher than the maximal possible objective for the single train, i.e., $\max_j w_j = 2$
2. p_{qubic} should be smaller than other hard penalties, as it is multiplied by 3 or 2 in some terms of HOBO—see Eq. (52).
3. We pick penalties of hard constraints as low as possible, to prevent the objective from being overridden by the noise of the quantum annealer.

The terms in Eq. (69) (the maximal penalty here is $w_{j_1} = 2.$) are "soft constraints," and the terms in Eqs. (70) and (74) are the "hard constraints" that cannot be broken for the solution to be feasible. Hence, we use the following penalty parameters $p_{\text{sum}} = 2.5$, $p_{\text{pair}} = 1.25$ (as each element is taken twice) and $p_{\text{qubic}} = 2.1$.

Both for the default settings and rerouting, we had 176 logical variables, out of which 55 were the $\mathbf{x}$ variables and 121 were the auxiliary $\tilde{\mathbf{x}}$ variables. Here, we have a relatively large overhead due to the cubic term. Hence, the single track occupation condition has to be used with caution when handling large railway problems. To test the validity of the model, we first solved the two problems using the simulated annealer (SA) from the D-Wave Ocean SDK, which is a classical heuristic algorithm for solving combinatorial optimization problems stated as QUBOs. When running SA or QA, the output is a list of samples (0-1 assignments to the binary variables) and the corresponding energies (value of $q(x)$). The lowest energy solution is called the ground state. Using SA, we got the same solutions as the linear solver with the following energies $q(\mathbf{x}, \tilde{\mathbf{x}}) = -12.0$ and $q^r(\mathbf{x}, \tilde{\mathbf{x}}) = -12.1$. The energies correspond

to the ground state as -12.5 is the offset (the constant term in the QUBO formulation), and 0.5 and 0.4 are the optimal (lowest possible) penalties for delays.

Next, we solved the problem on D-Wave Advantage quantum processing unit (QPU) [10]. In D-Wave Advantage QPU, not all the qubits are interconnected via couplers, and the underlying graph has the specific structure known as the Pegasus topology [40]. Hence, before running a problem on the D-Wave, a procedure called *minor embedding* is required to map the logical variables to the physical qubits on the machine. Due to limited connectivity, a single logical qubit is often represented with a *chain* of physical qubits that are coupled strong enough so that they end up in the same value representing the same variable. The coupling between the qubits in the chain is known as the *chain strength*, and a low chain strength may result in chain breaks while a high chain strength may override the problem parameters. In our experiments on D-Wave Advantage, we used the default minor embedding algorithm provided by Ocean SDK and used various chain strengths. The number of logical variables is 176, and the number of physical qubits used in the machine after embedding is $\sim$ 900. For both problems, the degree of completeness of the problem graph was approximately 0.1.

Another parameter that needs to be set is the annealing time. Annealing time depends on the problem and problem size and is also limited by the current technology of D-Wave Advantage QPU. In our experiments, the annealing time is set as $250\mu s$. Results of the D-Wave experiments are presented in Fig. 6. A solution is feasible, if it can be technically realized on the railroad infrastructure, i.e., all hard constraints are fulfilled. A solution is optimal if the order of the trains on conflicting resources (i.e., tracks that are used by more than one train) is the same as the order in the ground state solution. We reached optimal solutions (in the sense of the train order) using the D-Wave machine, both for the default settings and rerouting.

For the default settings, D-Wave results for chain strength 4 are: $t_1 = 4$, $t_2 = 8$ (adding an additional 0.7 to the objective), $t_3 = 9$ (adding an additional 0.1 to the objective) and $t_1^* = 10$, see Table 4a. The solution is feasible, since j_2 leaves s_1 at $t_2 = 8$, late enough to have no conflict with j_1 which will leaves s_1 at $t_1 = 4$. Furthermore, j_2 will arrive to s_2 at $t_2 + \tau^{(\text{pass})}(j_2, s_1, s_2) = 8 + 8 = 16$, i.e., after j_1 leaves s_2 at $t_1' = 11$. The order of trains is the same as in the optimal solution, and the energy of the state is -11.7. This energy does not correspond to the ground state as there are some additional delays of the trains, however, do not affect the feasibility and the order of trains.

For rerouting, the results of D-Wave are: $t_1 = 6$, (adding an additional 0.4 to the objective) $t_2 = 4$ (adding an additional 0.3 to the objective), $t_3 = 13$ (adding an additional 0.5 to the objective) and $t_1^* = 11$, see Table 4b. The solution is feasible, since j_2 will arrive to s_2 at $t_2 + \tau^{(\text{pass})}(j_2, s_1, s_2) = 4 + 8 = 12$, i.e., after j_1 leaves s_2 at $t_1^* = 11$, and before j_3 leaves s_2 at $t_3 = 13$. The order of trains is the same as in the optimal solution, the state energy is -11.3. Like in the default settings case, the found solution is feasible but not the ground state.

Another alternative is to use the hybrid solver for binary quadratic models provided by D-Wave. The hybrid solver runs in parallel modules consisting of a heuristic classical component to explore the search space and a quantum component that makes queries to D-Wave QPU to guide the optimization process and improve the existing

Table 4 Solutions obtained from the D-Wave

(a) Default settings

	S_1	S_2		S_1
	Dep.	Arr.	Dep.	Arr.
j_1	4	8	10	
j_2	8	16		
j_3			9	17

(b) Rerouting

	S_1	S_2		S_1
	Dep.	Arr.	Dep.	Arr.
j_1	6	10	11	
j_2	4	12		
j_3			13	21

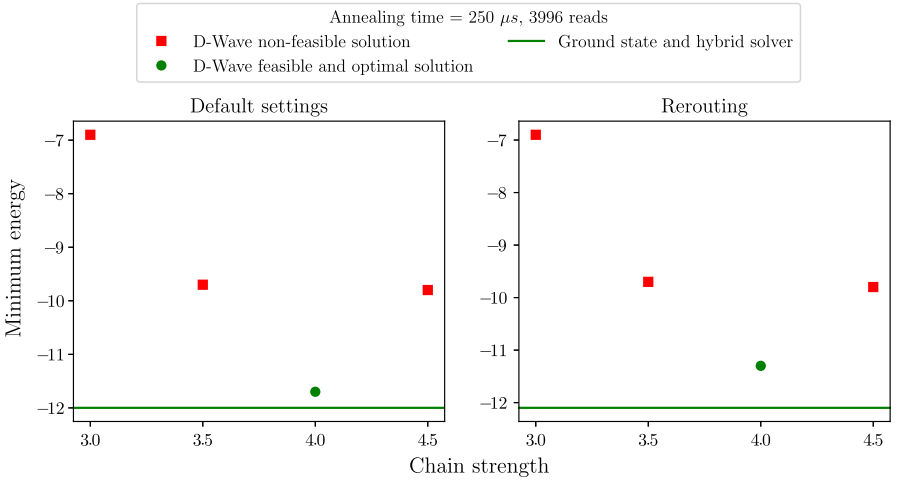


Fig. 6 Lowest energy solutions obtained from D-Wave Advantage and D-Wave hybrid solvers. In the case of D-Wave Advantage, only one feasible solution was found for each panel. For chain strength 4.0 at each panel, the percentage of feasible solutions over total number of solutions is roughly 2.5×10^{-4}

solutions. The best solution found among the parallel runs is returned to the user [41]. Using the hybrid solver, we obtained the ground state, both in the case of default settings and the rerouted setting.

With our example, we have demonstrated that although it is possible to have the optimal solution for the D-Wave, it is not straightforward and requires at least an extensive parameter sweep. On the other hand, the D-Wave hybrid solver found the ground state on the first try. More importantly, the hybrid solver can be used for tackling larger problems as those solvers can work on problem instances with up to 20,000 variables that are fully connected or up to 1 million variables for sparse Q matrices [42].

4.2 Assessment of solvers on larger instances

To demonstrate the feasibility of the hybrid solver, we have assessed both the D-Wave Advantage and the D-Wave hybrid solver on a bit larger examples. In both examples, we use the same parameters settings as in Sect. 4.1 for the calculations.

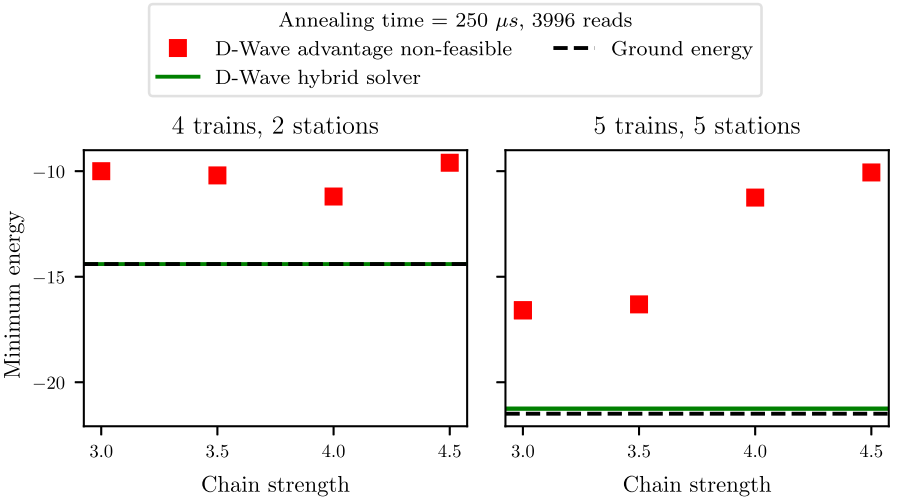


Fig. 7 Lowest energy solutions obtained from D-Wave Advantage and D-Wave hybrid solvers for larger problems. We present also comparison with the ground state achieved from the ILP (classical) approach

The first is a bit enlarged default setting one with infrastructure as in Fig. 5. Here, in addition, train j_3 is followed by another stopping train j_4, and the conflict occurs on the minimal headway between j_3 and j_4. We call the problem the 4 trains 2 stations example. The problem is a bit larger with 187 logical variables. Although the number of connections is larger, the degree of completeness of the graph is a bit smaller and equals roughly 0.09. The ground state energy, consistent with the solution of the ILP, equals to $q(\mathbf{x}, \tilde{\mathbf{x}}) = -14.4$.

The second example concerns a larger number of trains and a larger number of stations on a more branched network. We call the problem the 5 trains 5 stations example. The problem is encoded on 341 logical variables, but with a much smaller degree of completeness of the graph which equals roughly 0.04. The ground state energy, consistent with the solution of the ILP, equals to $q(\mathbf{x}, \tilde{\mathbf{x}}) = -21.49$.

Results of calculations for both additional examples are presented in Fig. 7. As we can see for slightly larger problems than in Sect. 4.1, the D-Wave Advantage does not give any feasible solution. The D-Wave hybrid solver, on the other hand, still has promising outcomes. Actual characteristics of the problem are presented in Table 5. Here, we have observed that the larger the railway problem is, the smaller the degree of completeness. This observation coincides with Tab. IV [16] and discussion in Sect. 3 as the number of variables and number of nonzero QUBO terms are roughly linear in the number of trains and stations. Referring to Figs. 6, 7, and Tab. IV in [16], we can generally conclude that smaller railway problems, with the graph's degree of completeness of 0.1 or larger, are solvable on the D-Wave machine without the need for the D-Wave hybrid solver. For larger problems, the hybrid solver is necessary.

From a practical point of view, the above-presented problems are still of small size due to the small size of the current D-Wave machine. To estimate the amount of logical resources needed to solve real-life problems, let us consider an hour cycle on the dense

Table 5 Problem characteristics

	Problem			
	Default	Rerouted	4 Trains 2 stations	5 Trains 5 stations
# logical vars.	176	176	187	341
# edges	1492	1504	1577	2159
Density versus full graph	0.097	0.098	0.091	0.037
# physical vars.	869	739	979	1419

The first 3 rows refer to the characteristics of the QUBO and the last one to the characteristics of the solver. Number of physical variables is computed via D-Wave's default embedding algorithm minorminer which is a heuristic algorithm; hence, the data are approximate

traffic (one train per 2 min. in each direction) on the double-track metro line with 20 stations. (In an hour cycle, we have 60 trains.) We then consider $d_{\max} = 5$ min, and 1 min resolution. According to Eq. (49), we would have roughly 7_200 variables. If each train is assumed to be in potential conflict with $H(d_{\max}) = 5$ other trains (that many trains pass in $2d_{\max} = 10$ min. interval), then according to Eq. (55) we will have roughly 216_000 auxiliary variables. (Obviously, in both cases, the particular number of variables depends on the details of the topology of the problem.) Such a problem would be solvable on the not very large device but with possible third-order connections or a much larger one with second-order connections only.

5 Conclusions and outlook

As current classical technologies are insufficient to compute feasible results in a reasonable time, fully developed quantum annealing offers potential polynomial speed-ups for these problems. However, switching from the conventional notation to the one demanded by the quantum annealer is a challenge. Our paper is the first to present the quadratic and higher-order unconstrained binary optimization representation for the railway rescheduling problem concerning the determination of the order of trains on conflicted resources, rescheduling, and rerouting trains on single-, double- and multi-track lines and stations.

The number of qubits is one of the bottlenecks of current quantum devices. It is thus desirable to use the smallest possible number of qubits when modeling. When quadratic and higher-order models are compared, the latter is more efficient in terms of the number of qubits required. Although currently, it is not possible to utilize HOBO with quantum annealers, the need for quantum annealers allowing such interactions is evident [43]. There is also ongoing work for building architectures that allow solving optimization problems involving higher-order terms directly [44] in the gate-based model. Furthermore, algorithms like quantum approximate optimization algorithm (QAOA) [26] allow solving higher-order problems natively [13, 14].

Four demonstrative problems were implemented on the current D-Wave machine. Two smaller problems were successfully solved both using the D-Wave Advantage QPU and using the D-Wave hybrid solver. Two larger problems were successfully

solved only on D-Wave hybrid solver which we find promising for solving larger instances. Importantly, we have presented the HOBO/QUBO formulation that can be used with quantum-inspired architectures designed for solving combinatorial optimization problems stated in QUBO form such as Fujitsu digital annealers [45].

Determination of penalty values poses a challenge for solving QUBO problems in general. Although we have determined the penalty values using heuristic methods, note that there are some recent algorithms dedicated to penalty determination like the cross-entropy optimization discussed in [46] and the one discussed in [47] (see, e.g., Sect. 3.2) is tested successfully on the particular Fujitsu digital annealer.

Curiosity arises on how quantum annealers or other Ising-based heuristics behave in solving real-life problems compared to conventional methods. Further research should be undertaken to explore the applicability of the presented approach for real-life train rerouting and rescheduling problems. In particular, when considering the railway traffic on the regional scale where delays can be large and the number of trains is not very large, the QUBO formulation that will be obtained from the ILP representation presented in this paper may be worth investigating.

Besides wide railway potential applications (ordinary railways, metro, trams), discussed rules of problem conversion into HOBO / QUBO can be applied generically in many branches of operational research. Let us list a few:

1. Electric bus scheduling, where the charging place occupation condition can be modeled in analogy to our track occupation condition.
2. Automated guided vehicle (AGV) scheduling in the factory, where there are many railway analogies. AGVs have a pre-designed schedule that is conflicted and needs to be rescheduled. AGVs follow the paths that are uni- or bi-directional; hence, there is a headway and single track line condition. There are places that can be occupied by one AGV at a time (track occupation condition), paths of AGVs cross (switch condition), and there is the sequence of tasks for the given trolley (rolling stock circulation condition). Rerouting of AGVs can be treated as an extra task beyond the optimization as in Fig. 4, and finally, AGVs may have various priorities.

In general, our HOBO approach (generated by track occupation condition) may be applicable for models consisting of "stations" that can be occupied by only one "vehicle" at a time, with waiting possibilities on stations and no-waiting elsewhere.

Acknowledgements The research was supported by the Foundation for Polish Science (FNP) under grant number TEAM NET POIR.04.04.00-00-17C1/18-00 (KD); and the National Science Centre (NCN), Poland, under project number 2019/33/B/ST6/02011 (A.K. and Ö.S.) and by the Silesian University of Technology Rector's Grant no. BKM-700/RT2/2022 12/020/BKM2022/0233 (KK). We acknowledge the cooperation with Koleje Śląskie sp. z o.o. (eng. Silesian Railways) and appreciate the valuable and substantive discussions. We acknowledge the consultation with company Aiut ltd. (Wyczółkowskiego street Gliwice) on the rescheduling/rerouting of industrial trolleys (AGVs).

Data availability The code and the data used for generating the numerical results can be found in https://github.com/iitis/railways_HOBO.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included

in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

A Appendix: Railway terminology

For the clarity of presentation for the non-railway community, we include the clear definitions of railway terminology in the form of Table 6.

Table 6 Summary of definitions of railway terminologies

Railway terms	Definitions
Schedule	Pre-set sequences of blocks with departure times assigned to selected station blocks
Station blocks	Track section between two signalling utilities at stations.
Line blocks	Track section between two signalling utilities on lines.
Rerouting	Change of the track used by a train within a line or a station
Rescheduling	Train departure time modification in a way to avoid conflict
Meet and pass (M–P)	Meeting of two trains at the same spatial location while following the same route in opposite directions.
Meet and overtake (M–O)	Overtaking one train by another while following the same route in the same direction.
Unavoidable delay	Delay from outside the model that is propagated through the network, not including any delay that may be caused by other trains' traffic
Additional delay	Delay beyond unavoidable caused by solving conflicts due to traffic, that is in control of our model
Headway	Minimal time span between trains.

References

1. Mulley, C., Nelson, J.D., Ison, S., Tsoi, K.H., Loo, B.P.Y.: The Routledge Handbook of Public Transport. Routledge, London (2021)
2. Batty, P., Palacin, R., González-Gil, A.: Challenges and opportunities in developing urban modal shift. Travel Behav. Soc. **2**(2), 109–123 (2015). https://doi.org/10.1016/j.tbs.2014.12.001
3. Gawlak, K.: Mapa zdarzeń i wydarzeń kolejowych w roli innowacyjnego projektu wspomagajacego system zarzadzania bezpieczeństwem regionalnego przewoźnika Koleje Śląskie sp. z o.o. (Map of railroad events as an innovative project supporting safety management system of the regional carrier Koleje Śląskie sp. z o.o.). Urzad Transprotu Kolejowego (Eng. Office of Rail Transport) (2022). https://utk.gov.pl/download/1/67088/MKBekran6122021.pdf
4. Larsen, R., Pranzo, M., D'Ariano, A., Corman, F., Pacciarelli, D.: Susceptibility of optimal train schedules to stochastic disturbances of process times. Flex. Serv. Manuf. J. **26**(4), 466–489 (2014)
5. Törnquist, J., Persson, J.A.: N-tracked railway traffic re-scheduling during disturbances. Transp. Res. Part B: Methodol. **41**(3), 342–362 (2007). https://doi.org/10.1016/j.trb.2006.06.001

6. Corman, F., D'Ariano, A., Pranzo, M., Hansen, I.A.: Effectiveness of dynamic reordering and rerouting of trains in a complicated and densely occupied station area. Transp. Plan. Technol. **34**(4), 341–362 (2011)
7. Apolloni, B., Carvalho, C., De Falco, D.: Quantum stochastic optimization. Stoch. Process. Appl. **33**(2), 233–244 (1989)
8. Kadowaki, T., Nishimori, H.: Quantum annealing in the transverse Ising model. Phys. Rev. E **58**(5), 5355 (1998)
9. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum computation by adiabatic evolution. arXiv preprint arXiv:quant-ph/0001106 (2000)
10. Johnson, M.W., et al.: Quantum annealing with manufactured spins. Nature **473**(7346), 194–198 (2011)
11. Glover, F., Kochenberger, G., Du, Y.: A tutorial on formulating and using QUBO models. arXiv preprint arXiv:1811.11538 (2018)
12. Lucas, A.: Ising formulations of many NP problems. Front. Phys. **2**, 5 (2014)
13. Glos, A., Krawiec, A., Zimborás, Z.: Space-efficient binary optimization for variational quantum computing. NPJ Quantum Inf. **8**(1), 1–8 (2022)
14. Tabi, Z., El-Safty, K.H., Kallus, Z., Hága, P., Kozsik, T., Glos, A., Zimborás, Z.: Quantum optimization for the graph coloring problem with space-efficient embedding. In: 2020 IEEE International Conference on Quantum Computing and Engineering (QCE), pp. 56–62. IEEE (2020)
15. Salehi, Ö., Glos, A., Miszczak, J.A.: Unconstrained binary models of the travelling salesman problem variants for quantum optimization. Quantum Inf. Process. **21**(2), 1–30 (2022)
16. Domino, K., Koniorczyk, M., Krawiec, K., Jałowiecki, K., Deffner, S., Gardas, B.: Quantum annealing in the NISQ era: railway conflict management. arXiv preprint arXiv:2112.03674 (2021)
17. Mattsson, L.-G., Jenelius, E.: Vulnerability and resilience of transport systems—a discussion of recent research. Transp. Res. Part A: Policy Pract. **81**, 16–34 (2015). https://doi.org/10.1016/j.tra.2015.06.002
18. Wen, C., Huang, P., Li, Z., Lessan, J., Fu, L., Jiang, C., Xu, X.: Train dispatching management with data-driven approaches: a comprehensive review and appraisal. IEEE Access **7**, 114547–114571 (2019). https://doi.org/10.1109/ACCESS.2019.2935106
19. Cordeau, J.-F., Toth, P., Vigo, D.: A survey of optimization models for train routing and scheduling. Transp. Sci. **32**(4), 380–404 (1998). https://doi.org/10.1287/trsc.32.4.380
20. Corman, F., Meng, L.: A review of online dynamic models and algorithms for railway traffic management. IEEE Trans. Intell. Transp. Syst. **16**(3), 1274–1284 (2015). https://doi.org/10.1109/TITS.2014.2358392
21. Liu, S.Q., Kozan, E.: Scheduling trains as a blocking parallel-machine job shop scheduling problem. Comput. Oper. Res. **36**(10), 2840–2852 (2009). https://doi.org/10.1016/j.cor.2008.12.012
22. Corman, F., D'Ariano, A., Hansen, I.A., Pacciarelli, D., Pranzo, M.: Dispatching trains during seriously disrupted traffic situations. In: 2011 International Conference on Networking, Sensing and Control, pp. 323–328. https://doi.org/10.1109/ICNSC.2011.5874901 (2011)
23. Harrod, S.: Capacity factors of a mixed speed railway network. Transp. Res. Part E: Logist. Transp. Rev. **45**(5), 830–841 (2009). https://doi.org/10.1016/j.tre.2009.03.004
24. Borowski, M., et al.: New hybrid quantum annealing algorithms for solving vehicle routing problem. In: International Conference on Computational Science, pp. 546–561. Springer (2020)
25. Neukart, F., Compostella, G., Seidel, C., Von Dollen, D., Yarkoni, S., Parney, B.: Traffic flow optimization using a quantum annealer. Front. ICT **4**, 29 (2017)
26. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014)
27. D'Ariano, A., Pacciarelli, D., Pranzo, M.: A branch and bound algorithm for scheduling trains in a railway network. Eur. J. Oper. Res. **183**(2), 643–657 (2007)
28. Lange, J., Werner, F.: Approaches to modeling train scheduling problems as job-shop problems with blocking constraints. J. Sched. **21**(2), 191–207 (2018)
29. Szpigel, B.: Optimal train scheduling on a single line railway. J. Oper. Res. **72**, 344–351 (1973)
30. Sotskov, Y.N., Shakhlevich, N.V.: NP-hardness of shop-scheduling problems with three jobs. Discrete Appl. Math. **59**(3), 237–266 (1995)
31. Pinedo, M.L.: Scheduling, vol. 29. Springer, New York (2012)
32. Boros, E., Hammer, P.L.: Pseudo-boolean optimization. Discrete Appl. Math. **123**(1–3), 155–225 (2002)

33. Glover, F., Hao, J.-K., Kochenberger, G.: Polynomial unconstrained binary optimisation—part 1. Int. J. Metaheuristics **1**(3), 232–256 (2011)
34. Luenberger, D.G., Ye, Y.: Linear and Nonlinear Programming. International Series in Operations Research & Management Science. Springer, New York (1984)
35. Mandal, A., Roy, A., Upadhyay, S., Ushijima-Mwesigwa, H.: Compressed quadratization of higher order binary optimization problems. In: Proceedings of the 17th ACM International Conference on Computing Frontiers, pp. 126–131 (2020)
36. Rosenberg, I.G.: Reduction of bivalent maximization to the quadratic case. Cahiers du Centre d'Etudes de Recherche Operationnelle (1975)
37. Shimada, D., Shibuya, T., Shibasaki, T.: A decomposition method for makespan minimization in job-shop scheduling problem using ising machine. In: IEEE 8th International Conference on Industrial Engineering and Applications (ICIEA), pp. 307–314. IEEE (2021)
38. Mitchell, S., Consulting, S.M., Dunning, I.: PuLP: A Linear Programming Toolkit for Python. The University of Auckland, Auckland (2011)
39. Forrest, J., Lougee-Heimer, R.: CBC user guide: emerging theory, methods, and applications. In: INFORMS TutORials in Operations Research, pp. 257–277 (2005)
40. Dattani, N., Szalay, S., Chancellor, N.: Pegasus: the second connectivity graph for large-scale quantum annealing hardware. arXiv preprint arXiv:1901.07636 (2019)
41. D-Wave: D-Wave Hybrid Solver Service + Advantage: Technology Update [WhitePaper]. https://www.dwavesys.com/media/m2xbmlhs/14-1048a-a_d-wave_hybrid_solver_service_plus_advantage_technology_update.pdf (2020)
42. McGeoch, C., Farré, P., Bernoudy, W.: D-Wave Hybrid Solver Service+ Advantage: Technology Update. Technical report, D-Wave Systems Inc, Burnaby, BC, Canada (2020)
43. Perdomo-Ortiz, A., Feldman, A., Ozaeta, A., Isakov, S.V., Zhu, Z., O'Gorman, B., Katzgraber, H.G., Diedrich, A., Neven, H., de Kleer, J., et al.: Readiness of quantum optimization machines for industrial applications. Phys. Rev. Appl. **12**(1), 014004 (2019)
44. Ender, K., ter Hoeven, R., Niehoff, B.E., Drieb-Schön, M., Lechner, W.: Parity quantum optimization: compiler. arXiv preprint arXiv:2105.06233 (2021)
45. Tsukamoto, S., Takatsu, M., Matsubara, S., Tamura, H.: An accelerator architecture for combinatorial optimization problems. Fujitsu Sci. Tech. J. **53**(5), 8–13 (2017)
46. Roch, C., Impertro, A., Linnhoff-Popien, C.: Cross entropy optimization of constrained problem Hamiltonians for quantum annealing. In: International Conference on Computational Science, pp. 60–73. Springer (2021)
47. Ayodele, M., Allmendinger, R., López-Ibáñez, M., Parizy, M.: Multi-objective QUBO solver: bi-objective quadratic assignment. arXiv preprint arXiv:2205.13399 (2022)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.