Application of a Hybrid Algorithm Based on Quantum Annealing to Solve a Metropolitan Scale Railway Dispatching Problem

Mátyás Koniorczyk,* Krzysztof Krawiec,† Ludmila Botelho,‡ Nikola Bešinović,§ Krzysztof Domino ¶

September 14, 2023

Abstract

We address the applicability of quantum-classical hybrid solvers for practical railway dispatching/conflict management problems, with a demonstration on real-life metropolitan-scale network traffic. The railway network includes both single-and double segments and covers all the requirements posed by the operator of the network. We build a linear integer model for the problem and solve it with D-Wave's quantum-classical hybrid solver as well as with CPLEX for comparison. The computational results demonstrate the readiness for application and benefits of quantum-classical hybrid solvers in the a realistic railway scenario: they yield acceptable solutions on time; a critical requirement in a dispatching situation. Though they are heuristic they offer a valid alternative and outperform classical solvers in some cases.

1 Introduction

Rail transport is expected to experience an increase in capacity demands due to changes in mobility needs resulting from climate policy, leading to traffic

*koniorczyk.matyas@wigner.hu, ORCID: 0000-0002-2710-493X, Wigner Research Centre for Physics, Konkoly-Thege M. út 29-33, H-1525 Budapest, Hungary

†krzysztof.krawiec@polsl.pl, ORCID: 0000-0002-7447-3447, Silesian University of Technology, Faculty of Transport and Aviation Engineering, Akademicka 2A, 44-100 Gliwice, Poland

‡lbotelho@iitis.pl, ORCID: 0000-0003-4792-9768, Institute of Theoretical and Applied Informatics Polish Academy of Sciences Bałtycka 5, 44-100 Gliwice, Poland, Joint Doctoral School, Silesian University of Technology, Akademicka 2A, 44-100 Gliwice, Poland

§nikola.besinovic@tu-dresden.de ORCID: 0000-0003-4111-2255, "Friedrich List" Faculty of Transport and Traffic Sciences, Technical University of Dresden, 01069 Dresden, Germany

¶kdomino@iitis.pl, ORCID: 0000-0001-7386-5441, Institute of Theoretical and Applied Informatics Polish Academy of Sciences Bałtycka 5, 44-100 Gliwice

challenges in passenger and cargo rail transport. The situation is aggravated by the fact that the rolling stock and especially railway infrastructure cannot keep up with the increase in transport needs, which makes railway systems overloaded.

Rail transport, due to its technical and organisational characteristics is very sensitive to disturbances in traffic: these extraordinary events have an impact on railway operations, typically resulting in delays [1]. Examples of such disturbances include: late train departures/arrivals, extended dwell times, or (partial) track closures[1]. These can last from several minutes up to hours. The impact of disturbances can propagate to multiple sections in the railway network (cf. [2] and references therein). Thus, ensuring stable railway traffic and providing reliable service for passengers, rail cargo companies, and their clients is in the best common interest of railway infrastructure managers, and train operators. To limit these impacts as much as possible it is necessary to make proper dispatching decisions quickly. The dispatchers need to reschedule and partially reroute the trains, aiming at the minimisation of negative consequences. Still, in many places, dispatchers are using their own intuition or simple heuristics like FCFS (First Come First Served); resulting in decisions far from the objective of consequence minimisation. In the last decades, there has been a growing research interest in mathematical optimisation methods in support of rail dispatchers in decision making. While doing so, diverse objective functions were addressed, such as the (weighted) sum of delays [3], the maximal delay that cannot be avoided [4], or fuel consumption measures [5].

As a railway network is a complex non-local structure, modeling a bigger portion of it is necessary for efficient suppression of the consequences of disturbances. Hence, large-scale dispatching problems have to be addressed, and as the time to come up with a decision is limited, they have to be solved almost real time. Therefore it is vital to develop efficient algorithms. However, the current conventional rescheduling optimisation models have difficulties addressing complex and large-scale instances in suitable computation time [6, 7, 8].

The railway dispatching problem can broadly be recognised as being equivalent to job-shop scheduling with blocking and no-wait constraints [9, 10]. A possible modeling approach is based on alternative graphs employing order and precedence variables [3], facilitating the formulation as a mixed integer linear program which is often large, hence, specialised algorithms are often used instead [8, 11]. An alternative approach is to use time-indexed models: discrete-time units and binary decision variables that assign events to particular time instants. While this approach leads to very large problems, this approach is applied both in timetabling as well as dispatching/rescheduling [6, 12, 7, 13].

[1]Closures may be referred to as disruptions.

As time indexed models result in integer or 0-1 programs, they are suitable [14] for to be solved on new types of hardware: quantum annealers [15, 16]. These physical devices can be considered as stochastic heuristic solvers for Quadratic Unconstrained Binary Optimisation (QUBO) problems, and there are few commercially available options, including D-Wave [17]. The exploration of their potential applications attracts a growing research interest (c.f. Section 2.2), in which railway applications are not yet strongly represented.

In this paper, we address the train rescheduling problem in complex railway networks with mixed infrastructure including single, double, and multiple track railway lines with given planned train paths. We consider shunting as movement of rolling stock between depots and stations followed by rolling stock connections. We apply a new hybrid dispatching algorithm combining classical-quantum modelling and based on quantum annealing. This work extends on a particular linear modeling strategy, partly explored on a toy model in [18]. We develop an integer linear programming (ILP) model, which is solved with proprietary D-Wave solvers as well as with CPLEX for comparison.. The D-Wave approach is analysed in detail, with respect to its applicability and performance in a practically relevant situation. The results suggest that quantum computing and QA in particular, although an early-stage technology, are ready for tackling challenging railway dispatching problems.

The main contributions of the present work are the following:

- A railway traffic dispatching model is introduced for rescheduling trains due to disturbances and disruptions taking into account given timetable, rolling stock connections and a network with single or multi-track segments.

- Quantum annealing-based hybrid heuristics are, for the first time, applied to real-life problems of railway dispatching optimisation on a metropolitan scale.

- The performance of the hybrid (quantum classical) D-Wave solver is demonstrated on a real-life network in Poland and diverse disturbance scenarios of different sizes and types.

- It is found that the D-Wave hybrid solver provides good quality solutions in short time, also for the complex instances in limited computational time.

- It is found that for certain instances (and particular setting of parameter of minimal processing time) the hybrid (quantum classical) D-Wave solver outperforms classical solvers in terms of computational time, yielding still feasible solutions.

The paper is organised as follows. In Section 2 we review state of art literature and identify scientific gaps we intend to fill, in Section 3 we describe problem under investigation, in Section 4 we evaluate our model, in Section 5 we discuss the hybrid solver we use, in Section 6 we present computational results.

2 Literature review

In this section, we present time-indexed models known from the literature, comparing them briefly with our modelling approach. For a general review of railway timetabling and rescheduling we refer to [19, 20]. Second, we review recent quantum annealing applications in optimisation in general, and also in the rail/transport domain. Finally, we summarise the existing scientific gaps.

2.1 Classical railway dispatching

Time-indexed modelling of railway scheduling and rescheduling is quite common in literature for routing [12, 6] and scheduling [21, 5, 7, 22] . Caimi et al. [6], who focus on determining train's path for a complex central railway station area (called the blocking - stairway), used a discrete-time model resulting in a $0-1$ program. Their model is successfully demonstrated on an operational day at the central railway station area Berne, Switzerland. Similarly, Lusby et al. [12] consider train movements in a single major (and thus complex) railway junction, including a freight yard and a few minor stations. They build a mixed integer programming model that can also be solved close to optimal in case of practically relevant instances. They address the train movements on the level of detail of acceleration and deceleration strategies, hence adopting a significantly finer discrete time scale than our one-minute resolution. When compared to our problem, these two models addressed a high traffic in a complex station area. Instead, we are concerned more with a bigger area, multiple stations, and mixed track, but a lower train density and complexity. Our train routes fixed within the stations, and we also model shunting movements that is not a subject of the cited reference.

Caprara et al. [21] proposed a graph theoretic formulation for the problem using a directed multigraph in which nodes correspond to departures/arrivals at a certain station at a given time instant. This formulation is used to derive an integer linear programming model that is solved using a Lagrangian relaxation. Sasso et al. [22] introduced a new pure 0-1 programming formulation, and call it Tick Formulation, to model the Deadlock Detection (DD) problem. Meng and Zhou [7] focused "on the train dispatching problem on an N-track network, with the main challenge of how to formulate specific retiming, reordering, retracking and rerouting options in combination". Their model was demonstrated on generic networks of different structure than what we address. In our situation we are limited to retiming and reordering of trains. Structurally, their model shows similarities to ours, including the role of binary variables and also the introduction of "cells" that generalise the objects the occupancy of which can be controlled: they can be one or more blocks, stations, etc. In our model, the decision variables will be explicitly linked to a subset of stations.

2.2 Quantum annealing and its applications

Quantum annealing [23] is an optimisation method analogous to simulated annealing (SA). Both can be used for the unconstrained minimisation of a quadratic function of ± 1-valued variables, which represents the energy configuration for a set of spins of the Ising model [24, 25]; an important and celebrated problem in physics. The Ising problem can be equivalently rewritten in a form of a Quadratic Unconstrained Binary Optimisation (QUBO) problem with binary $(0-1)$ variables. The Max-Cut problem [26] is also equivalent to the Ising model or QUBO. The problem is NP-hard, has tremendous literature in classical optimisation, and there is significant ongoing progress in the development of algorithms to solve it [27, 28]. An extensive list of Ising problems and their formulations are presented in [29].

Hardware quantum annealers, like the D-Wave machine, implement a quantum version of the Ising model, assigning a quantum bit, i.e. a two-level physical quantum system to each bit. The system employs real physical tow-level quantum systems with a tunable energy operator (Hamiltonian), the energy being the objective function of the optimisation problem. An adiabatic evolution of the quantum system is implemented: the interactions between the spins are slowly changed from an initial Hamiltonian with a simple minimal-energy state to the Hamiltonian corresponding to the objective according to the rules of quantum mechanics. According to the adiabatic theorem of quantum mechanics [30], under certain conditions the physical system remains in its lowest energy state during the evolution, and an optimal, minimum-energy solution can be possibly read out at the end.

An ideal adiabatic quantum computer would be a device with a large number of perfect quantum bits completely decoupled from their environment, very close to zero temperature. Any pair of qubits could be coupled, and even, more general couplings, e.g. involving 3 or more spins could take place. Such a system would exhibit perfect quantum coherence and entanglement. Even in such an ideal system, the time required for the evolution to reach at a minimum energy state with high enough probability depends on the gap between the minimal energy configuration and the one closest to it in energy; the time required is inverse proportional to the gap. Certainly this is not known in advance, it is determined by experimenting in practice. A more severe issue is that the gap can be very small , hence, it is not possible to solve all possible hard problems on such a setup. In spite of that, there are problems in which this approach can be efficient.

Meanwhile, the state-of-the-art physical quantum annealers are smaller systems of a few hundred quantum bits in which not all qubits are coupled, and the arrangement operates at a finite temperature. The fixed topology of qubit couplings means that the problem's graph defined by the nonzero coupling has to be embedded as an induced subgraph of the topology of the system. This embedding is a hard problem itself [31]. It often requires to couple multiple physical qubits to represent a logical bit of the problem. As for the finite temperature, the adiabatic evolution works also in systems that interact with their environment [32]. However, the finite temperature results in noise, and with the larger system size the impact of the noise increases [33].

Operationally the optimiser's point of view quantum annealers can be viewed as probabilistic heuristic solvers returning a statistical sample [34] of configurations which are supposed to be optimal or close to optimal. It must be stressed that the quantum annealers are not algorithms running on digital computers; they are analog devices implemented physically. This latter implies that the coefficients of the problem are encoded with a limited accuracy, and the algorithmic properties of the particular optimisation problem such as its complexity class will not determine the solver's behaviour.

To overcome the problem of limited size and accuracy, quantum annealers of the present state of the art are often used in hybrid (quantum - classical) solvers, orchestrating classical algorithms and using quantum annealing as a subroutine in order to address hard problem instances more efficiently. Solvers available in D-Wave's 'Leap hybrid solver service (HSS)' [35], including the one used in the present work, belong to this family. Meanwhile, quantum technology keeps on developing, systems of bigger size and better topology are regularly announced, they are more and more affordable, and there is a growing community around them.

Currently, the applicability of quantum annealing technologies is being explored. Benchmark problems are solved [36, 37], comparisons with digital computers are made [38], etc. The application of quantum annealers in transportation optimisation is a new area with only a few contributions so far. There is an apparent interest in this research direction in the aviation domain [39, 40, 41], which relates also to the already mentioned benchmarking of annealers [36]. Shipment rerouting was also addressed in the context of quantum annealing [42].

In railway context, the first applications of QA can be found in train dispatching [43, 18] and rolling stock planning [44]. In particular, [13] was the first proof-of-concept demonstration of a (pure) quantum computing approach to railway dispatching. A followup paper [18] laid down the principles of a more general modeling approach to railway rescheduling in light of QA, introducing a suitable QUBO / HOBO (higher-order binary optimisation) encoding of these problems.

Finally, there are a few examples of hybrid solvers applications including traffic flow optimisation [45], multi-car paint shop optimisation [46], tail assignment problem [47], and vehicle optimisation [48].

2.3 Scientific gaps

We recognise several important scientific gaps. First, existing optimisation models suffer from the curse of dimensionality, and inability to solve larger real-life instances. Second, current QA models [43, 18] represent pure QA implementations and demonstrated only on simple network setups. Third, no hybrid QA-based models have been used for real-life railway, or even other schedule-based modes, like public transport and air traffic planning and/or rescheduling.

In this paper we demonstrate the quantum readiness of medium-scale railway dispatching models: we successfully apply quantum methods in the dispatching problem of a metropolitan-scale real-life railway network. To do so, we use a hybrid approach combining quantum and classical computing. For a fair comparison of current and future QA approaches with state-of-the-art classical approaches, we elaborate the railway model readily suitable for both classical and hybrid (quantum-classical) solvers in a comparable manner. Quantum computing is believed to develop rapidly in near future. Hence, demonstration quantum readiness for railway problems right now is step forward towards the demonstration of quantum advantage in future. On future quantum devices the comparison with classical solvers is expected to be in strong favour for the quantum ones. In this way we also contribute in developing a novel general set of railway dispatching models that has a potential to scale well with the size of the problem, aiming to overcome the curse of dimensionality.

3 Problem description

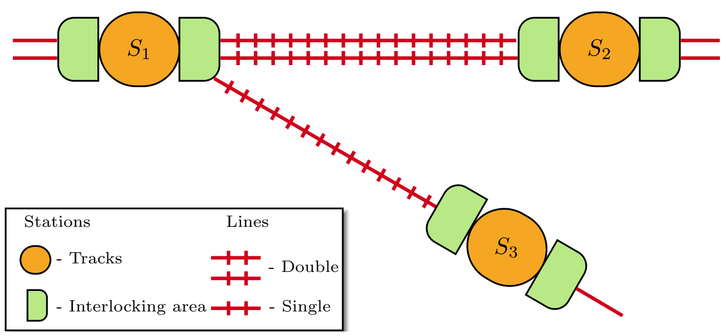

Figure 1: Exemplary network. Interlocking areas of stations in green.

We consider a railway dispatching problem that includes train rescheduling, retiming, and shunting movements with rolling stock circulation at stations. Also, we consider an urban railway network with mixed tracks from single, double, up to quadruple tracks.

We model a railway network with edges and nodes. Figure 1 depicts an exemplary network layout composed of nodes (station or junction) and edges (single, double, multiple track lines). Each edge is composed of one or more *tracks*. Each track consists of blocks sections defined by pairs of signals [12]. Each block section can be occupied by at most one train at a time. A subsequent train is allowed to enter the block only after a minimal *headway*: minimal time span between the trains. We assume a 2-block signalling system, meaning that two free blocks are required between the consecutive trains. We also assume green way policy (trains have the free way to move at the maximal allowed speed between stations) and constant running times.

Each node is composed of station tracks (blocks) and interlocking areas. One station track can be occupied by one train at a time. Routing dependencies between pairs of trains competing for the same resource in interlocking areas are considered in order to to guarantee that only one train from the pair can occupy the area at the time.

A train's *route* is the sequence of blocks the train passes during its journey. A train *path* is a sequence of arrival and departure times of a particular train assigned to a train route. We assume a one-minute resolution for all time parameters such as timetable time, running and dwell time minimal headway time or passing time, resulting in integer variables. This assumption of discretised time is needed to enable the use of quantum-based solvers, and, more importantly, introduces the possibility of pruning the inherently binary variables. The relaxation of integer constraints on time variables is not expected to improve computation time significantly, as the real difficulty is tied to precedence of trains, encoded later on precedence binary variables.

In the network, two trains can follow each other, i.e. keep the given order, meet and pass (M-P) when going the opposite directions, and meet and overtake (M-O), i.e. change the order, when going the same direction. Single tracks are designed for bidirectional traffic, double-tracks for unidirectional, one for each direction, and *multi-track* lines can be combination of unidirectional and bidirectional. On *single-track* lines, M-Ps and M-Os are only possible at stations. To prevent M-P on single-track lines, we determine the set $\mathcal{J}^2_{\text{single}}$ as the set of all pairs of trains that can potentially meet on the single track line heading in opposite directions.

Trains in the same direction preserve their order between stations, and keep the minimal headway time between each other. To prescribe these we determine set $\mathcal{J}^2_{\text{headway}}$: the set of all pairs of trains that can potentially violate the minimal headway condition.

The train traffic is scheduled on the basis of the given timetable. The timetable contains all the train paths. In the train dispatching problem, we are given an initial timetable that is conflict-free. However, conflicts may appear due to disturbances such as late departures and/or arrivals due to excessive passenger demand, malfunctioning rolling stock, etc. Following [49], a *conflict* is an inadmissible situation when at least a pair of trains claim the same resource (e.g., block section, switch) simultaneously. We assume that the conflict may occur either on the railway line or at the station. Hence, the conflicts have to be solved by modifying the original timetable, applying decisions on the train sequencing and retiming for trains claiming for the same resource.

Let us denote by $\hat{S}$ the set of all stations. As an initial step of our approach, we determine a subset $S \subset \hat{S}$, the *decision stations*, and assume that the direct decisions implied by our model affect these stations only. As decision stations we select those stations, where routes of trains intersect, where trains start or terminate, or where selected part network is bounded. The motivation is that if the routes of trains are fixed, the decisions on modifying their train paths has to be made with respect to these stations. (If we change some of the trains' routes in a re-routing process, new model with new parameters if developed, e.g. additional decision stations may appear.) In what follows, by a station we always mean a decision station. This also means that non-decision stations appear only through the parameter values in the model, they do not appear as indices of decision variables or parameters. Headways, for instance, are calculated between decision stations, taking into account all the line blocks and station blocks of non-decision stations in between.

On the station where a train terminates or sets off, shunting is also modeled. The goal of shunting is to move the train from the passengers' service track to the depot or vice versa. The depot is treated as the station, and consider it as a black box, without detailed layout. We treat shunting movements as service train from depot to the starting station of the service train, or from the terminating station to the depot. The rolling stock circulation condition is applied to ensure the precedence between the service train and the actual train.

4 Methods and Model

In the following we describe our model in detail. Section 4.1 defines sets, parameters and decision variables. Section 4.2 describes our integer linear programming (ILP) formulation.

4.1 Sets, parameters and decisions

To formulate our decision variables, constraints and the objective function, we determine sets of index tuples needed to find the actual index sets of variables, from the given infrastructure data, timetable data, and the rolling stock circulation plan. Also, we introduce parameters calculated from the same input.

4.1.1 Sets

The main objects of our model for railway dispatching are the trains $j \in \mathcal{J}$ and the stations $s \in \mathcal{S}_j$ in their path. Set $\mathcal{S}_j$ includes decision stations only and it is an ordered set. In addition to $\mathcal{J}$ and $\mathcal{S}_j$, the relevant sets are the following. Set $\mathcal{J}_s^{2\,(\text{turn})} \subset \mathcal{J} \times \mathcal{J}$ $(\forall s \in \mathcal{S})$ is the set of all pairs of trains so that the first train of the pair terminates at station s and its rolling stock continues as the second train of the pair. This set is deduced from the rolling stock circulation plan and the timetable. Set $\mathcal{J}^{2\,(\text{close})} \subset \mathcal{J} \times \mathcal{J}$ is the set of trains which are close enough to each other in time so that precedence variables have to be defined for them. More details will be given at the description of the parameter $d_{\max}$ which this set depends on. Set $\mathcal{J}^{2\,(\text{headway})} \subset \mathcal{J}^{2\,(\text{close})}$ is the set of train pairs that can potentially violate the minimal headway time between two subsequent trains on a line segment. For a pair of trains to be in $\mathcal{J}^{2\,(\text{headway})}$, both their routes need to include the same line segment so that the trains are moving in the same direction on it and can meet there according to model parameters, i.e. maximal allowed delay. Set $\mathcal{J}^{2\,(\text{single})} \subset \mathcal{J}^{2\,(\text{close})}$ is the set of all trains that share at least one single-track line segment as a common part of their routes so that they are heading in the opposite direction. Set $(\forall s \in \mathcal{S})$ $\mathcal{J}_s^{2\,(\text{track})} \subset \mathcal{J}^{2\,(\text{close})}$ is the set of train pairs that are planned to occupy the same track on station s anytime during the planning horizon, thereby competing for the same track. Set $(\forall s \in \mathcal{S})$ $\mathcal{J}_s^{2\,(\text{switch,out})} \subset \mathcal{J}^{2\,(\text{close})}$ is the set of train pairs that are planned to pass the same interlocking area of s upon their departure from station s. Set $\left(\forall (s, s') \in \mathcal{S}^{\times 2}\right)$ $\mathcal{J}_{s,s'}^{2\,(\text{switch,out,in})} \subset \mathcal{J}^{2\,(\text{close})}$ is the set of train pairs that are planned to pass the same interlocking area of s in order to have j to departure s while j' arrive s from the direction of s'. Set $\left(\forall s \in \mathcal{S}^{\times 2}\right)$ $\mathcal{J}_{s,s'}^{2\,(\text{switch,in,noMP})} \subset \mathcal{J}^{2\,(\text{close})}$ is the set of train pairs that are planned to pass the same interlocking area of s upon their arrival at s from the direction of s', and there is no M-P possibility for them between s and s' Set $\left(\forall s \in \mathcal{S}^{\times 2}\right)$ $\mathcal{J}_{s,s'}^{2\,(\text{switch,in,MP})} \subset \mathcal{J}^{2\,(\text{close})}$ is the set of train pairs that are planned to pass the same interlocking area of s upon their arrival at s so that either they both come from the direction of s' but there is a M-P possibility for them between s and s', or one of them is approaching s from a direction other than s'. Set $(\forall j \in \mathcal{J})$ $\mathcal{C}_j^2$ is the set of all station pairs that are subsequent in the route of j. Set $\left(\forall (j, j') \in \mathcal{J}^{\times 2}\right)$ $\mathcal{C}_{j,j'}^{2\,(\text{common})}$ is the set of all station pairs that appear as subsequent stations in the route of both j and j' heading in the same direction. Set $\left(\forall (j, j') \in \mathcal{J}^{\times 2}\right)$ $\mathcal{C}_{j,j'}^{2\,(\text{common, single})}$ is the set of all station pairs that appear as subsequent stations in the route of both j and j' *which are connected with a single-track line segment*, and trains are heading in opposite directions. The order within the pairs is determined by j. All these sets can be enumerated on the basis of the input data in a straightforward manner.

4.1.2 Parameters

The parameters that appear in our model are the following:

- $\tau^{(\text{pass})}(j, s \to s')$ is the running time of j from s to s'.
- $\tau^{(\text{headway})}(j, j', s \to s')$ is the minimal headway time for j' following j from s to s'.
- $\tau^{(\text{switch})}(j, j', s)$ is the running time of j over a interlocking area of s, where j may be in conflict with j'. (In our examples we do not consider cases when, e.g., on bigger stations there are multiple switches with different technological times. The model could trivially be extended to cover such scenarios by adding extra indices if relevant.)
- $\tau^{(\text{dwell})}(s, j)$ is the minimal dwell time of j at s.
- $\tau^{(\text{turn})}(s, j, j')$ is the minimum turnaround time for the rolling stock of a train j terminating at s to continue its journey as train j'.
- $\sigma(j, s)$ is the scheduled departure time of s from j, given by the original timetable.
- $\upsilon(j, s)$ is the earliest possible departure time of j from s when a disturbance and/or disruption in the network happens. It is the maximum of $\sigma(j, s)$ and the technically feasible earliest departure time, and may be more constraining than the first depending on the initial delays. The latter is calculated assuming that the train follows its planned route at minimum running time, and that there are no other trains present on the network.
- $d_{\max}$ is an upper bound assumed for the secondary delay $t^{(\text{out})}(j, s) - \upsilon(j, s)$. This parameter sets an upper limit for the possible secondary delays that can arise on the particular part of the network. Setting such a bound is common in the literature, see e.g. [11]. We set the same value of this parameter for all our testing instances. It has to be big enough so that no secondary delay exceeds it (e.g. $d_{\max} = 40$ excludes the possibility of a one-hour delay due to waiting for another delayed train). Meanwhile it is desirable to set its value as small as possible; restricting the time variables to small intervals decreases the number of binary decision variables in the mode.

A small number of binary decision variables has benefits when using a solver based on quantum annealing: the current NISQ devices are of limited size, and their noise significantly increases with the problem size [33]. Unnecessary variables would generate additional constraints, which, in addition to the increase in size, also increase the connectivity of the graphs which can lead to embedding problems in quantum systems. Therefore, even when using hybrid (classical + quantum) algorithms, maintaining a small model is desirable. In our case, the reduction of the model size by the choice of a small $d_{\max}$ parameter is achieved via defining the set $J^{2\,(\text{close})}$. A pair of trains (j, j') is included in this set if and only if they can meet on any station, given the original timetable, the disturbed/disrupted timetable, and assuming that no train can have a secondary delay greater than $d_{\max}$. The number of constraints will be proportional to the average number of trains that can meet another train at a station; this will be linear in $d_{\max}$.

4.1.3 Decision variables

Our decision variables are the following. First, we use departure time variables:

$$t^{(\text{out})}(j, s) \in \mathbb{N} \tag{1}$$

defining the departure time of train $j \in \mathcal{J}$ from station $s \in \mathcal{S}_j$. Such a variable is defined for all decision stations $\mathcal{S}_j$. In the case of trains that terminate within the modeled part of the network, the last station is not included in $\mathcal{S}_j$. The arrival time of the trains at stations, $t^{(\text{in})}(j, s) \in \mathbb{N}$ are trivially related to the $t^{(\text{out})}(j, s)$ variables through a constant offset (c.f. Eq. (3)), given the fixed running time assumption.

Second, in addition to the time variables, we use three sets of binary precedence variables. Decision variables $y^{(\text{out})}(j, j', s) \in \{0, 1\}$ determine the order of trains to leave stations: the variable takes the value 1 if train j leaves station s before train j', and 0 otherwise. The (j, j', s) tuples for which an y variable is defined will be specified later. Similarly, the precedence variables $y^{(\text{in})}(j, j', s) \in \{0, 1\}$ prescribe the order on the entry to stations; the value is 1 if j arrives to s before j'. The last set of precedence variables describes the precedence of trains at some resource located between stations s and s' (e.g. single track used by j and j' heading in opposite direction). The binary variable $z(j, j', s, s') \in \{0, 1\}$ will be 1 if j uses the given resource before j'. Also in this case, the quadruples (j, j', s, s') for which we have such a variable will be specified later.

4.2 ILP formulation

Given the index sets, variables, and the parameters and decision variables of the model, now we formulate the optimisation objective and the constraints.

4.2.1 Objective function

As our model is restricted to a metropolitan-scale part of the whole railway network, our goal is to minimise the secondary delays generated within this part. Hence, a suitable objective function is the *weighted sum of secondary delays at the destination station*:

$$f(t) = \frac{1}{d_{\max}} \sum_{j \in \mathcal{J}} w(j) \left(t^{(\text{out})}(j, s^*) - \upsilon(j, s^*) \right). \tag{2}$$

where s^* is the last element of S_j, and w_j are weights for each train representing its priority. The constant multiplier $1/d_{\max}$ is optional; we use it for better comparability of different instances.

4.2.2 Constraints

The constraints of the model are the following.

Minimal running time Each train needs a minimal time to get to the subsequent station:

$$\forall j \in \mathcal{J} \ \forall (s, s') \in \mathcal{C}_j \quad t^{(\text{in})}(j, s') = t^{(\text{out})}(j, s) + \tau^{(\text{pass})}(j, s \to s') \tag{3}$$

Headways A minimal headway time is required between subsequent train pairs on the common part of their route as

$$\begin{aligned} &\forall (j, j') \in \mathcal{J}^{2\,(\text{headway})} \ \forall (s, s') \in \mathcal{C}^{2\,(\text{common})}_{j,j'} t^{(\text{out})}(j', s) \geq \\ &t^{(\text{out})}(j, s) + \tau^{(\text{headway})}(j, j', s \to s') - C \cdot y^{(\text{out})}(j', j, s), \end{aligned} \tag{4}$$

where C is a constant big enough to make the constraint satisfied whenever the binary variable $y^{(\text{out})}(j', j, s)$ takes the value of 1. In our implementation we calculate and use the smallest suitable value of C given particular $d_{\max}$ value, i.e.

$$C = -\min(t^{(\text{out})}(j', s)) + \max(t^{(\text{out})}(j, s)) + \tau^{(\text{headway})}(j, j', s \to s') \tag{5}$$

Single-track occupancy Trains moving in opposite directions cannot meet on the same single-track line segment:

$$\begin{aligned} &\forall (j, j') \in \mathcal{J}^{2\,(\text{single})} \ \forall (s, s') \in \mathcal{C}^{2\,(\text{common, single})}_{j,j'} \\ &t^{(\text{out})}(j', s') \geq t^{(\text{in})}(j, s') - C \cdot z(j', j, s', s), \end{aligned} \tag{6}$$

where C is a big enough constant chosen similarly to that in Eq. (4).

Minimal dwell time Each train has to occupy the station node for a prescribed time duration at each station:

$$\forall j \in \mathcal{J} \forall s \in S_j \quad t^{(\mathrm{out})}(j,s) \geq t^{(\mathrm{in})}(j,s) + \tau^{(\mathrm{dwell})}(j,s). \tag{7}$$

Timetable No train is allowed to depart before its scheduled departure time:

$$\forall j \in \mathcal{J} \forall s \in S_j \quad t^{(\mathrm{out})}(j,s) \geq \sigma(j,s). \tag{8}$$

Station track occupancy Station tracks can be occupied by at most one train at a time:

$$\forall s \in S \forall (j,j') \in \mathcal{J}_s^{2\,(\mathrm{track})} t^{(\mathrm{in})}(j',s) \geq t^{(\mathrm{out})}(j,s) - C \cdot y^{(\mathrm{out})}(j',j,s), \tag{9}$$

where C is chosen similarly to that in Eq. (4) again. Note that this requirement may not be needed for depot tracks; the exceptions can be handled by the proper definition of $\mathcal{J}_s^{2\,(\mathrm{track})}$.

Interlocking area occupancy These ensure that trains cannot meet in interlocking area:

$$\begin{aligned}
\forall s \in S \forall (j,j') \in \mathcal{J}_s^{2\,(\mathrm{switch,out})} t^{(\mathrm{out})}(j',s) \geq \\
t^{(\mathrm{out})}(j,s) + \tau^{(\mathrm{switch})}(j,j',s) - C \cdot y^{(\mathrm{out})}(j',j,s), \quad (10) \\
\forall (s,s') \in S^{\times 2} \forall (j,j') \in \mathcal{J}_{s,s'}^{2\,(\mathrm{switch,out,in})} t^{(\mathrm{in})}(j',s) \geq \\
t^{(\mathrm{out})}(j,s) + \tau^{(\mathrm{switch})}(j,j',s) - C \cdot z(j',j,s',s), \quad (11) \\
\forall (s,s') \in S^{\times 2} \forall (j,j') \in \mathcal{J}_{s,s'}^{2\,(\mathrm{switch,in,noMP})} t^{(\mathrm{in})}(j',s) \geq \\
t^{(\mathrm{in})}(j,s) + \tau^{(\mathrm{switch})}(j,j',s) - C \cdot y^{(\mathrm{out})}(j',j,s'), \quad (12) \\
\forall (s,s') \in S^{\times 2} \forall (j,j') \in \mathcal{J}_{s,s'}^{2\,(\mathrm{switch,in,MP})} t^{(\mathrm{in})}(j',s) \geq \\
t^{(\mathrm{in})}(j,s) + \tau^{(\mathrm{switch})}(j,j',s) - C \cdot y^{(\mathrm{in})}(j',j,s), \quad (13)
\end{aligned}$$

with a choice of C similar again to that in Eq. (4).

Rolling stock circulation constraints are used to bind the train with the shunting movement called also service train. If the train set of train j which terminates at s is supposed to continue its trip as (service train) j' or vice versa, a precedence of these trains including a minimum turnaround time has to be ensured:

$$\forall s \in S \forall (j,j') \in \mathcal{J}_s^{2\,(\mathrm{turn})} t^{(\mathrm{out})}(j',s) \geq t^{(\mathrm{in})}(j,s) + \tau^{(\mathrm{turn})}(j,j',s). \tag{14}$$

Order of trains We have additional conditions on the y-variables, concerning the case when M-O is not possible on the line or station.

$$
\begin{aligned}
\forall (j,j') \in \mathcal{J}^{2\,(\text{headway})}\ \forall (s,s') \in \mathcal{C}^{2\,(\text{common})}_{j,j'} \\
\forall (j,j') \in \mathcal{J}^{2\,(\text{headway})} \cap \mathcal{J}^{2\,(\text{track})}_{s'} \\
y^{(out)}(j,j',s) = y^{(out)}(j,j',s') \\
\forall (s,s') \in S^{\times 2} \forall (j,j') \in \mathcal{J}^{2\,(\text{switch,in,MP})}_{s,s'} \cap \mathcal{J}^{2\,(\text{track})}_{s'} \\
y^{(in)}(j,j',s') = y^{(out)}(j,j',s')
\end{aligned}
\tag{15}
$$

The objective function in Eq. (2), together with the constraints in Eq. (3)-(14) define our mathematical programming model.

The model yields an integer linear program, and the time variables can be constrained even into a finite range using the parameter $d_{\max}$. Note that the binary variables have the obvious symmetry property

$$
\begin{aligned}
y^{(\text{out})}(j',j,s) &= 1 - y^{(\text{out})}(j,j',s) \\
z(j',j,s',s) &= 1 - z(j,j',s,s'),
\end{aligned}
\tag{16}
$$

reducing number of variables to use independent binary variables only.

As for the scaling of our model, the number of time variables i.e. $\#t$ is bounded by number of trains times number of stations $\#\mathcal{J}\#\mathcal{S}$ (see also Section 3.1 of [18]). To estimate more precisely number of these variables, we observe that in our model trains does not visits all stations. We assume that in average each train visits $\alpha \in (0,1)$ fraction of the stations, we have then

$$
\#(t) \approx \alpha \#\mathcal{J}\#\mathcal{S}. \tag{17}
$$

as a tighter bound, see Tabs.1 2.

Suppose, that most of the network is composed mainly of double track lines. Then there will be typically a single precedence variable per train and station ($y^{(out)}(j,j',s)$), and thus their number can be estimated by:

$$
\#y + \#z \approx \#y \approx \alpha N(d_{\max}) \#\mathcal{J}\#\mathcal{S}, \tag{18}
$$

where $N(d_{\max})$ is number of trains the train can meet at a station on average. Under such assumptions the total number of variables grows linearly with number of trains and stations:

$$
\#vars = \#t + \#y + \#z \approx \text{const}\#\mathcal{J}\#\mathcal{S}. \tag{19}
$$

If, on the other hand, the network has dominantly single track lines, two precedence variables per train and station ($y^{(out)}(j,j',s)$ and $z(j,j',s,s')$) will be needed, and thus

$$
\#y = \#z \approx \alpha N(d_{\max}) \#\mathcal{J}\#\mathcal{S}. \tag{20}
$$

In both cases we assume that there are not many $(y^{(in)}(j, j', s))$ variables, as they are tied certain particular interlocking area conditions.

Based on Eq. (16) there are typically 2 constraints per each y variable as headway and station track occupation constraints, and 2 constraints per each z variable as single track occupation constraints. For the interlocking area occupation constraints it is more complicated as it involves both z and y variables; a good assumption is to consider 2 constraints for each y variable. For the running time and minimal dwell time we expect one constraint per train and station. Thus number of constraints can be estimated by

$$\#constr. \approx 6\#y + 2\#z + 2\#\mathcal{J}\#\mathcal{S} \tag{21}$$

where the second term is tied to single track line conditions (one condition per z variable).

$$\#constr. \approx \begin{cases} \alpha(6N(d_{\max}) + 2)\#\mathcal{J}\#\mathcal{S} & \text{for double track} \\ \alpha(8N(d_{\max}) + 2)\#\mathcal{J}\#\mathcal{S} & \text{for single track.} \end{cases} \tag{22}$$

Henceforth, because of the limitations in time imposed by $d_{\max}$ the model can be viewed as linear in number of trains and number of stations. Hence, as expected, the model is more complex for the network in which single track lines dominate. Although, analysed network is mostly composed of the double track lines, we will analyse certain use-cases with the fraction of single track lines increased.

Observe that without limiting the secondary delays by $d_{\max}$, the value of $N(d_{\max})$ will only be bounded by $\#\mathcal{J} - 1$ (as each train is considered to possibly meet each other), which makes the scaling of the number of precedence variables and number of constraints quadratic in the number of trains. This would increase the model size which we want to avoid.

5 Hybrid quantum-based approach

To overcome the limitations described in in Section II of current hardware quantum annealers, we apply hybrid quantum-based solver. In particular, we use the 'Leap hybrid solver service (HSS)' [35]; a cloud-based proprietary solver which is developed by the market leader of quantum annealing hardware and is available as a service. It utilises a hybrid approach that combines classical computational power with quantum processing.

In particular, we used Constrained Quadratic Model (CQM)[50] Solver. The CQM accepts constrained problem in its input, which it is handling internally using penalty methods. Hence, we submit the constraints and the solver's internal preprocessing mechanism adds them to the model automatically.

After the preprocessing, which includes subproblem identification and decomposition to smaller instances [51], the hybrid solver implement a workflow that combines a portfolio of various classical heuristics (including tabu search, simulated annealing, etc.) running on powerful CPUs and GPUs. In the course of the solution process the hardware quantum annealer is invoked. This inputs a smaller quadratic unconstrained binary optimisation (QUBO) problem or problems, computed similarly to state-of-the-art transformation method [52], that is general for ILP problems, performs the required embedding, runs the QUBO subproblem on the physical annealing, and returns a sample of potential solutions. After these readouts, the sample is incorporated into the solution workflow. In this way, even though the physical annealer supports problems with limited size (e.g. the Pegasus machine has 5500 physical qubits [53], and several of these may be needed to represent a binary variable because of the embedding), the approximate solution of small subproblems can boost the classical heuristics. The workflow results in a solution that is not guaranteed to be optimal, so it can be considered as a heuristic solver.

While D-Wave's proprietary solvers hide the exact details of the described process process from the user, the output is supplemented with timing parameters, including *run time*: the total elapsed time including system overhead, and *QPU access time* which is is the time spent accessing the actual quantum hardware. These parameters enable a comparison with other solvers.

In addition to the actual problem input, the CQM solver has an optional input parameter, `t_min`, which is a time limit for the heuristics running in parallel[50]. Each thread is stopped at the actual best solution if this time has been reached. We have sampled the CQM problems with various settings of `t_min` and uncovered the impact of this setting to the model performance and solution quality for our problem.

The CQM Solver can solve problems encoded in the form of Eq. (2) – (15) (c.f. Section 4.2) with up to 5000 binary or integer variables and $100,000$ constraints. Computational results of railway dispatching problems were obtained using classical as well as hybrid quantum-classical solvers. As a classical solver, IBM ILOG CPLEX Interactive Optimizer (version 22.1.0.0) was used. The CPLEX computation was performed on 16 cores of *Intel(R) Core(TM) i7-10700kF CPU 3.80GHz* with 64GBs of memory.

6 Computational results

In this section, we demonstrate the performance of the discussed hybrid quantum-based approach on a network at Polish railways - central part of the Upper Silesia Metropolis. The purpose of experiments is compare the hybrid approach with the state of art commercial solver, CPLEX. This is performed for various experiment settings: single track line, double track line, network, disturbances from outside the network, disturbances within the network (e.g. closures). As the objective is tied to secondary delays, we can roughly apply it to asses the degree of difficulty of the problem.

Section 6.1 describes the considered network and traffic characteristics. Section 6.2 presents the results of comparison between CPLEX and hybrid quantum-classical CQM solver on generic examples on the core line of the network, this is performed to compare double track line scenarios with single track line scenarios and closure scenarios. Section 6.3 presents analogous results for larger network with real live timetable. Here the case studies concerns various scenarios with different degree of difficulty.

6.1 Considered network

We use the Open Railway Map[2] representation of the selected part of the Polish railway network located in the central part of the Metropolis GZM (Poland), presented in Fig 2. It comprises 25 nodes including 11 stations and 3 branch junctions, 146 blocks, and 2 depots. The infrastructure is managed by Polish state infrastructure manager PKP PLK; for further details of the traffic management consult [54].

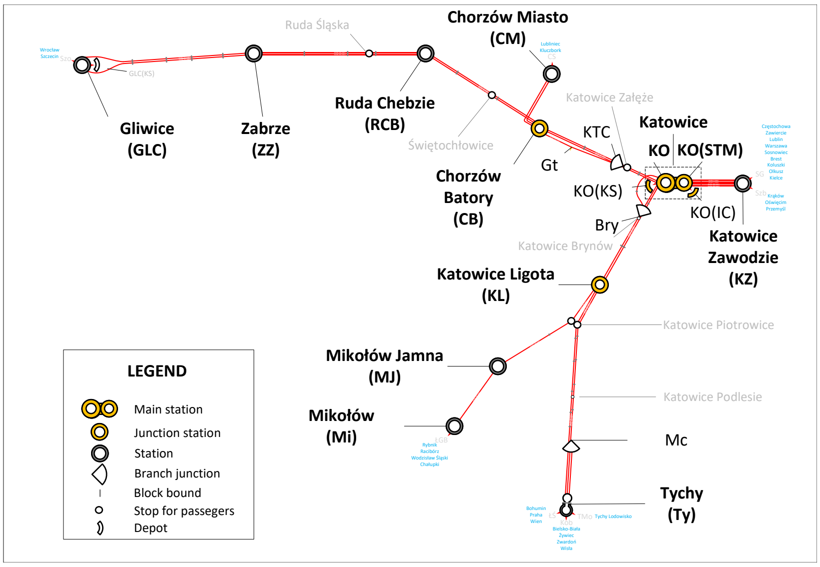

Figure 2: The network under consideration.

The decision stations include junction stations {KO, KO(STM), CB, KL}, stations bounding the analysed part of the network {GLC, CM, KZ, Ty, Mi}, depots {KO(KS), KO(IC)} from which regional and intercity trains

[2] https://www.openrailwaymap.org/, visited 2022-11-25

are being shunted to and from {KO}, and stations on the single track line – {MJ} to allow meet and pass (M-P) or meet and overtake (M-O) there. As described before, the decision stations are the only ones that will appear explicitly in our model. Due to operational reasons resulting from track design, the decisions on train departure times (and thus on trains' order) are, in the current operational practice, *de facto* made with respect to the decision stations.

All other stations and station-like objects are treated as line blocks as long as no rerouting or retracking is considered; they will appear in the model implicitly via parameters. These stations include:

- stations $\{\mathrm{ZZ}, \mathrm{CB}\}$ in which usually there are rigid assignments of the platform tracks to the traffic direction (i.e., track 1 towards {GLC}; track 2 towards {KO})
- branch junctions $\{\mathrm{KTC}, \mathrm{Bry}, \mathrm{Mc}\}$ in which usually there is a rigid assignment to the tracks.

As an example, let for a particular train j, $\hat{\mathcal{S}}_j$ = {KZ, KO(STM), KO, KTC, CB, RCB, ZZ, GLC} be the ordered set of stations. Then, the dispatching decisions *de facto* will be made in above mentioned *decision stations* $\mathcal{S}_j$ = {KZ, KO(STM), KO, CB, GLC}. If such approach is not satisfactory, following [18], we will retrack trains not only at the *decision stations* $\mathcal{S}_j$, but also beyond.

In all computation priority weights in Eq. (2) we use $w_j = 1$ for stopping (local) trains $w_j = 1.5$ for intercity (fast) trains $w_j = 1.75$ for express trains, and $w_j = 0$ for shunting (is applicable).

6.2 Generic examples on a selected railway line

As the first set of experiments we address generic instances on a part of the network in Fig 2; line KO-GLC in particular, which is double track line. The goal with presenting this scenarios is to compare the solvers for the double track line (case 1), the double track line with closures (case 2), and single track line (case 3). Shunting movement, and rolling stock circulation are not considered in this set of experiments.

1. Case 1, the double track line with dense traffic. We use cyclic 3-hour timetable with 10 trains each hour and each direction, i.e. we have 59 trains. (The decision stations are KO, CB and GLC).

2. Case 2, similar to case 1, but with the additional disturbance of one of the tracks between ZZ and RCB being closed, what is not included in the timetable. We use a cyclic 2-hour timetable with 10 trains each hour and each direction, i.e. we have 40 trains. (The decision stations are KO, CB, RCB, ZZ, and GLC and timetable is not feasible due to the closure).

c.	$\#\mathcal{S}$	$\#\mathcal{J}$	# int vars		# precedence vars		# constraints	
			act. mean	est. $\approx$	act. mean	est. $\approx$ doub./sing.	act. mean	est. $\approx$ doub./sing.
1	3	59	118	118	1538	1415 / -	7889	8731 / -
2	5	40	142	133	1394	1600/3200	8491	6867/13067
3	5	21	79	70	592	- / 840	3057	- / 3499

Table 1: Estimated and real number of variables and constraints (we use $\alpha = 2/3$ see Eq. (17))

3. Case 3, the whole line is considered as a single track, with a feasible timetable of 3 hours traffic and 21 trains (decision stations are KO, CB, RCB, ZZ, and GLC).

For each of the cases we compute 12 instances, each but first with different initial delays of trains subsets (instance 0 yields no initial delays and is feasible in cases 1 and 3). We use the parameter value $d_{\max} = 40$ min, in all cases.

To estimate the sizes of the problem and compare with real values (act. mean) we expect each train to be in possible interaction with $N(d_{\max}) = 12$, for case 1 and 2, and $N(d_{\max}) = 6$ for case 3 (less dense traffic to start from the feasible timetable on the single track line). Given this, the estimated and real numbers of variables are presented in Tab 1.

The computational results are presented in Figs. 3, 4, and 5 for cases 1, 2 and 3 respectively.

Case 2, namely the double track line with closures and dense traffic, has appeared to be most challenging for both the classical solvers (in terms of computational time) and for the quantum solver (in terms of objective). Nevertheless, on this example, there are instances, where the current CQM D-Wave hybrid solver outperforms CPLEX in terms of computational time. In the next subsection, we will test the CQM solver on more realistic scenarios of the railway traffic on presented network. We will also focus on the track closure situation (change double track line into the single track one under dense traffic) to elaborate our findings from case 2.

6.3 Results for real railway situations

Our actual computations address various use cases of train delays outside the network and within the network, numbered from 1 to 9. With the increase of the number of the case the level of difficulty of the problem is expected to increase (e.g., more trains ore involved or disturbances are spread more over the network) - it is approximately reflected in the increasing objective values of the optimal CPLEX solution, see Tab. 3. In details, cases 1 - 3 concern only delayed trains with no closures.

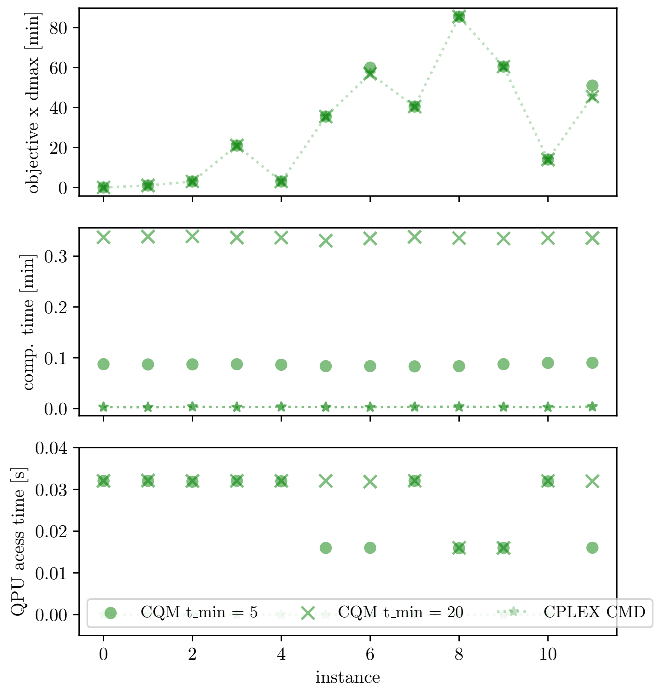

Figure 3: Comparison of the performance of classical solver (CPLEX) with that of the hybrid CQM for case 1. All the displayed solutions are feasible. Total computational time (middle panel) and QPU times (lower panel) were provided by the D-Wave output.

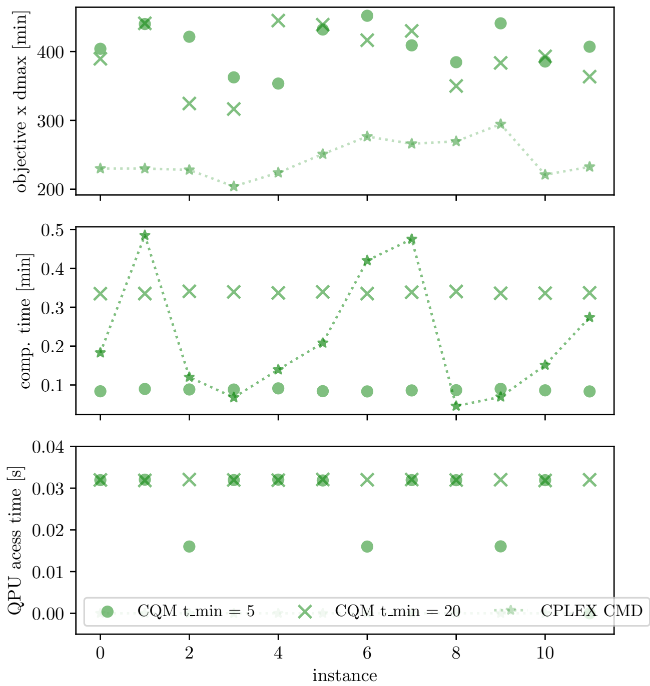

Figure 4: Comparison of the performance of classical solver (CPLEX) with that of the hybrid CQM for case 2. Note that the CQM is fast, even when compared to CPLEX. The found solutions are suboptimal but can be useful. Observe that all solutions were feasible, and the increment of `t_min` parameter in most cases boost the quality of solution, obviously at the cost of computational time.

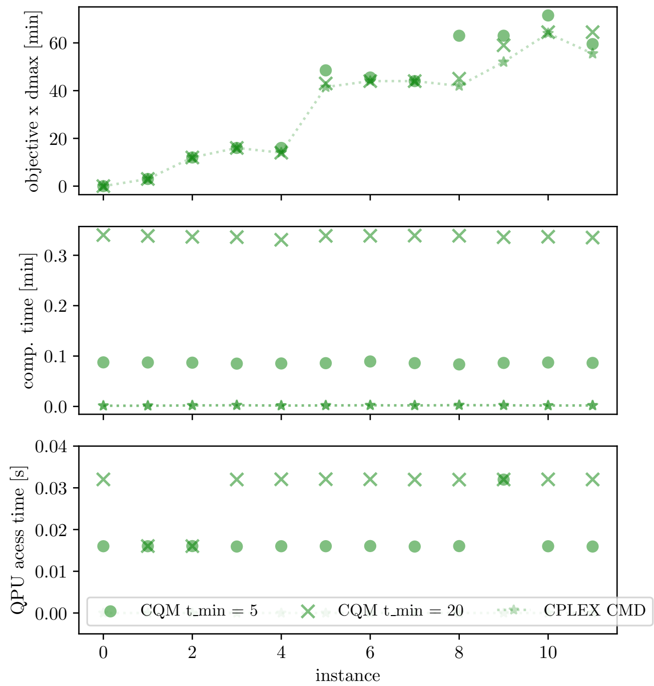

Figure 5: Comparison of the performance of classical solvers (CPLEX) with that of the hybrid CQM for case 3. All the displayed solutions are feasible.

Cases with number 4 or higher concern also rerouting of trains, due to closures. Case 4 and 5 concerns rerouting trains from double track line KTC-CB to the single track line with higher passing times, see Fig. 2 (trains have no initial delays in case 3 and some initial delays in case 4). Case 6 concerns multiple closures, i.e. change of multiple line KZ-KO(STM) and double track line KO-KL to single track lines. (Fig. 2, the justification may come from upcoming reconstruction works on this part of network). Cases 7 -9 have closures of both case 5 and case 6, but with different initial delays of trains. The intention of last 3 cases is to create a really challenging dispatching problem. For comparison, case 0 is the default problem with no disturbances.

The general rough estimated numbers of variables (from Eq. (17) - Eq. (19)) and constraints (Eq. (22)) for these cases, given $\#\mathcal{S} = 10$ (# decision stations without depots), $\#\mathcal{J} = 27$ $N(d_{\max}) = 4$, and $\alpha = 0.5$, are presented in Tab. 2. The increasing number of closures, the bigger portion of single-track parts in the network results in an increase in the number of constraints and variables.

Results of optimisation with CPLEX and CQM hybrid solver are presented in Tab. 3. Observe that the CQM solver always returns a feasible

	network of single track lines	network of double track lines
# vars	1215	675
#constr.	4590	3510

Table 2: Estimated number of constraints and variables.

case	CPLEX			CQM hyb. `t_min` = 5 s mean value over 5 realisations		
	#vars / # constr.	obj. × dmax	comp. time [s]	obj. × dmax	comp. time [s]	QPU time [s]
0	556 /1756	0.0	0.072	0.0	5.19	0.026
1	556 / 1756	1.0	0.076	1.0	5.24	0.022
2	556 / 1740	6.0	0.079	6.0	5.37	0.032
3	556 / 1769	7.5	0.085	7.5	5.05	0.032
4	662 / 2210	78.25	0.196	82.70	5.10	0.026
5	662 / 2204	114.75	0.25	132.55	5.25	0.022
6	711 / 2599	91.25	0.41	142.3	5.14	0.026
7	817 / 3029	188.75	7.98	263.4	5.12	0.019
8	817 / 3074	157.75	3.70	271.65	5.12	0.022
9	817 / 3081	185.5	6.51	263.85	5.11	0.022

Table 3: Results of ILP optimisation on CPLEX and CQM D-Wave hybrid solver. $d_{\max} = 40$ was set for all cases. Computational times and QPU time were reported in D-Wave's output.

solution, however, the objective value is somewhat higher than the actual optima obtained with the CPLEX solver. For hardest instances (e.g. case 7 and case 9) the CQM hybrid solver even outperforms CPLEX with respect to computational time (given particular `t_min` setting), but does not find the actual optimum, just a feasible solution close to it. In conclusion, we can expect benefits from application of the hybrid solver on medium scale railway network given scenarios with multiple closures (leading to one track lines) that are not involved in the timetable, as in cases 7 - 9. These results coincide with analogical use-case on synthetic data in Fig. 4.

Let us analyse in more detail the role of the `t_min` solver parameter. For the cases 7 and 9 we have performed runs with `t_min` sweeping over a range; the results are presented in in Fig. 6 and Fig. 7. The right choice of `t_min` can improve results meaningfully. Observe that for small `t_min` parameter values the computational time is shorter, whereas for large `t_min` the improvement in the objective is overwhelmed by the increase of the computational time. The objective value is roughly log-linear in computational time. This is important form the point of view of the algorithm layout, as it may be tied to a power law scaling. Such a behaviour is plausible in the case of quantum

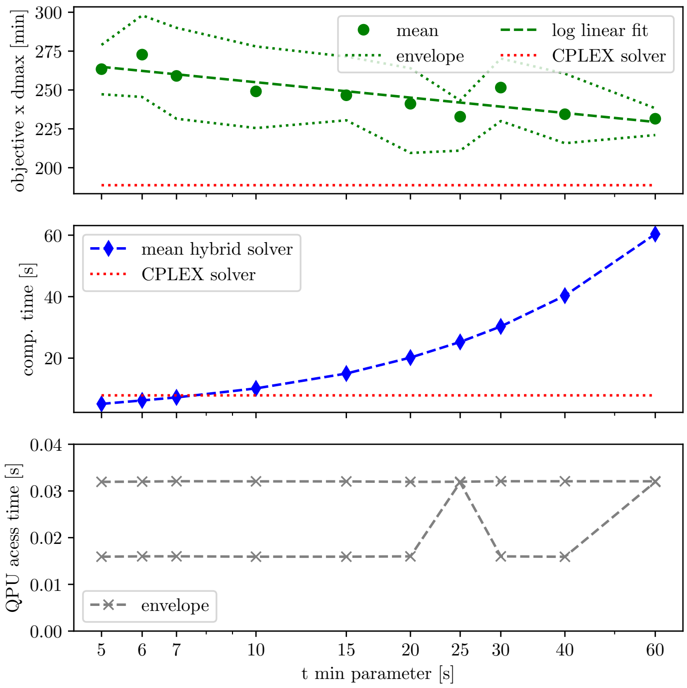

Figure 6: The sweep over `t_min` parameter for case 7 from Tab.3, all solutions were feasible. For higher `t_min` we expect the objective to fall but at the cost of measured total computational time. For each parameter value there were 5 realisations of the experiment were performed. Interestingly, a linear (negatively sloped) relation between the objective and the logarithm of `t_min` can be observed, suggesting some power law scaling.

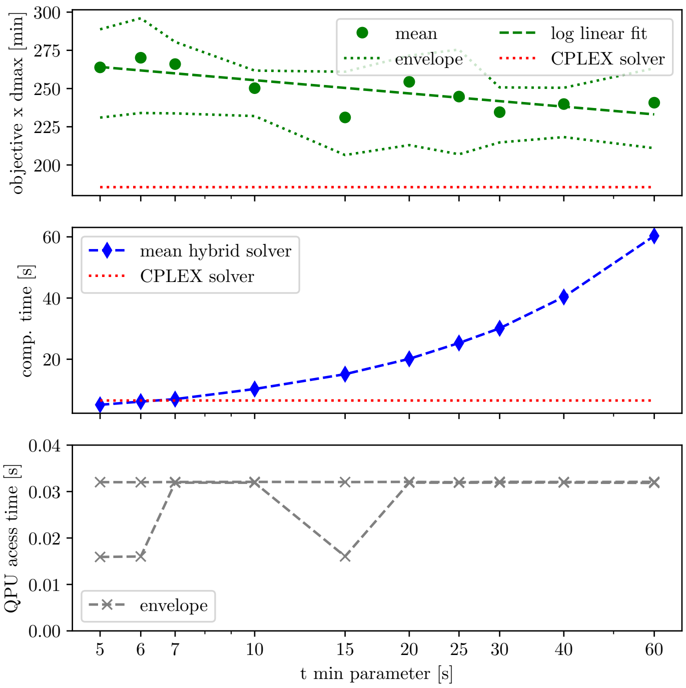

Figure 7: The sweep over `t_min` parameter for case 9 from Tab.3. All solutions were feasible, with results similar to those in Fig. 6.

annealing methods [33, 34].

7 Conclusions

We have demonstrated that quantum annealers can be readily applicable in train dispatching optimisation on a metropolitan scale. While they do not outperform classical solvers in general, we have found examples in which they were actually better. This supports the expectation that future quantum devices will be efficient in solving larger railway problems, e.g. on the country scale; even in the range that are beyond the scope of current exact models and heuristics.

While the quantum-based solvers can possibly return suboptimal solutions, they can still outperform the solutions obtained manually or based on smaller scale models. Importantly for hybrid solvers the QPU times were never zero, meaning that QPU always gave some contribution, though it was not big in particular (compared to the total computational time). Hence, we believe that such a little help from the QPU must have boosted the classical heuristics in the hybrid solver.

Recall that quantum annealer devices are subjects of many debates: previous studies claim that the quantum nature of their operation is limited [55], while recent works argue that their operation relies more on the thermalisation with a cold environment in place of actual quantum adiabatic evolution [56]. They are often criticised for the small size of problems they can address and various classical solvers do outperform them on certain problems at the present state of the art. In spite of all these, however, our computational results demonstrate the readiness of hybrid quantum-classical solvers to handle real-life railway problems.

Data availability

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

Acknowledgements

This research was partially supported by the National Research, Development, and Innovation Office of Hungary under project numbers K133882 and K124351, the Ministry of Innovation and Technology and the National Research, Development and Innovation Office within the Quantum Information National Laboratory of Hungary; and the Polish National Science Center under grant agreements number 2019/33/B/ST6/0201 (LB) and 2023/07/X/ST6/00396 (KD) and by the Silesian University of Technology Rector's Grant no. BKM - 720/RT2/2023 12/020/BKM_2023/0252 (KK).

We would like to thank to Sebastian Deffner, Bartłomiej Gardas, and Zbigniew Puchała for valuable motivation and discussion; and to Katarzyna Gawlak, Akash Kundu, and Özlem Salehi for data validation and assistance with coding. We acknowledge the cooperation with Koleje Śląskie sp. z o.o. (eng. Silesian Railways).

References

[1] L. Ge, S. Voß, and L. Xie, "Robustness and disturbances in public transport," *Public Transport*, Jun 2022. [Online]. Available: https://doi.org/10.1007/s12469-022-00301-8

[2] J. Törnquist, "Railway traffic disturbance management—An experimental analysis of disturbance complexity, management objectives and limitations in planning horizon," *Transportation Research Part A: Policy and Practice*, vol. 41, no. 3, pp. 249–266, 2007.

[3] J. Lange and F. Werner, "Approaches to modeling train scheduling problems as job-shop problems with blocking constraints," *Journal of Scheduling*, vol. 21, no. 2, pp. 191–207, 2018.

[4] A. D'ariano, D. Pacciarelli, and M. Pranzo, "A branch and bound algorithm for scheduling trains in a railway network," *European Journal of Operational Research*, vol. 183, no. 2, pp. 643–657, 2007.

[5] S. Harrod, "Modeling network transition constraints with hypergraphs," *Transportation Science*, vol. 45, no. 1, pp. 81–97, 2011.

[6] G. Caimi, M. Fuchsberger, M. Laumanns, and M. Lüthi, "A model predictive control approach for discrete-time rescheduling in complex central railway station areas," *Computers & Operations Research*, vol. 39, no. 11, pp. 2578–2593, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054812000093

[7] L. Meng and X. Zhou, "Simultaneous train rerouting and rescheduling on an n-track network: A model reformulation with network-based cumulative flow variables," *Transportation Research Part B: Methodological*, vol. 67, pp. 208–234, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0191261514000782

[8] L. Lamorgese, C. Mannino, D. Pacciarelli, and J. T. Krasemann, *Train Dispatching*. Cham: Springer International Publishing, 2018, pp. 265–283. [Online]. Available: https://doi.org/10.1007/978-3-319-72153-8_12

[9] B. Szpigel, "Optimal train scheduling on a single line railway," *Operational Research*, vol. 72, pp. 343–352, 1973.

[10] A. Mascis and D. Pacciarelli, "Job-shop scheduling with blocking and no-wait constraints," *European Journal of Operational Research*, vol. 143, no. 3, pp. 498–517, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221701003381

[11] A. D'Ariano, D. Pacciarelli, and M. Pranzo, "A branch and bound algorithm for scheduling trains in a railway network," *European Journal of Operational Research*, vol. 183, no. 2, pp. 643–657, 2007. [Online]. Available: https://EconPapers.repec.org/RePEc:eee:ejores:v:183:y:2007:i:2:p:643-657

[12] R. M. Lusby, J. Larsen, M. Ehrgott, and D. M. Ryan, "A set packing inspired method for real-time junction train routing," *Computers & Operations Research*, vol. 40, no. 3, pp. 713–724, 2013, transport Scheduling. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054811003595

[13] K. Domino, M. Koniorczyk, K. Krawiec, K. Jałowiecki, S. Deffner, and B. Gardas, "Quantum annealing in the NISQ era: railway conflict management," *Entropy*, vol. 25, no. 2, p. 191, 2023.

[14] D. Venturelli, D. J. Marchand, and G. Rojo, "Job-shop scheduling solver based on quantum annealing," *aint S ing an*, p. 25, 2016.

[15] B. Apolloni, C. Carvalho, and D. De Falco, "Quantum stochastic optimization," *Stochastic Processes and their Applications*, vol. 33, no. 2, pp. 233–244, 1989.

[16] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Physical Review E*, vol. 58, no. 5, p. 5355, 1998.

[17] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk *et al.*, "Quantum annealing with manufactured spins," *Nature*, vol. 473, no. 7346, pp. 194–198, 2011.

[18] K. Domino, A. Kundu, Ö. Salehi, and K. Krawiec, "Quadratic and higher-order unconstrained binary optimization of railway rescheduling for quantum computing," pp. 1–33, 2022.

[19] R. M. Lusby, J. Larsen, and S. Bull, "A survey on robustness in railway planning," *European Journal of Operational Research*, vol. 266, no. 1, pp. 1–15, 2018.

[20] V. Cacchiani, D. Huisman, M. Kidd, L. Kroon, P. Toth, L. Veelenturf, and J. Wagenaar, "An overview of recovery models and algorithms for real-time railway rescheduling," *Transportation Research Part B: Methodological*, vol. 63, pp. 15–37, 2014.

[21] A. Caprara, M. Fischetti, and P. Toth, "Modeling and solving the train timetabling problem," *Operations research*, vol. 50, no. 5, pp. 851–861, 2002.

[22] V. Dal Sasso, L. Lamorgese, C. Mannino, A. Onofri, and P. Ventura, "The tick formulation for deadlock detection and avoidance in railways traffic control," *Journal of Rail Transport Planning & Management*, vol. 17, p. 100239, 2021.

[23] A. Das and B. K. Chakrabarti, "*Colloquium* : Quantum annealing and analog quantum computation," *Reviews of Modern Physics*, vol. 80, no. 3, pp. 1061–1081, Sep. 2008. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.80.1061

[24] E. Ising, "Beitrag zur theorie des ferromagnetismus," *Z. Physik*, vol. 31, no. 1, pp. 253–258, feb 1925. [Online]. Available: https://doi.org/10.1007/bf02980577

[25] Z. Bian, F. Chudak, W. G. Macready, and G. Rose, "The Ising model: teaching an old problem new tricks," *D-wave systems*, vol. 2, pp. 1–32, 2010.

[26] E. Boros and P. L. Hammer, "The max-cut problem and quadratic 0–1 optimization; polyhedral aspects, relaxations and bounds," *Annals of Operations Research*, vol. 33, no. 3, pp. 151–180, 1991.

[27] I. Dunning, S. Gupta, and J. Silberholz, "What Works Best When? A Systematic Evaluation of Heuristics for Max-Cut and QUBO," *INFORMS Journal on Computing*, vol. 30, no. 3, pp. 608–624, aug 2018. [Online]. Available: https://doi.org/10.1287/ijoc.2017.0798

[28] N. Gusmeroli, T. Hrga, B. Lužar, J. Povh, M. Siebenhofer, and A. Wiegele, "BiqBin: A parallel branch-and-bound solver for binary quadratic problems with linear constraints," *ACM Trans. Math. Softw.*, vol. 48, no. 2, pp. 1–31, jun 2022. [Online]. Available: https://doi.org/10.1145/3514039

[29] A. Lucas, "Ising formulations of many np problems," *Frontiers in physics*, p. 5, 2014.

[30] J. E. Avron and A. Elgart, "Adiabatic theorem without a gap condition," *Commun. Math. Phys.*, vol. 203, no. 2, pp. 445–463, 1999. [Online]. Available: https://doi.org/10.1007/s002200050620

[31] S. Zbinden, A. Bärtschi, H. Djidjev, and S. Eidenbenz, "Embedding Algorithms for Quantum Annealers with Chimera and Pegasus Connection Topologies," in *International Conference on High Performance Computing*. Springer, 2020, pp. 187–206.

[32] L. C. Venuti, T. Albash, D. A. Lidar, and P. Zanardi, "Adiabaticity in open quantum systems," *Phys. Rev. A*, vol. 93, p. 032118, Mar 2016. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.93.032118

[33] T. Albash, V. Martin-Mayor, and I. Hen, "Temperature scaling law for quantum annealing optimizers," *Physical review letters*, vol. 119, no. 11, p. 110502, 2017.

[34] K. Domino, M. Koniorczyk, and Z. Puchała, "Statistical quality assessment of ising-based annealer outputs," *Quantum Information Processing*, vol. 21, no. 8, p. 288, 2022.

[35] "D-Wave Hybrid Solver Service: An Overview [WhitePaper]," https://www.dwavesys.com/media/4bnpi53x/14-1039a-b_d-wave_hybrid_solver_service_an_overview.pdf.

[36] D. Willsch, M. Willsch, C. D. G. Calaza, F. Jin, H. D. Raedt, M. Svensson, and K. Michielsen, "Benchmarking advantage and d-wave 2000q quantum annealers with exact cover problems," *Quantum Inf Process*, vol. 21, no. 4, apr 2022. [Online]. Available: https://doi.org/10.1007/s11128-022-03476-y

[37] C. R. McLeod and M. Sasdelli, "Benchmarking D-Wave Quantum Annealers: Spectral Gap Scaling of Maximum Cardinality Matching Problems," in *Computational Science – ICCS 2022*. Springer International Publishing, 2022, pp. 150–163. [Online]. Available: https://doi.org/10.1007/978-3-031-08760-8_13

[38] M. Jünger, E. Lobe, P. Mutzel, G. Reinelt, F. Rendl, G. Rinaldi, and T. Stollenwerk, "Quantum annealing versus digital computing," *ACM Journal of Experimental Algorithmics*, vol. 26, pp. 1–30, dec 2021. [Online]. Available: https://doi.org/10.1145%2F3459606

[39] T. Stollenwerk, E. Lobe, and M. Jung, "Flight Gate Assignment with a Quantum Annealer," in *Quantum Technology and Optimization Problems*, S. Feld and C. Linnhoff-Popien, Eds. Cham: Springer International Publishing, 2019, pp. 99–110.

[40] P. N. Tran, D.-T. Pham, S. K. Goh, S. Alam, and V. Duong, "An Interactive Conflict Solver for Learning Air Traffic Conflict Resolutions," *Journal of Aerospace Information Systems*, vol. 17, no. 6, pp. 271–277, jun 2020. [Online]. Available: https://doi.org/10.2514%2F1.i010807

[41] T. Stollenwerk, B. O'Gorman, D. Venturelli, S. Mandrà, O. Rodionova, H. Ng, B. Sridhar, E. G. Rieffel, and R. Biswas, "Quantum anneal-

ing applied to de-conflicting optimal trajectories for air traffic management," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 285–297, 2020.

[42] S. Yarkoni, A. Huck, H. Schülldorf, B. Speitkamp, M. S. Tabrizi, M. Leib, T. Bäck, and F. Neukart, "Solving the shipment rerouting problem with quantum optimization techniques," in *Computational Logistics*, M. Mes, E. Lalla-Ruiz, and S. Voß, Eds. Cham: Springer International Publishing, 2021, pp. 502–517.

[43] K. Domino, M. Koniorczyk, K. Krawiec, K. Jałowiecki, and B. Gardas, "Quantum computing approach to railway dispatching and conflict management optimization on single-track railway lines," 2021.

[44] C. Grozea, R. Hans, M. Koch, C. Riehn, and A. Wolf, "Optimising rolling stock planning including maintenance with constraint programming and quantum annealing," Sep 2021. [Online]. Available: http://arxiv.org/abs/2109.07212v1

[45] F. Neukart, G. Compostella, C. Seidel, D. Von Dollen, S. Yarkoni, and B. Parney, "Traffic flow optimization using a quantum annealer," *Frontiers in ICT*, vol. 4, p. 29, 2017.

[46] S. Yarkoni, A. Alekseyenko, M. Streif, D. Von Dollen, F. Neukart, and T. Bäck, "Multi-car paint shop optimization with quantum annealing," in *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2021, pp. 35–41.

[47] L. N. Martins, A. P. Rocha, and A. J. Castro, "A QUBO Model to the Tail Assignment Problem." in *ICAART (2)*, 2021, pp. 899–906.

[48] A. Glos, A. Kundu, and Ö. Salehi, "Optimizing the production of test vehicles using hybrid constrained quantum annealing," *SN Computer Science*, vol. 4, no. 5, p. 609, 2023.

[49] F. Corman, A. D'Ariano, D. Pacciarelli, and M. Pranzo, "Bi-objective conflict detection and resolution in railway traffic management," *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 79–94, 2012, special issue on Optimization in Public Transport+ISTT2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X10001452

[50] "Hybrid Solver for Constrained Quadratic Model [WhitePaper]," https://www.dwavesys.com/media/rldh2ghw/14-1055a-a_hybrid_solver_for_constrained_quadratic_models.pdf.

[51] T. Tran, M. Do, E. Rieffel, J. Frank, Z. Wang, B. O'Gorman, D. Venturelli, and J. Beck, "A hybrid quantum-classical approach to solving scheduling problems," in *Proceedings of the International Symposium on Combinatorial Search*, vol. 7, no. 1, 2016, pp. 98–106.

[52] S. Karimi and P. Ronagh, "Practical integer-to-binary mapping for quantum annealers," *Quantum Information Processing*, vol. 18, no. 4, pp. 1–24, 2019.

[53] N. Dattani, S. Szalay, and N. Chancellor, "Pegasus: The second connectivity graph for large-scale quantum annealing hardware," Jan 2019. [Online]. Available: https://arxiv.org/abs/1901.07636

[54] PKP Polskie Linie Kolejowe S.A., "Instrukcja o prowadzeniu ruchu pociagów Ir-1 (Eng. Instruction on the operation of train traffic)." [Online]. Available: https://www.plk-sa.pl/klienci-i-kontrahenci/akty-prawne-i-przepisy/instrukcje-pkp-polskich-linii-kolejowych-sa/

[55] S. W. Shin, G. Smith, J. A. Smolin, and U. Vazirani, "How "quantum" is the d-wave machine?" Jan 2014. [Online]. Available: http://arxiv.org/abs/1401.7087v2

[56] L. Buffoni and M. Campisi, "Thermodynamics of a quantum annealer," *Quantum Sci. Technol.*, vol. 5, no. 3, p. 035013, jun 2020. [Online]. Available: https://doi.org/10.1088/2058-9565/ab9755