scientific reports

Check for updates

OPEN

On the Baltimore Light RailLink into the quantum future

Krzysztof Domino[1,6]✉, Emery Doucet[2,3,6]✉, Reece Robertson[2,3,4], Bartłomiej Gardas[1] & Sebastian Deffner[2,3,5]

In the current era of noisy intermediate-scale quantum (NISQ) technology, quantum devices present new avenues for addressing complex, real-world challenges including potentially NP-hard optimization problems. Acknowledging the fact that quantum methods underperform classical solvers, the primary goal of our research is to demonstrate how to leverage quantum noise as a computational resource for optimization. This work aims to showcase how the inherent noise in NISQ devices can be leveraged to solve such real-world problems effectively. Utilizing a D-Wave quantum annealer and IonQ's gate-based NISQ computers, we generate and analyze solutions for managing train traffic under stochastic disturbances. Our case study focuses on the Baltimore Light RailLink, which embodies the characteristics of both tramway and railway networks. We explore the feasibility of using NISQ technology to model the stochastic nature of disruptions in these transportation systems. Our research marks the inaugural application of both quantum computing paradigms to tramway and railway rescheduling, highlighting the potential of quantum noise as a beneficial resource in complex optimization scenarios.

It is an important goal to realize scalable and fault-tolerant quantum computers[1–3]. However, the technological challenges are immense and currently available devices are characterized and governed by noise. These noisy intermediate-scale quantum (NISQ) devices[4] are already capable of practical computations, yet any application will have to embrace numerical imperfections and shortcomings. Thus, the obvious and important question arises whether the inevitably noisy characteristics can be exploited as a resource.

The applicability of quantum computing has already been demonstrated in optimization problems for industrial and operations research tasks[5]. A particular area of interest is railway transportation, where, for instance, a few hours of operation of a single metro line with about 13 stations and 150 trains was solved on a current classical computer in approximately 5 h. In practice, re-scheduling and dispatching decisions must be made in seconds[6] and hence new technologies such as a quantum approach are desirable[6–8].

Only rather recently, employing NISQ for railway scheduling has been demonstrated [9]. In this work, re-scheduling was expressed as a quadratic unconstrained binary optimization (QUBO) problem, which was then solved using a D-Wave quantum annealer—for the conceptual extension of the model to a higher order unconstrained binary optimization (HOBO) approach see [10]. Importantly, while ideally the output of a quantum annealer is the exact optimal solution (corresponding to the ground state of the Ising model used to encode the QUBO), higher-lying excited states are often returned instead. In terms of the original scheduling problem, these higher-lying states may correspond to rail traffic with additional disturbances. Referring to railways see Ref. [11] (high-speed trains timetable optimization on coherent Ising machine quantum simulator), Ref. [12] (rolling stock planning on the quantum annealer for German Railways Network), and Refs. [13,14] (finding the rail transportation plan on the IBM 32-qubit QASM simulator using QUBO/HOBO and quantum approximate optimization algorithm—QAOA—encoding) for a complementary approaches. However, these studies did not tie the noise of NISQ device with the stochastic behavior of a particular public transportation system.

In the present work, we demonstrate the utility of NISQ devices (and systematically compare annealers and gate-based computers) for the problem of railway scheduling by leveraging the inherent noise present in all such devices as a tool to model stochastic behavior in real-world problems, an idea that will prove especially relevant

[1]Institute of Theoretical and Applied Informatics, Polish Academy of Sciences, Bałtycka 5, 44-100 Gliwice, Poland. [2]Department of Physics, University of Maryland, Baltimore County, Baltimore, MD 21250, USA. [3]Quantum Science Institute, University of Maryland, Baltimore County, Baltimore, MD 21250, USA. [4]Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250, USA. [5]National Quantum Laboratory, College Park, MD 20740, USA. [6]Krzysztof Domino and Emery Doucet contributed equally to this work. ✉email: kdomino@iitis.pl; emery.doucet@umbc.edu

to the development of hybrid quantum/classical solution methods—see Supplementary Information for pseudo code. Such hybrid methods can tackle significantly larger problems than either the classical or NISQ approach by themselves [15].

Technically, our approach follows the ideas presented in Ref. [16] but with a focus on the practical application of railway re-scheduling. We focus on rail systems with some stochastic features as can be found in tramways, which partially share infrastructure with road traffic. Some research has been performed on this problem[17] and on stochastic optimization of tramway scheduling in general, e.g., in the Hong Kong tram network[18], however literature on this topic is sparse[8] and generally with a different focus such as infrastructure repair scheduling[19] or vehicle and crew rescheduling[20]. In our previous research, non-optimal but practically sound solutions of railway scheduling [21] and analogous Automated Guided Vehicles scheduling [22] problems have been obtained from D-Wave's proprietary hybrid solvers. Unfortunately, these solvers are closed black boxes and it isn't easy to assess the actual effort done by the QPU during optimization. Given such experience, in this work we concentrate on pure quantum computing with trackable performance.

In our work, we focus on a specific real-world transportation system: the Baltimore Light RailLink in Baltimore, Maryland[23] The Baltimore Light RailLink system is an important factor in the economic development in the urban area surrounding the network[24], and current data on its performance[25] indicates that there is a margin for improvement. These properties of the Baltimore Light RailLink infrastructure are in no way unique and are found in other tramway infrastructures (for example that in Karlsruhe, Germany[26]), therefore the conclusions we draw in this work are general.

In Fig. 1 we depict the portion of the Baltimore Light RailLink network we consider in this work on a map and schematically. Between Camden Station and Mount Royal the trains are subject to essentially random variations in transit time due to road traffic, hence we denote the segment of the network bounded by these two stations as the "stochastic zone". Outside of this region, for example north to Mount Royal, the trains travel on dedicated infrastructure and travel times are deterministic. In this work, we analyze dispatching situations for this portion of the network in terms of *decision stations*[21], with a focus on the passing time of trains through the stochastic zone.

Our approach can serve as one component of a multilevel combinatorial optimization approach combining the strengths of existing classical solvers with quantum architectures[27] for the purposes of railway scheduling, especially when the underlying network exhibits some form of stochasticity. These multilevel algorithms allow the solution of problems an order of magnitude larger than current quantum hardware can handle directly, and we expect that a hybrid algorithm based on our techniques can be applicable to large and practically-relevant railway problems, for example of size comparable with these in Ref. [7].

This article is organized as follows: In "Mathematical models" we present the model of the Baltimore Light RailLink and solution methods for both the quantum annealing and gate-based quantum computing paradigms. In "Computational results" we present results from experiments with each of these approaches, using machines from D-Wave and IonQ, alongside simulations. We discuss these results in light of real-life data measured from railway traffic, before presenting our conclusions in "Conclusions". Additional details on encoding constraints in QUBO problems, on a proposed hybrid algorithm that can be applied to larger railway scheduling problems, on hardware details and embeddings or circuits used for our experiments with a D-Wave quantum annealer and IonQ's gate-based trapped ion computer, and on some tests using IBM's superconducting gate-based computers are available as supplementary material.
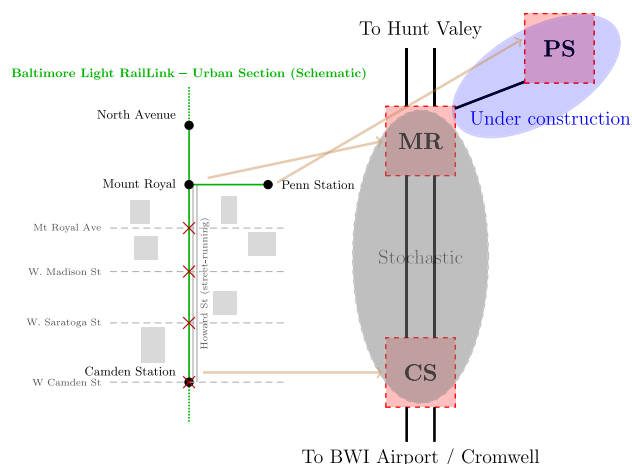

Fig. 1. Map based on[28] (left) and schematic representation (right) of the Baltimore Light RailLink system we consider. The stochastic zone with shared infrastructure runs between Camden Station (CS) and Mount Royal (MR). There is then a deterministic connection with dedicated infrastructure connecting Mount Royal to Penn Station (PS)—Figure generated by *LaTeX tikz (pdfTeX 3.141592653-2.6-1.40.25)*.

Mathematical models

The basic idea behind our research is to apply the NISQ device to model the trains' traffic in the stochastic zone in Fig. 1, as quantum computers are NISQ devices [4] and so their output is inherently noisy and stochastic. For encoding this problem in a form amenable to quantum computing, we follow the outline of [9]. We first encode the scheduling problem as an Integer Linear Programming (ILP) problem, which is then transformed into a Quadratic Unconstrained Binary Optimization (QUBO) problem which serves as the input for various optimization methods developed for quantum computers.

Integer linear programming approach

The state-of-the-art approach[6] to railway scheduling—and re-scheduling in particular—is to encode the problem as an ILP problem and solve it using high-quality solvers such as CPLEX. Stochastic components are often handled by solving many ILP problems with various parameters [18], or with simulations, see e.g. [29]. In this section, we present the mapping of our train scheduling problem to an ILP problem. To do so it is necessary to specify the variables which are used to encode a solution timetable, the constraints imposed on those variables, and finally the objective used to evaluate solution timetables for optimality. Throughout, we will concentrate primarily on re-scheduling problems where the input timetable requires modification, for example, due to it being infeasible as a result of disturbances causing constraint violations.

Variables

To begin, we assign numeric identifiers to the trains we wish to schedule and to the stations they will visit. We will use the subscript j to select a particular train, and $\#J$ to denote the total number of trains. Similarly, the subscript s will select a specific station and $\#S$ will indicate the total number of stations.

We introduce a set of non-negative integer variables

$$t_{s,j}^{\text{in}} \in \mathbb{Z}^+, \tag{1}$$

which represent the time train j enters station s. As is typical for railway scheduling, we measure times with a resolution of one minute[6]. Note that if s is the first station in the train's j route, then $t_{s,j}^{\text{in}}$ is the time the train is ready to start its route. After the minimal stop at the station, it can proceed.

We denote the arrival time of train j at station s as given by the not-disturbed (reference) timetable by $\tau_{s,j}^{\text{in}}$. The earliest time at which the train could arrive is denoted by $l_{s,j} \geq \tau_{s,j}^{\text{in}}$. This is just the non-disturbed timetable arrival time if the train has not been delayed, or if the input timetable is infeasible the lowest possible arrival time given no rail traffic. If we assume that there is a maximal allowed additional (called also secondary[30]) delay given by $d_{\max}$, then we may also define a latest possible arrival time $u_{s,j} = l_{s,j} + d_{\max}$ and so

$$l_{s,j} \leq t_{s,j}^{in} \leq u_{s,j} \quad \text{equivalently} \quad t_{s,j}^{in} \in R_{s,j}. \tag{2}$$

Importantly, the maximal delay parameter $d_{\max}$ determines the problem size by constraining the range of time variables. We have at most $\#S\#J$ such time variables, as we do not consider recirculation [31].

For the purposes of our model, we assume that the stay time of the train at the station is constant and equal to δ^{station}. We then define the time at which train j leaves station s,

$$t_{s,j}^{\text{out}} = t_{s,j}^{\text{in}} + \delta^{\text{station}}. \tag{3}$$

Alongside the variables we have just defined which describe the times at which trains arrive at and depart from stations, we introduce binary variables which specify the precedence of trains. Specifically, define

$$y_{j,j',s} \in \{0,1\} \tag{4}$$

so that $y_{j,j',s} = 1$ if train j enters station s before train j' (by considering the reversed case, we see that $y_{j,j',s} = 1 - y_{j',j,s}$).

Since we have imposed a maximal delay time of $d_{\max}$, then following the discussion in[21] we expect that each train has a dependency on (i.e., is potentially in conflict with) a roughly constant number of trains, regardless of how large the problem is. Hence, the number of y variables is proportional to $\#J$ and $\#S$.

Constraints

We impose three types of constraints which must be satisfied for a solution to represent a valid schedule. Additionally, as all trains move with similar average speeds and have the same priority we expect that they will not meet and overtake at or between stations. This condition is not encoded explicitly but will be checked for any solution.

The first of constraint we consider is the **minimal headway** constraint, which is taken to be deterministic. If two trains are heading in the same direction, they are required to maintain a minimal headway time δ^{headway} between them. Explicitly, let H_s be the set of all pairs of trains which have a minimal headway dependency entering s. Then:

$$\forall_s \forall_{(j,j') \in H_s} \begin{cases} t_{s,j'}^{\text{in}} \geq t_{s,j}^{\text{in}} + \delta^{\text{headway}} & \text{if } y_{j,j',s} = 1 \\ t_{s,j}^{\text{in}} \geq t_{s,j'}^{\text{in}} + \delta^{\text{headway}} & \text{if } y_{j,j',s} = 0. \end{cases} \tag{5}$$

From this, we conclude that we have one minimal headway constraint per y variable. The average size of H_s over the set of stations depends on the $d_{\max}$ parameter and is expected to be proportional to the number of trains.

The second constraint is the **rolling stock circulation** constraint, which is also taken to be deterministic. If two trains are heading in opposite directions, they may use the same rolling stock so be dependent. In such a case, if the train j terminates at s, then after a preparation time $\delta^{\text{preparation}}$ the subsequent train j' is ready to start its journey backwards. Let RS_s be the set of all such pairs of trains, which is timetable-dependent. Then:

$$\forall_{(j,j')\in RS_s} t^{\text{in}}_{s,j'} \geq t^{\text{in}}_{s,j} + \delta^{\text{preparation}} + \delta^{\text{station}}. \tag{6}$$

There are far fewer rolling stock circulation constraints than there are minimal headway constraints.

The final constraint we consider is a constraint on the **minimal passing time**, and it is here where the underlying stochasticity of the model appears. In the deterministic scheduling case, the minimal passing time between stations is given by the train-independent constant $\delta^{\text{pass}}_{s,s'}$. Then the minimal passing time constraint implies:

$$\forall_j \forall_{(s,s')\in SP_j} t^{\text{in}}_{s',j} \geq t^{\text{out}}_{s,j} + \delta^{\text{pass}}_{s,s'}, \tag{7}$$

where SP_j is a set of all pairs of subsequent stations in the route of j. The average size (over trains) of SP_j, namely $|SP_j|$, is expected to be proportional to the total number of stations in the model.

As argued when introducing our model shown in Fig. 1, the effect of road traffic on the shared infrastructure will be to add some randomness to the passing time through the stochastic zone. To model this, we modify Eq. (7) to:

$$\forall_j \forall_{(s,s')\in SP_j} t^{\text{in}}_{s',j} \geq t^{\text{out}}_{s,j} + \delta^{\text{pass}}_{s,s'} + w. \tag{8}$$

where $w \in \mathcal{W}$ is the particular realization of a non-negative stochastic factor representing the additional unpredictable delay. To gather statistics of the problem, such an approach would require solving many ILP problems that employ the constraints of Eq. (8) with a range of actual values of w, as there are $|\mathcal{W}|$ such possibilities for each inequality. In the worst case, we expect to need to solve approximately $\#J\#S|\mathcal{W}|$ ILP problems, which could be a very large number. In this work, we demonstrate that a similar outcome can be achieved from a single set of runs of a quantum device, which can be performed in seconds.

All together, from Eqs. (7) and (5) and the surrounding discussion we expect the number of constraints to be linear in $\#J$ and $\#S$.

Objective
The objective function we seek to minimize when constructing a timetable is the sum of delays at some selected stations listed in S^*,

$$f = \sum_j \sum_{s\in S^*} \frac{t^{\text{in}}_{s,j} - \tau^{\text{in}}_{s,j}}{d_{\max}}. \tag{9}$$

We assume that the deterministic approach with minimal passing time constraints given by Eq. (7) yields the optimal solution. As w in Eq. (8) is non-negative, the stochastic approach may result in feasible solutions with higher values for the objective f. This is also true of the output of a minimization using a NISQ device. The remainder of this paper is dedicated to examining this observation.

QUBO approach for quantum computing

The ILP problem formulated in the previous section defines our scheduling problem, however it is not in a form which is easily solvable with NISQ devices. For that, it is necessary to move to a QUBO representation (for a detailed discussion of the QUBO formulation, see[32]). Following[9], we define the optimization problem in terms of the binary decision variables

$$x_{s,j,t} = \begin{cases} 1 & \text{if train } j \text{ enters station } s \text{ at time } t \\ 0 & \text{otherwise} \end{cases}. \tag{10}$$

where t is in the interval $R_{s,j}$ defined in Eq. (2). The number of these decision variables N is at most $\#J\#S(d_{\max}+1)$, as some trains may serve not all stations.

The binary decision variables in Eq. (10) can be gathered into a vector $\vec{x} \in \{0,1\}^N$, where the elements of $\vec{x}$ are labeled x_i, where each i is the flattened version of the corresponding multi-index s, j, t introduced in Eq. (10). Then, the (re-)scheduling problem becomes equivalent to finding the vector $\vec{x}$ which minimizes the quadratic form

$$E(\vec{x}) = \vec{x}Q\vec{x}^T \in \mathbb{R}, \tag{11}$$

to determine the minimum objective value $E_0 = \min_{\vec{x}} E(\vec{x})$ and optimal solution $\vec{x}_0 = \operatorname{argmin} E(\vec{x})$. Q is the matrix encoding the constraints and objective function, which can be taken to be symmetric. The details on encoding the minimal passing time [cf. Eq. (7)] minimal headway [cf. Eq. (5)] and rolling stock circulation [cf. Eq. (6)] constraints as discussed in "Integer linear programming approach" in terms of a quadratic form on

the binary decision variables are presented in the Supplementary Information. Ultimately, we expect both the number of qubits (N) and the number of non-zero entries in the Q matrix to be proportional to $\#J\#S$. Hence, for constant $d_{\max}$, the mean degree of the problem graph is expected to be independent on $\#J$ and $\#S$.

These constraints are all quadratic and have the form

$$\sum_{i,i'} x_i x_{i'} = 0. \tag{12}$$

Transforming from the constrained optimization problem of the original scheduling problem and subsequent ILP formulation to an unconstrained problem which can be encoded as a QUBO is performed by making the hard constraints soft with an associated penalty for constraint violations,

$$\sum_{i<i'} p_{\text{pair}}(x_i x_{i'} + x_{i'} x_i). \tag{13}$$

Once expressed in this form, it is straightforward to input the constraints into the Q matrix of Eq. (11). Following [9], we chose one value p_{pair} for all quadratic terms in Eq. (13) associated with violations of the constraints, but the generalization is straightforward. The number of these constraints is expected to be linear in $\#S\,\#J$, as discussed in "Integer linear programming approach" and the Supplementary Information.

We have another set of constraints, the so-called one-hot constraints[33]. These constraints ensure that each train leaves each station once and only once:

$$\forall_{s,j \in (S,J)} \sum_{t \in R_{j,s}} x_{s,j,t} = 1, \tag{14}$$

wherein the set of pairs of trains and stations is denoted by (S, J) and where t runs over the range $R_{j,s}$ introduced in Eq. (2). The number of these sums is limited by $\#S\#J$, as at most each train passes each station. This can be written in unconstrained form with a penalty,

$$\sum_{s,j \in (S,J)} p_{\text{sum}} \left(\sum_{t \in R_{j,s}} x_{s,j,t} - 1 \right)^2, \tag{15}$$

which is then transformed using the identity $x_{s,j,t}^2 = x_{s,j,t}$ yielding finally,

$$\sum_{s,j \in (S,J)} p_{\text{sum}} \left(\sum_{\substack{t,t' \in R_{j,s} \\ t \neq t'}} x_{s,j,t} x_{s,j,t'} - \sum_{t \in R_{j,s}} x_{s,j,t}^2 \right). \tag{16}$$

We expect approximately $(d_{\max} + 1)^2 \#J\#S$ such terms.

The objective is linear and is directly derived from Eq. (9):

$$f(\vec{x}) = \sum_{j} \sum_{s \in S^*} \left(\sum_{t \in R_{j,s}} \frac{t - \tau_{s,j}^{\text{in}}}{d_{\max}} x_{s,j,t} \right). \tag{17}$$

The solution to the QUBO of Eq. (11) will correspond to the optimal solution to the scheduling problem. We expect that nearly optimal solutions whose objective values are not much larger will correspond to solutions which are consistent with the constraints but which include increased passing time between stations. Solutions further from optimal with larger values of the objective function will then typically correspond to solutions which fail to satisfy the pair and sum constraints of the forms presented in Eqs. (13) and (15), respectively. The interplay between these two situations is controlled by the penalty parameters in the QUBO, p_{sum} and p_{pair}.

Identifying good values p_{sum} and p_{pair} for the penalty parameters is a complicated problem, and recent years have seen the development of dedicated algorithms for this purpose[34]. These algorithms have, however, complexity that must be included for in estimates of the computational time of the final problem (e.g., the general procedure of finding the proper quantum encoding/circuits is NP-hard itself [35]). Furthermore, algorithmic approaches are typically based on deriving a lower bound for the penalty coefficients necessary to enforce the various constraints by splitting feasible and non-feasible parts of the spectrum. Then, one expects the penalty from the violation of constraint to be larger than any possible changes of the objective. In practice, better results are often recorded for penalty coefficients smaller than the above-mentioned lower limits[36]. In this case, the spectra corresponding to the feasible and non-feasible solution spaces overlap. We expect that this yields some positive impact on the solution process, e.g. by smoothing local minima in the quantum evolution process.

To understand this effect, the studies presented in this work were all repeated twice. In one case which we call the *overlapping spectrum* case, we choose the penalty values such that one broken constraint corresponds to $4\times$ the worst objective at a single train and station, $p_{\text{sum}} = 4$ and $p_{\text{pair}} = 2$. The other case we call the *split spectrum* case, where the penalties are made $10\times$ larger: $p_{\text{sum}} = 40$ and $p_{\text{pair}} = 20$.

Recall that if we explicitly require that trains do not meet and overtake at or between stations (see "Integer linear programming approach"), a HOBO encoding such as in Ref.[10] would be required. However, such an approach would substantially complicate the problem for the annealer, leading to rather limited practical benefit in our case. For example, in Ref.[10] HOBO problems corresponding to at most 3 trains and 2 stations were handled successfully by pure quantum annealing on the D-Wave machine. Nevertheless, we acknowledge, that the HOBO encoding can be more competitive when applying the gate-based quantum computing via the QAOA approach, which opens a particular avenue for further research.

Ising model for quantum annealing

The final step necessary to express the scheduling problem in a form amicable to solution on a NISQ device is to transform from QUBO to Ising representation. This reformulation is straightforwardly implemented by introducing the spin variables $s_i \in \{-1, 1\}$ and taking $s_i = 2x_i - 1$, where x_i are the variables in the original QUBO representation. In this way, the QUBO formulation in Eq. (11) is transformed into the Ising Hamiltonian:

$$H(\vec{s}) = \sum_{(i,i')} J_{i,i'} s_i s_{i'} + \sum_i s_i h_i. \tag{18}$$

Here $J_{i,i'}$ and h_i are couplings and constant terms resulting from particular Q in QUBO in Eq (11). Minimization of the latter over spin configurations is consistent with the minimization of the corresponding Eq. (11) with binary variables.

Quantum annealers such as those manufactured by D-Wave implement the quantum version of the Ising model, where spin variables become Pauli matrices acting on spin's subspace. The Ising representation is the native input of the quantum annealer, where the original problem in Eq. (18) turns into the search for the lowest-energy (or low-energy in practice) eigenstate of the following Hermitian Hamiltonian:

$$H = \sum_{(i,i')} J_{i,i'} \sigma_i^z \sigma_{i'}^z + \sum_i \sigma_i^z h_i. \tag{19}$$

The spin variable $s_i = \pm 1$ in Eq. (18) is associated with σ_i^z via $\sigma_i^z |s_i\rangle = s_i |s_i\rangle$, where $|s_i\rangle$ form a local computational basis.

The basic idea of quantum annealing relies on the celebrated adiabatic theorem [37]. One prepares the initial state of the quantum system in the ground state of the initial Hamiltonian $H_0 = \sum_i \sigma_i^x h_i$, then slowly evolves the Hamiltonian to H_p (the evolution time being called the annealing time T). If the evolution is sufficiently slow, then the final state will ideally be the ground state or at least a low-lying excited state of the final Hamiltonian, Eq. (19) in this case, and hence a low-energy state of the original Ising model of Eq. (18). The performance of this process is dependent on a variety of factors, e.g. on the eigenvalues of the Hamiltonian of Eq. (19) and their spacing. These relationships are complex, and it is not obvious what will be observed when analyzing two QUBOs with different penalty coefficients and hence different spectra.

As a final note, the Ising Hamiltonians corresponding to our scheduling problems are defined on a dense graph. Real quantum annealers do not have all-to-all connectivity, and so mapping our problems onto the native coupling graph of an actual quantum annealer [38] requires the use of a standard procedure called "minor embedding". This embedding leads the overhead in the number of quantum bits and limits the size of the problem that can be handled by any given quantum annealer. As the size of the problem graph is linear in $\#J$ and $\#S$ in terms of number of logical qubits and number of connections, we expect the number of physical qubits to be also proportional to $\#J$ and $\#S$. Hence, we do not expect an explosion in the required size of quantum processor necessary for large problems of practical relevance.

Model for gate quantum computing

Besides quantum annealing, the most common approaches to solving combinatorial optimization problems on NISQ-era quantum computers use the gate-based paradigm of quantum computing with variational algorithms. These algorithms make use of standard optimization algorithms running on a classical computer, with a quantum computer being used to evaluate the objective function which is to be minimized. There are various such algorithms[39,40], which employ various variational ansatzes and are tuned for different purposes.

In this work, we use the quantum approximate optimization algorithm (QAOA)[41]. Given an N-qubit diagonal Hamiltonian H_C representing a cost function to be minimized (such as the Ising Hamiltonian encoding the scheduling problem given by Eq. (19)) and a mixing Hamiltonian (typically $H_M = \sum_i \sigma_i^x$), the algorithm proceeds by using a quantum computer to generate the ansatz state

$$|\vec{\beta}, \vec{\gamma}\rangle = e^{-i\beta_p H_M} e^{-i\gamma_p H_C} \dots e^{-i\beta_1 H_M} e^{-i\gamma_1 H_C} |+\rangle^{\otimes N} \tag{20}$$

and compute the expectation value of the cost Hamiltonian $\langle \vec{\beta}, \vec{\gamma} | H_C | \vec{\beta}, \vec{\gamma} \rangle$ where $\vec{\beta}$ and $\vec{\gamma}$ are p-dimensional vectors of coefficients characterizing each of the p layers of the ansatz. Once the $2p$ parameters that minimize this expectation value are identified, the corresponding ansatz state provides a good approximation of the ground state of the cost Hamiltonian, and so the prepared ansatz state in the computational basis yields an approximate solution to the initial combinatorial optimization problem. To find the optimal parameters, the computation of the expectation value using a quantum computer is treated as a black box function to be optimized with some classical optimization method, for example, the COBYLA method[42] which has been shown to work well on these types of problems[43].

When comparing the process of solving a QUBO using QAOA against using a quantum annealer such as D-Wave, it is clear that QAOA is, in some sense, a more expensive process. One optimization using the algorithm-essentially equivalent to a single shot on a quantum annealer-requires a classical optimization to be run. This requires that a quantum computer be used to run a series of circuits with different parameter values to extract the objective function. Each of these requires many independent shots to sample the prepared ansatz to compute the objective function reliably.

It is nonetheless interesting to compare the performance of a general-purpose gate-based quantum computer against a dedicated quantum annealer on combinatorial optimization problems. While QAOA is inspired by adiabatic computation with a quantum annealer, it is not clear if the two models should be expected to exhibit similar behaviour as various problem parameters are adjusted (e.g., the problem size or the choice of penalty parameters). We also expect that the effect of noise present in current NISQ-era implementations may not affect the optimization result from a gate-based computer implementing QAOA in the same way as a quantum annealer.

As a final note, our decision to employ QAOA as our algorithm of choice for solving our problem instances using gate-based quantum hardware is motivated primarily by simplicity, efficacy, and ease of comparison to the results of quantum annealing. One of the major limitations of this approach is the limited qubit count available with presently-available gate-based quantum computers. This limitation can be mitigated in several ways, for example by considering hybrid quantum-classical optimizers that only solve subproblems on the quantum hardware (as we discuss in the Supplementary Information), or by considering clever encoding schemes that allow the solution of large problems with few qubits[44–46]. Such approaches are especially important on current NISQ hardware, as smaller circuits on fewer qubits promise higher accuracy and better optimization performance. An performance advantage due to encoding may also be possible if instead of distilling the scheduling problem to a QUBO, it is encoded as a higher-order binary optimization problem (HOBO). QAOA applied to higher-order problems can require fewer qubits, at the cost of increased circuit depth[47,48].

Embracing the gate-based paradigm completely, there also exist approaches capable of solving scheduling problems without initially reducing the problem to a QUBO, for instance through quantum versions of the branch-and-bound algorithm[49,50]. These algorithms pose a challenge for current devices, but in the future may be a promising avenue towards solving large-scale problems with gate-based quantum hardware.

Computational results

To test our approach, we concentrate primarily on the stochastic zone of the Baltimore Light RailLink identified in the introduction and depicted in Fig. 1. This stochastic zone is the central part of the Baltimore Light RailLink between station Camden Station (CS) and Mount Royal (MR) station where railway traffic is shared with road traffic, and is connected to Penn Station (PS) with a section of track, currently under reconstruction. We model these three stations as *decision stations*[21], the stations at which scheduling decisions are made.

From a real Baltimore Light RailLink timetable[51] we can derive a set of model parameters. For the whole network, these are the minimum headway time $\delta^{\text{headway}} = 2$, the minimum preparation time $\delta^{\text{preparation}} = 3$, and minimum time in station $\delta^{\text{station}} = 1$. For the passage between stations MR and PS, the minimum passing times in the two directions are $\delta^{\text{pass}}_{\text{MR, PS}} = 14$ and $\delta^{\text{pass}}_{\text{PS, MR}} = 2$.

To model train traffic on this segment, we start with the real Baltimore Light RailLink timetable without any disturbances. Note that in our model we also include several trains whose route either starts from or ends at station PS, though these trains do not run at the current time as station PS is closed due for reconstruction[52]. This allows us to model rather dense railway traffic, and so we may limit the maximum secondary delay d_{max} which ensures the QUBO representations are not too large.

To ensure that our analysis is as thorough as possible, we generate various scheduling problem instances which incorporate from a number of trains $\#J \in \{1, 2, 4, 6, 8, 10, 11, 12\}$ and which have maximal delay parameters d_{max} of either 2 or 6 min. We divide the scheduling problems we consider into two classes:

- Non-disturbed, where the initial condition (input timetable) is feasible, as there are no delays of trains. In this case, the optimal solution (the solution to the deterministic scheduling problem) has an objective value of zero. Feasible solutions with non-zero objective values are expected to correspond to random disturbances in the stochastic zone.
- Disturbed, where delays have been injected into the input timetable. Here, the input problem can be infeasible even without stochastic disturbances as the delay may lead to constraint violations without adjustments to the schedule. Again, the optimal solution solves the deterministic version of the problem, and feasible solutions with slightly higher objective values are expected to correspond to random disturbances in the stochastic zone (bear in mind, that for one train instance, we do not have a disturbed case).

In total, we consider 30 distinct scheduling problems. When translated into the QUBO formulation, the smallest problem we consider (with a single train serving 2 stations and a maximum delay of 2 min, where the delay is counted from zero) requires 6 variables to encode. Mapped to a quantum Ising model for quantum annealing, this corresponds to 6 logical qubits. The largest problem (with 12 trains, each serving 2 or 3 stations, and a maximum delay of 6 min) requires 196 variables, which is comparable in size to the largest railway scheduling problems solved previously on a D-Wave machine in[9]. Note that since the D-Wave machine does not support arbitrary connectivity between its qubits, there is some overhead in embedding the QUBOs corresponding to our scheduling problems. For a deeper discussion, see the Supplementary Information. Fig. 2a shows the relationship between the number of logical qubits required for each QUBO we consider and the number of physical qubits necessary after embedding, which can be seen to be approximately linear. Also shown in Fig. 2b is an extrapolation of this scaling to a large enough number of logical qubits to handle problematic real-world
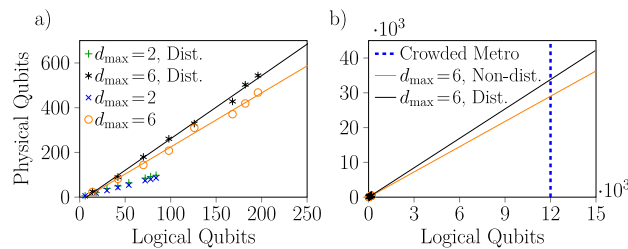
Fig. 2. Relationship between the number of logical qubits (variables) necessary to encode the scheduling problems considered in this work as QUBOs and the number of physical qubits required after embedding into the native coupling graph of the D-Wave annealer, for disturbed and non-disturbed instances with two choices for d_{max}. The trend for the problems we considered (**a**) is approximately linear, which can be extrapolated (**b**) to roughly estimate the number of qubits required to fit a crowded metro scheduling problem[7].

railway scheduling problems[7], which are around $60\times$ larger than the largest problem we consider in this work (in terms of $\#J\#S$, e.g. including around 13 stations and 150 trains).

For each of these 30 problems, solutions can be obtained with classical ILP solvers or through the solution of the corresponding QUBO problem on NISQ devices. In the latter case, the results will be stochastic due to the pervasive noise in current quantum computing platforms. Through the remainder of this section, we present details of the solution of these problems on a D-Wave quantum annealer and on a trapped-ion quantum computer by IonQ.

Quantum annealer
For the exploration of quantum annealing in this work, we have employed the D-Wave `Advantage_system6.3` machine with 5614 qubits and a configurable annealing time T (for hardware specifics, see the Supplementary Information). For each scheduling problem, we executed 25, 000 runs to build a large distribution of solutions. We did this for each of the 30 scheduling problems we defined, each run twice with annealing times of $10\mu s$ and $1000\mu s$ to study the effects on performance. The D-Wave embedding strategy uses chains of physical qubits to represent individual logical qubits. Here, we have used the default setting of *dwave.system*, the *EmbeddingComposite* Python library for all calculations on the D-Wave. The actual percentage of the number of solutions with no broken chains is greater than 50% for all calculations. We acknowledge, however, that properly setting chain strengths can have a positive impact on the annealer's performance[53], and assessing this problem is the sound direction of further investigation.

An example of the result distributions we obtained is presented in Fig. 3, which shows the distribution of solution energies produced for the 11 train scheduling problem broken down by annealing time and choice of penalty parameter values, with solutions categorized as feasible and infeasible (for the 12 train problems, certain parameter settings yielded no feasible solutions). The best results are obtained with a shorter annealing time and when the penalty values are chosen to give an overlapping spectrum as seen in Fig. 3a. The preference for a shorter annealing time is presumably a consequence of noise, but the improved performance when the spectrum is overlapping is a more non-trivial observation. From it, we conclude that for larger problems, when the annealing time is short it may be that penalty values which allow the feasible and non-feasible regions of the spectrum to overlap provide some computational advantage. A plausible intuitive explanation is given by considering that when the spectrum is split, if during the annealing process the system enters an excited state it may be locked in the excited and non-feasible region due to the spectral gap between the two regions.

To leverage these results practically for the problem of rail traffic scheduling, we concentrate on the statistics of passing times through the stochastic zone from feasible solutions. For the various 11 train scheduling problems, these statistics are presented in Fig. 4. In a later section, these distributions will be compared to the distribution of passing times measured from the actual Baltimore Light RailLink network. More generally, the findings presented in Fig. 4 qualitatively reproduce features of typical tramway delay frequency distributions due to disturbances as presented in Figs. 3, 4, 7, and 8 of[54]. Our results are also similar to the right tail of histograms of typical passing times of road-based public transportation as presented in Fig. 1a of[55].

As a final note, we observe that the shape of the result distributions is sensitive to the maximal delay parameter d_{max}. It is not just that the distribution widens, but the overall contour appears to develop additional features. This alongside the variations in the distributions due to different annealing times and penalty values indicates that it should be possible to exert a certain level of control over the shape of the passing time distribution by adjusting these parameters.

Gate-based computers
For the gate-based approach, we solved our scheduling problems with QAOA running on the trapped ion hardware and noisy simulator provided by IonQ. IonQ's "Aria-1" device provides 25 qubits with all-to-all connectivity, meaning that there is no embedding overhead when solving any given QUBO unlike with D-Wave's quantum annealers. We solved four non-disturbed scheduling problems with variable counts of 6, 10, 14, and 18 on the Aria-1 device and one disturbed scheduling problem with 18 variables. These problems required the execution of quantum circuits operating on 6 to 18 qubits. For details on these circuits and on the hardware properties,
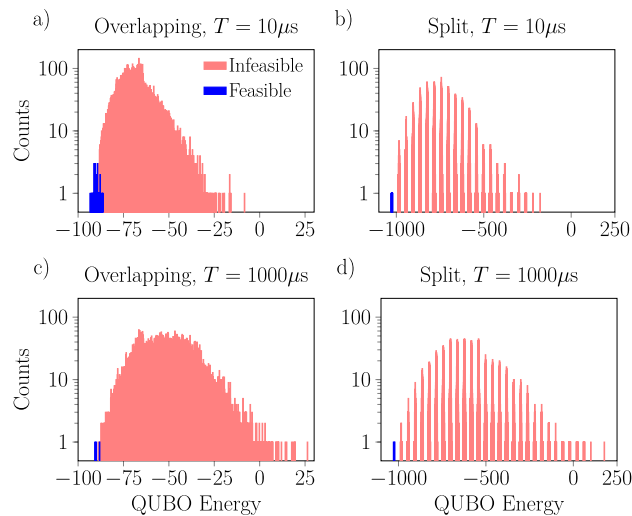
Fig. 3. Histograms of result energies measured with the D-Wave annealer showing the impact of the annealing time and penalty values on the QUBOs' energy spectrum. Results are shown for the disturbed instance of 11 trains with $d_{max} = 6$ requiring 182 logical and 503 physical qubits, with annealing times of $T = 10\mu s$ (**a,b**) and $T = 1000\mu s$ (**c,d**) and penalty values $p_{pair} = 2, p_{sum} = 4$ (**a,c**) and $p_{pair} = 20, p_{sum} = 40$ (**b,d**), which yield overlapping and split spectra respectively. Each histogram shows the results of 25,000 shots. Comparing the left and right panels makes the splitting of the spectrum due to the use of larger penalty values very clear. Comparing the top and bottom panels shows that in this case, a shorter annealing time typically yields better results.
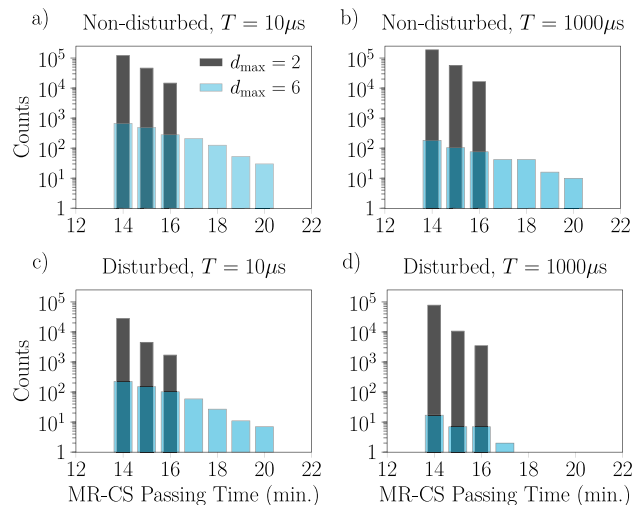
Fig. 4. Histograms showing the impact of the maximal secondary delay d_{max} and the annealing time T on passing times through the stochastic zone computed from feasible solutions generated by the D-Wave annealer for large instances with 11 trains (182 variables at $d_{max} = 6$), both disturbed (**a,b**) and non-disturbed (**c,d**). Annealing times T of $10\mu s$ (**a,c**) and $1000\mu s$ (**b,d**) are shown. In all cases, we use the penalty values $p_{sum} = 4$ and $p_{pair} = 2$ corresponding to an overlapping spectrum. Recall that the ILP versions of the problems were solved using CPLEX, returning a single optimal result for each instance, yielding the 14 min of MR-CS passing time; the CPLEX computational times of ILP problems were in the range of 0.0016–0.07 s.

see the Supplementary Information. Taking the largest 18 variable QUBO, the corresponding QAOA circuit with a single-layer ansatz contained 54 single-qubit gates and 30 two-qubit gates. In terms of circuit complexity, the most challenging QUBO to solve is actually the 14 variable instance, which has a more complicated set of constraints leading to the single-layer ansatz compiling to a circuit with 42 single-qubit gates and 63 two-qubit gates.

On the Aria-1 device, the average gate error rates can vary considerably over time, with single-qubit gate errors generally averaging between 0.03% and 0.06% and with two-qubit gate errors generally averaging between 2.1% and 8.6%. Further details on the characterization of the device are given in the Supplementary

Information. To understand the impact of gate errors on the device performance for our scheduling problems, we solved each QUBO problem using QAOA with both a one- and two-layer ansatz. In principle, the more complicated two-layer variational ansatz should allow the prepared state to better approximate the true ground state of the problem Hamiltonian and therefore should lead to better optimization performance. In practice, however, this theoretical improvement is offset by the increased circuit depth and so increased susceptibility to gate errors and decoherence. Here, for the largest 18 variable QUBO the two-layer QAOA circuit required 90 single- and 60 two-qubit gates. This is approximately double the number of the simpler circuit for the single-layer ansatz.

Due to constraints on hardware availability, we solved each of the four non-disturbed scheduling problems four times with QAOA on the Aria-1 hardware, with the two ansatze and two choices of penalty parameters. To better understand the results, we also solved each scheduling problem (both disturbed and non-disturbed) with QAOA run on the noisy simulator 50 times for each problem size and choice of parameters. Since we observe reasonable agreement between the results from the simulator and the device, we can use the simulator results as a proxy to understand the expected distribution of results with the gate-based approach.

The full set results from these experiments and simulations with a single-layer ansatz are presented in Fig. 5. Based on the fact that in each case the single experiment performed with the real hardware produced a result which was either optimal or close to optimal and which corresponded to an obvious peak in the histogram of results of the noisy simulations, we conclude that the noisy simulator does a reasonable job of capturing the behavior of the hardware and can therefore be used to make qualitative statements and predictions about its performance for train scheduling. Interestingly, the results we obtain from the noisy simulations show that the Aria-1 device performs better when the QUBO spectrum is split. This is opposite to the behavior of the D-Wave machine on the hardest instance solved as shown in Fig. 3.

Figure 6 shows the results of the experiments and simulations solving the largest 18 variable problem with a two-layer ansatz running on the Aria-1 device. The results are somewhat mixed, and we can not conclude which approach is better in this case.

For one specific two train scheduling problem with a disturbance requiring 18 variables, a comparison of the results obtained with the IonQ Aria-1 device with both ansatze and the results obtained with the D-Wave annealer is presented in Fig. 7 (for an additional point of comparison, we present some results from simulations of IBM's superconducting devices in the Supplementary Information). In this case, it is clear that the single-layer QAOA ansatz performs better than the two-layer ansatz, especially when the feasible and infeasible spectra overlap. An interesting qualitative observation is that while there is some similarity in the shapes of the histograms of result energies obtained with the different platforms, they are nonetheless visibly different. This is true for the different variational ansatze on the Aria-1 device as well, especially in the distribution of
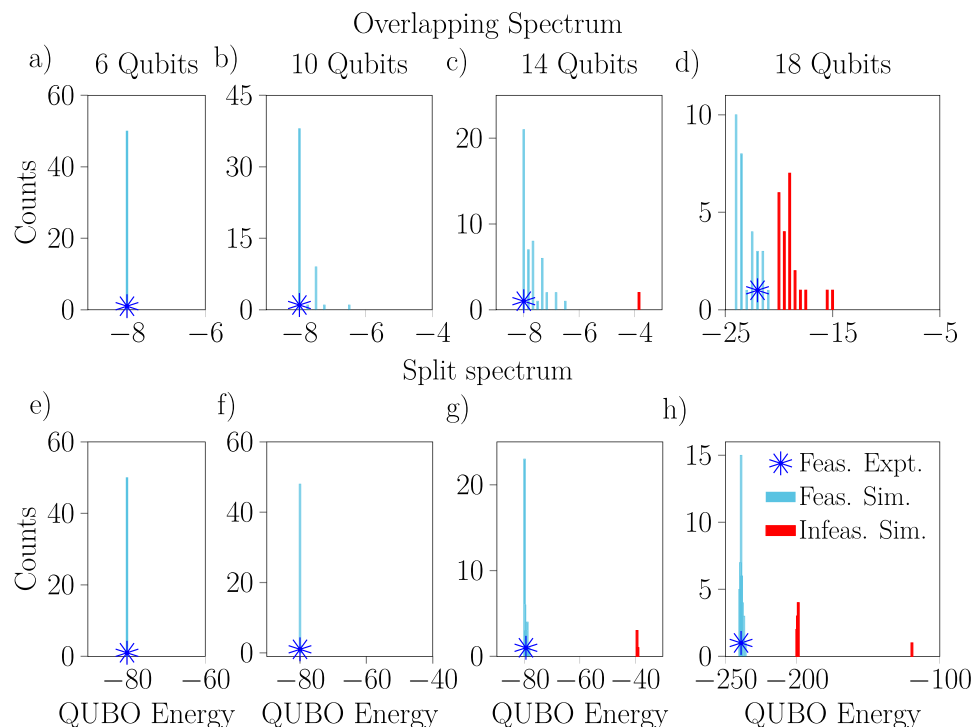

Fig. 5. Histograms of the result energy measured using QAOA with a single-layer ansatz to solve the QUBOs corresponding to small non-disturbed scheduling problems on IonQ's Aria-1 machine. The histograms show the results from noisy simulations, run 50 times per problem. Each QUBO was solved once using the real hardware, the resulting energies of these trials are shown with blue stars. From these histograms, we observe better results when the penalty parameters are such that the spectrum is split (bottom).
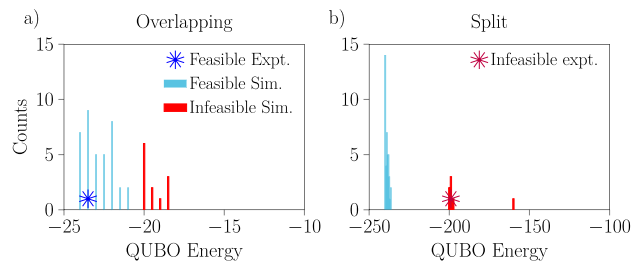
Fig. 6. Histograms of the result energy measured when solving the largest 18 variable QUBOs from a non-disturbed scheduling problem using QAOA with a two-layer ansatz on IonQ's Aria-1. As in Fig. 5, the histograms represent the results of 50 noisy simulations and the blue/red stars show the results of a single trial using the actual Aria-1 device. As before, we observe better results when the spectrum is split (right).
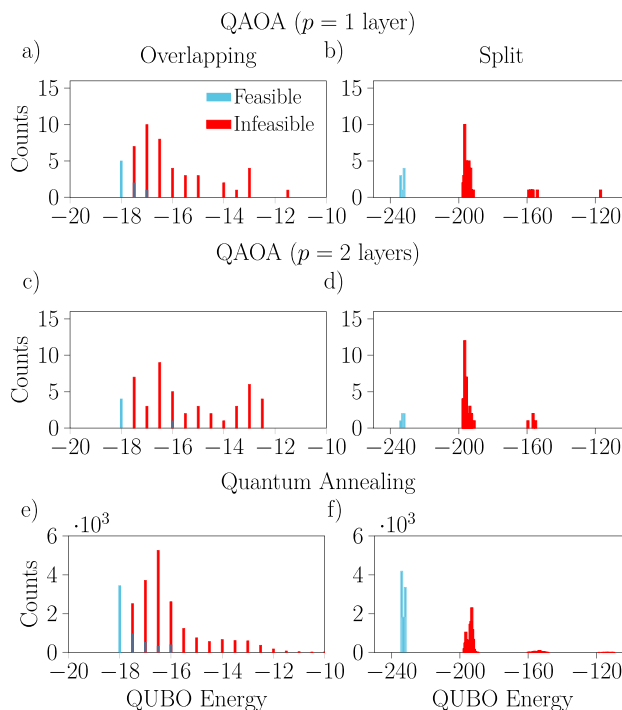

Fig. 7. Histograms of the measured energies for one particular disturbed 18 variable problem with 2 trains from noisy simulations of the IonQ Aria-1 device running QAOA with a single-layer (**a,b**) or two-layer (**c,d**) ansatz and from experiments with the D-Wave annealer with a 10 μs annealing time (**e,f**). Results are shown for two choices of penalty values leading to overlapping (**a,c,e**) or split (**b,d,f**) spectra. Notice that on the Aria-1 device we obtain better results with a single-layer ansatz than with a two-layer ansatz, especially when the penalties lead to an overlapping spectrum.

feasible solutions. To explain the potentially better performance in the cases with overlapping spectra, recall that high penalty values can reshape the low energy part of the optimization landscape. This potentially introduces gaps before good feasible solutions, and hence may trap the quantum optimization in higher energy states—see the Supplementary Information for further discussion including the entire spectra (split and overlapping) of selected QUBO problems.

Figure 8 continues the comparison between the results obtained with the D-Wave machine and the IonQ simulator running QAOA with the one-layer ansatz, now focusing on the properties of the solutions to the two train 18 variable railway scheduling problem, both with and without disturbances. Here, we show a comparison of the objective values (the tardiness) associated with the feasible solutions returned by each device, alongside the passing times from MR $\rightarrow$ CS encoded into each solution. We observe that the distribution of passing times produced by the two approaches are very similar, despite the distributions of objective values being quite different.
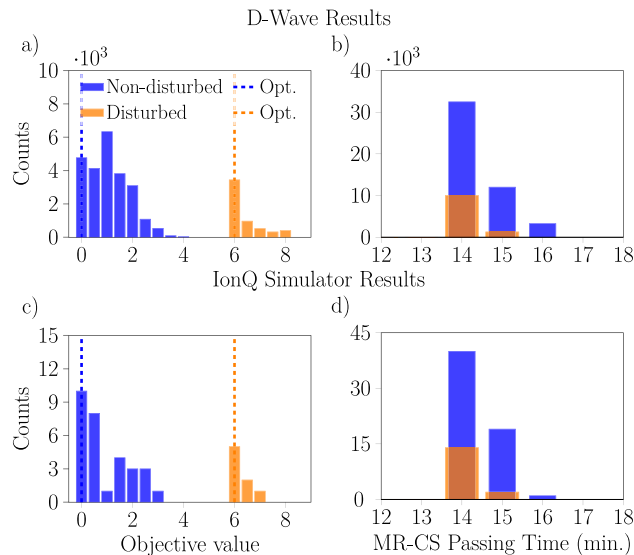
Fig. 8. Histograms of the objective values and of the MR-CS passing times from schedules obtained from the D-Wave annealer with a 10 μs annealing time (**a,b**) and from noisy simulations of the IonQ Aria-1 device running QAOA with a single-layer ansatz (**c,d**). In both cases, the penalty values $p_{sum} = 4$ and $p_{pair} = 2$ were used, and only feasible solutions fulfilling all constraints from "QUBO approach for quantum computing" have been counted. Both disturbed and non-disturbed problems are shown, with the optimal objective values indicated with dashed lines. Recall that the ILP versions of the problems were solved using CPLEX, returning optimal results (single for each instance) with an objective value of 0 and 6 (and 14 min of MR-CS passing time) in computational time of 0.0016 s.
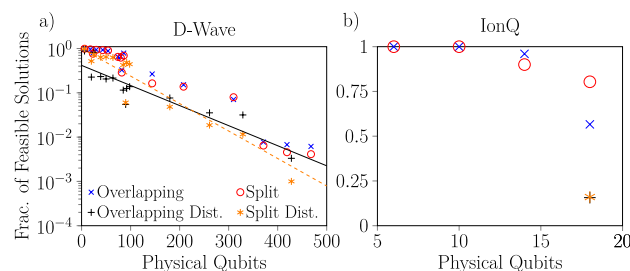
Fig. 9. Evolution of the fraction of solutions to scheduling problems which are feasible as a function of the problem size, measured by the number of physical qubits required to embed the associated QUBOs on each platform. For the D-Wave annealer (**a**), the results are taken from experiments with an annealing time of $T = 10$ μs. Fits are shown for the disturbed problems with penalty values corresponding to both the overlapping and split cases, which shows that the fraction of feasible solutions decreases approximately exponentially with increasing problem size. For the IonQ Aria-1 device (**b**), the results are taken from noisy simulations of QAOA run with a single-layer ansatz. Note that there is no embedding overhead on Aria-1, each variable is directly associated with one physical qubit.

Scaling

An important consideration for any proposed application of NISQ devices is scaling behavior, as that will indicate the size of problem instances that can be handled at present and inform what problems are likely to be accessible in the near future. To this end, in Fig. 9 we plot the fraction of schedules returned as solutions to our railway scheduling problems which were feasible (i.e., those which fulfill all constraints listed in "QUBO approach for quantum computing") as a function of the number of physical qubits required for their solution. With both IonQ's gate-based simulator and D-Wave's quantum annealer, we unsurprisingly observe that larger problems are more likely to yield purported solutions which violated one or more constraints. For the D-Wave device, we have sufficient statistics to infer that the fraction of feasible solutions decreases exponentially with the number of physical qubits necessary to embed the problem. Bear in mind that the translation of the scheduling problem to QUBO significantly increases the number of variables due to the time discretization (Eq. (10)), which is not necessary in the original ILP formulation. To reflect this in practice, we have solved the deterministic ILP problem with CPLEX (Python API version: 22.1.1.0 on the local CPU). In our case, the CPLEX solution was always optimal and the CPLEX computational time was short (0.0014 s to 0.08 s), pinpointing the better

performance of the ILP approach see https://github.com/iitis/quantum-stochastic-optimization-railways/blob/master/solutions/cplex_benchmarks.json for particular CPLEX results. However, our goal was to asses the accessibility of various NISQ devices to model actual railway stochastic system rather than to compare runtimes in finding optimal solutions.

Naively extrapolating to the sizes necessary to solve problems of practical significance in railway scheduling (e.g., scheduling a crowded metro[7] which is roughly $60\times$ larger than the largest problem solved in this work) indicates that such problems will not be accessible even to large quantum devices without some other improvements. Nevertheless, there exist approaches such as that of multilevel combinatorial optimization[27] based on hybrid computation which can allow the solution of problems an order of magnitude larger than current quantum devices. To this end, we examine the statistical distribution of the results produced by our experiments with a goal of using quantum computing as one component of a hybrid algorithm for railway scheduling.

Real-world railway rescheduling perspective

To best connect our results with the practical perspective of railway scheduling and operation, we focus on the large 11 train disturbed scheduling instance. The associated 182 variable QUBO can be solved on the D-Wave annealer (requiring 503 physical qubits) but is too large for current gate-based quantum computers.

For this particular problem, the input disturbed timetable includes conflicts between three pairs of trains. A train diagram representing this timetable is shown in Fig. 10a. To resolve this conflict, it is necessary that some of the trains involved in conflicts add an additional wait to their schedule. Figure 10b shows an optimal rescheduling that resolves these conflicts, generated with a classical ILP solver. Notice that passing times of all the trains are the same (i.e., the slopes of the lines are identical). This is not what would be expected for a timetable describing the movement of trains through stochastic areas of the rail network; in such a case, the passing times should be randomly perturbed and so the slopes should vary. This feature is visible in Fig. 10c, which depict timetables corresponding to the best (lowest energy) solution returned by the D-Wave annealer and a solutions corresponding to a low-lying excited state, respectively. This observation supports the assertion that when NISQ devices are used to solve scheduling problems with this type of QUBO representation, their inherent noise may manifest in a way that can capture stochastic behavior of the underlying real-world model. (Finally, Fig. 10d depicts one example of a highly-excited yet still feasible solution yielded by the annealer).

To elaborate this observation, we compare the statistics of the D-Wave output to the statistics measured from real-world rail traffic. For this comparison, we present in Fig. 11 the distribution of passing times northbound $MR \rightarrow CS$ and southbound $CS \rightarrow MR$. The data[56] was collected by aggregating passing times at peak hour during work days, as that is when the random disturbances due to road traffic are most intense.

We observe that there is some qualitative similarity between the right tails of these distrubutions and the results from the D-Wave annealer as were shown in Fig. 4. The most common passing time in the real data is 12 min compared to 14 min in the D-Wave results, though this difference is due to specifics of Baltimore Light RailLink timetable and is also potentially influenced by the details of how the model was constructed and of how the real passing time is measured. In the real data we observe that some trains are recorded as going faster than the Baltimore Light RailLink timetable specifies. This is likely due to the inclusion of some extra reserve time in the Baltimore Light RailLink timetable.
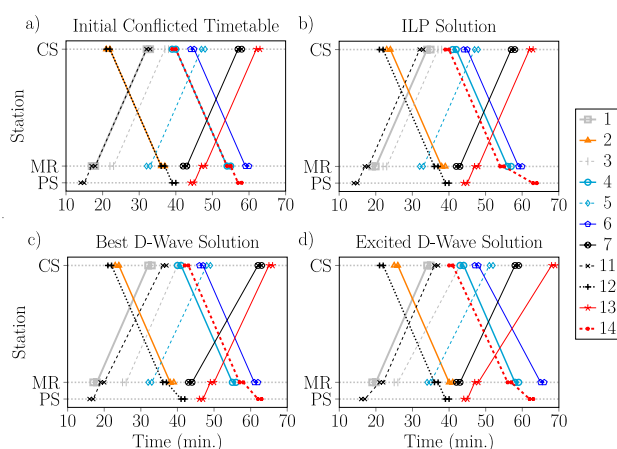

Fig. 10. Train timetable diagrams illustrating an input disturbed timetable for an 11 train schedule with conflicts (**a**) alongside several possible new solution timetables which resolve the conflict. Each line segment represents a train moving from station to station, and trains are numbered according to standard practice in railway scheduling. Shown is an initial conflicted timetable (**a**), an optimal solution found using ILP (**b**), the best solution obtained with the D-Wave annealer (**c**), and one example of a highly-excited yet still feasible solution yielded by the annealer (**d**). For the D-Wave experiments, the annealing time was 10 μs and the penalty values were $p_{sum} = 4$, $p_{pair} = 2$ yielding an overlapping spectrum.
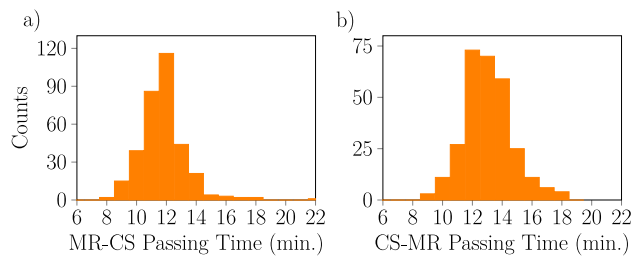
Fig. 11. Histogram of actual passing times between Mount Royal (MR) and Camden Station (CS) for Baltimore Light RailLink trains in morning peak hours (7 a.m.–10 a.m.) and afternoon peak hours (3 p.m.–6 p.m.). Data for (**a**) northbound and (**b**) southbound trains are displayed separately. Data were collected for all workdays from 11 through 31 January 2024, excluding the 17th, 18th, and 26th due to Baltimore Light RailLink issues.
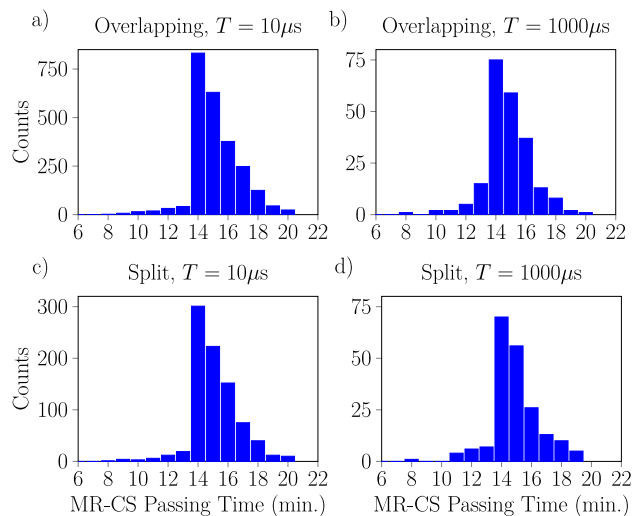

Fig. 12. Histogram of MR-CS passing times obtained by solving the disturbed 11 train problem with the D-Wave annealer. For this plot, the passing time constraint of Eq. (7) was *not* enforced when checking if a given solution was feasible. Results are shown for annealing times T of 10 μs (**a,c**) or 1000 μs (**b,d**), and with penalty values set to $p_{pair} = 2$ and $p_{pair} = 4$ to produce an overlapping spectrum (**a,b**) or to $p_{pair} = 20$ and $p_{pair} = 40$ to produce a split spectrum (**c,d**). If we compare these distributions to those measured from actual rail traffic shown in Fig. 11, we see that while there are differences there are also qualitative similarities (e.g., in the shape of the right tail of the measured southbound distribution vs. the annealing results).

  To better reflect this behavior and so to capture the left tail of the real-world distributions, we alter our model by softening the constraints slightly. The transformation of the scheduling problem into QUBO form remains unaltered and the minimization proceeds as normal including all penalty terms, but at the end, we do not check any **passing time constraints** in Eq. (7) when determining the feasibility of a solution. The distributions of passing times extracted from optimizations of the same 11 train scheduling problem with the D-Wave machine following this relaxed proscription are shown in Fig. 12. Note that the presence of solutions with passing times less than 14 minutes from solutions which violate passing time constraints but respect the remaining constraints. These solutions add a left tail to the distributions which brings them closer to the measured transit times shown in Fig. 11, especially with higher annealing times. We expect that further tuning of the various parameters (especially the annealing time) can bring these distributions even closer together, and it is these similarities are the foundation for the modeling of stochastic rail traffic with NISQ devices.

  As a final note, while we focus primarily on scheduling of train networks more specifically on scheduling the Baltimore Light RailLink network in this work, similar approaches can be followed for other types of transportation systems which will yield similar results. Distributions of typical passing times similar to those we present have been observed in road-based public transport, for example as shown in Fig. 1A of[55]. Moving forward, the techniques and approaches to rail scheduling discussed in this work to these and other types of public transportation networks and indeed to any analogous scheduling problems.

Quantum device	Total execution time	Total cost (USD)
D-Wave Advantage_system6.3	~ 30 min	$\approx \$1000$
IonQ Aria-1	~ 21 h	$\approx \$67,000$

Table 1. Execution time and costs expended on each quantum platform for the computational results presented in this paper. The costs reported in the table are those incurred by us. The actual costs may differ for other users due to individual pricing policies of the companies. Bear in mind, however, that a classical CPU can solve the ILP version of the problem to optimality more efficiently than the quantum devices considered and at significantly lower cost.

Assessing costs of quantum computing

To conclude our analysis, we assess the costs of the application of different paradigms of quantum computing for the real problem of Baltimore Light RailLink scheduling, both in terms of the compute time required to solve our scheduling problems and in terms of the monetary expense necessary to obtain access to the hardware.

On the D-Wave machine, we ran each problem instance twice, once with 25,000 shots and an annealing time of 10 μs and once with the same number of shots and a longer annealing time of 1000 μs. Combined, each problem instance consumed just over 25 s of compute time. Adding together the time taken to solve 30 different scheduling problems—and hence minimize 60 different QUBOs (split and overlapping spectrum for each)—in this way comes out to about 30 min of time required, including overhead.

As for our results from the IonQ Aria-1 device, the iterative optimization used in QAOA required the running of $\sim 30 - -50$ circuits to evaluate the objective function for each QUBO solved. Unlike with the D-Wave machine, the time required for the circuits corresponding to the QAOA ansatz for different problems varies depending on the details of the scheduling problem, as the circuit depth varies. The runtime is dominated by two-qubit gates, which on the Aria-1 device take 600 μs each. Most of the circuits run in the course of this work took on the order of 1–2 s to run 1024 shots, and each full optimization with QAOA required 40–60 min.

With IonQ's hardware, the expense of running a given circuit depends on the complexity of that circuit. For every optimization we ran but one, the circuits used were simple enough that they cost the minimum 97.50 USD per run on the read device per circuit (with error mitigation enabled). The most complicated circuit (solving the 14 variable problem with the two-layer ansatz, 70 single-qubit and 126 two-qubit gates) cost 141.57 USD per run.

We present a summary of the runtime requirements and approximate cost of the computations performed in this study in Table 1.

Conclusions

At present, quantum computing is in the era of NISQ devices, which are inherently noisy in an intrusive and unavoidable way. For this reason, it is important to incorporate some understanding of this noise at every level when attempting to build solutions for practical problems using these devices. Henceforth, leveraging quantum noise as a computational resource for optimization, rather than computational efficiency, was the primary goal of this work. In this work, we have demonstrated the use of current quantum computers in two different paradigms to study the dynamics of real-world rail or tramway operations. In other words, we have demonstrated the practical use of NISQ technology in solving real-world problems, particularly in optimizing transportation systems under stochastic disturbances. More specifically, we studied the problem of (re-)scheduling a set of trains which pass through portions of a rail network that have some degree of inherent stochasticity, e.g. due to sharing infrastructure with road traffic. By focusing on this inherently noisy optimization problem, we sought to actively exploit the noise present in all current hardware as a strength. Such research encourages collaboration across various fields, promotes innovation and new methodologies in both theoretical and applied research.

We focused on scheduling trains in a model of a portion of the Baltimore LightRail Link network, a tramway in Baltimore, Maryland, where between some stations trains must travel on tracks which share space with roadways leading to randomness in transit times. Hereby, our research potentially improves the efficiency and reliability of public transportation systems, benefiting commuters and urban infrastructure. We constructed a variety of test scheduling problems starting from an actual timetable for the tramway, with the smallest problem having only a single train and the largest having twelve. These scheduling problems were easily translated to QUBOs, at which point we solved them using two types of NISQ devices. The first device employed was a quantum annealer, the D-Wave `Advantage_system6.3`, which has 5614 qubits and could easily fit all of our scheduling problems. The second device was the IonQ Aria-1 gate-based trapped-ion computer, a 25-qubit device which was capable of solving several of the smaller scheduling problems we constructed using QAOA. While current gate-based computers have significantly fewer qubits than quantum annealers and while QAOA is a much more expensive approach to solving QUBOs than annealing, there is significant merit in the comparison. By testing our approach on two devices with completely different computational approaches, physical architectures, and noise models we can both demonstrate the broad applicability of these techniques as well as demonstrate how variations in the implementation alter the statistics of the output.

Our results show that when optimizing the schedules which govern trains passing through a stochastic zone, the effects of the random noise produce a set of schedules which when aggregated produce distributions of passing times which can potentially mimic the unpredictable delays visible in measurements of real railway traffic. Consequently, this study can serve as a starting point for the development of specialized hybrid quantum/

classical algorithms that integrate NISQ solvers alongside classical approaches to aid in capturing and optimizing the stochastic behavior of railway networks. This is an especially important point as large and practically-relevant railway scheduling problems[7] are approximately $60\times$ larger than the largest problem we have successfully solved herein. By splitting the computation between a NISQ device which handles the noisy optimization component and a classical computer which handles the larger deterministic sections of the network which encompass and connect the stochastic zone(s), it will be possible to exploit the advantages of NISQ devices while avoiding limitations due to their small size. The development of effective error correction strategies, such as the one assessing the D-Wave embedding strategy, or error mitigation in variational quantum algorithms such as QAOA [57] is a sound avenue for future work. Other future directions of work include the adoption of gate-based quantum optimization methods that propose better scalability in terms of the number of qubits[44,45] or that are are not based on QUBO formulations[46].

Finally, in demonstrating the potential for NISQ devices in the realm of stochastic scheduling we have taken an important step towards a resolution of several ongoing debates within the operational research community regarding the effectiveness of QUBO formulations such as the one we present in "QUBO approach for quantum computing" for the planning and re-planning of railway schedules, and for more general practical applications [5] or similar problem formulations with more general QUBOs [32]. To this end, our research opens avenues for further research in quantum computing applications, particularly in complex optimization scenarios and real-time system management.

Data availability

The code and data used in this work have been made publicly available at https://github.com/iitis/quantum-stochastic-optimization-railways. Data from Baltimore Light RailLink has been collected on https://github.com/iitis/Baltimore-Light-RailLink-data.

References

1. Pirnay, N., Ulitzsch, V., Wilde, F., Eisert, J. & Seifert, J.-P. An in-principle super-polynomial quantum advantage for approximating combinatorial optimization problems. *arXiv preprint* arXiv:2212.08678 (2023).
2. DiVincenzo, D. P. The physical implementation of quantum computation. *Fortschritte der Physik: Progr. Phys.* **48**, 771–783. https://doi.org/10.48550/arXiv.quant-ph/0002077 (2000).
3. Sanders, B. C. *How to Build a Quantum Computer*, 2399–2891 (IOP Publishing, 2017).
4. Preskill, J. Quantum computing in the nisq era and beyond. *Quantum*. **2**, 79. https://doi.org/10.22331/q-2018-08-06-79 (2018).
5. Katzgraber, H. Searching for applications of quantum computing in industry. *Bull. Am. Phys. Soc.* (2024). https://meetings.aps.org/Meeting/MAR24/Session/Y49.1.
6. Bešinović, N. Resilience in railway transport systems: a literature review and research agenda. *Transp. Rev.* **40**, 457–478. https://doi.org/10.1080/01441647.2020.1728419 (2020).
7. Zhou, H., Qi, J., Yang, L., Shi, J. & Mo, P. Joint optimization of train scheduling and rolling stock circulation planning with passenger flow control on tidal overcrowded metro lines. *Transp. Res. Part C Emerg. Technol.* **140**, 103708. https://doi.org/10.1016/j.trc.2022.103708 (2022).
8. Ge, L., Voß, S. & Xie, L. Robustness and disturbances in public transport. *Public Transp.* 1–71. https://doi.org/10.1007/s12469-022-00301-8 (2022).
9. Domino, K. et al. Quantum annealing in the nisq era: railway conflict management. *Entropy* **25**, 191. https://doi.org/10.3390/e25020191 (2023).
10. Domino, K., Kundu, A., Salehi, Ö. & Krawiec, K. Quadratic and higher-order unconstrained binary optimization of railway rescheduling for quantum computing. *Quantum Inf. Process.* **21**, 337. https://doi.org/10.1007/s11128-022-03670-y (2022).
11. Xu, H.-Z. et al. High-speed train timetable optimization based on space-time network model and quantum simulator. *Quantum Inf. Process.* **22**, 418. https://doi.org/10.1007/s11128-023-04170-3 (2023).
12. Grozea, C., Hans, R., Koch, M., Riehn, C. & Wolf, A. Optimising rolling stock planning including maintenance with constraint programming and quantum annealing. *arXiv preprint* arXiv:2109.07212 (2021).
13. Grange, C. *Design and application of quantum algorithms for railway optimisation problems*. Ph.D. thesis, Université de Montpellier (2024). https://hal.science/tel-04671228v2.
14. Grange, C., Pozzoli, V., Ramond, F. & de Almeida, D. Formulation and quantum resolution of a railway timetabling problem. In *World Congress on Railway Research*. https://hal.science/hal-04592820v1 (2022).
15. Callison, A. & Chancellor, N. Hybrid quantum-classical algorithms in the noisy intermediate-scale quantum era and beyond. *Phys. Rev. A* **106**, 010101. https://doi.org/10.1103/PhysRevA.106.010101 (2022).
16. Dixit, V. V., Niu, C., Rey, D., Waller, S. T. & Levin, M. W. Quantum computing to solve scenario-based stochastic time-dependent shortest path routing. *Transport. Lett.* 1–11. https://doi.org/10.1080/19427867.2023.2238461 (2023).
17. Zeng, A. Z., Durach, C. F. & Fang, Y. Collaboration decisions on disruption recovery service in urban public tram systems. *Transport. Res. Part E Logist. Transport. Rev.* **48**, 578–590. https://doi.org/10.1016/j.tre.2011.11.005 (2012).
18. Lai, D. S. & Leung, J. M. Real-time rescheduling and disruption management for public transit. *Transportmetr. B Transp. Dyn.* **6**, 17–33. https://doi.org/10.1080/21680566.2017.1358678 (2018).
19. Kiefer, A., Schilde, M. & Doerner, K. F. Scheduling of maintenance work of a large-scale tramway network. *Eur. J. Oper. Res.* **270**, 1158–1170. https://doi.org/10.1016/j.ejor.2018.04.027 (2018).
20. Malucelli, F. & Tresoldi, E. Delay and disruption management in local public transportation via real-time vehicle and crew re-scheduling: a case study. *Public Transp.* **11**, 1–25. https://doi.org/10.1007/s12469-019-00196-y (2019).
21. Koniorczyk, M., Krawiec, K., Botelho, L., Bešinović, N. & Domino, K. Application of a hybrid algorithm based on quantum annealing to solve a metropolitan scale railway dispatching problem. *J. Rail Transp. Plann. Manag.* **34**, https://doi.org/10.1016/j.jrtpm.2025.100521 (2025).
22. Śmierzchalski, T. *et al.* Hybrid quantum-classical computation for automatic guided vehicles scheduling. *Sci. Rep.* **14**, 21809. https://www.nature.com/articles/s41598-024-72101-y (2024).
23. Light raillink. https://www.mta.maryland.gov/schedule/lightrail (accessed 13 Dec 2023).
24. Barry, K. A. *The Economic Development Impacts of Light Rail Transit: A Case Study of the Baltimore Central Light Rail* (Georgetown University, 2012).
25. MTA performance improvement. https://www.mta.maryland.gov/performance-improvement (accessed 13 Dec 2023).

26. Kraśkiewicz, C. & Oleksiewicz, W. Tramwaj dwusystemowy w Karlsruhe eng. Dual-system tram in Karlsruhe. *Logistyka*. **4** (2015).
27. Ushijima-Mwesigwa, H. et al. Multilevel combinatorial optimization across quantum architectures. *ACM Trans. Quantum Comput.* **2**, 1–29. https://doi.org/10.1145/3425607 (2021).
28. Map of the Baltimore Light RailLink system from OpenRailwayMap. https://www.openrailwaymap.org/?style=standard &lat=39.29687852644235 &lon=-76.61566972732543 &zoom=15. (accessed 05 Jun 2025).
29. Kroon, L., Maróti, G., Helmrich, M. R., Vromans, M. & Dekker, R. Stochastic improvement of cyclic railway timetables. *Transport. Res. Part B Methodol.* **42**, 553–570. https://doi.org/10.1016/j.trb.2007.11.002 (2008).
30. D'Ariano, A., Pacciarelli, D. & Pranzo, M. A branch and bound algorithm for scheduling trains in a railway network. *Eur. J. Oper. Res.* **183**, 643–657. https://doi.org/10.1016/j.ejor.2006.10.034 (2007).
31. Pinedo, M. L. *Scheduling*, vol. 29 (Springer, 2012).
32. Glover, F., Kochenberger, G. & Du, Y. Quantum bridge analytics i: a tutorial on formulating and using qubo models. *4OR.* **17**, 335–371. https://doi.org/10.1007/s10288-019-00424-y (2019).
33. Venturelli, D., Marchand, D. J. & Rojo, G. Quantum annealing implementation of job-shop scheduling. *arXiv preprint* arXiv:1506.08479 (2015).
34. Alessandroni, E., Ramos, S., Roth, I., Traversi, E. & Aolita, L. Alleviating the quantum big-m problem. *arXiv preprint* arXiv:2307.10379 (2023).
35. Bittel, L. & Kliesch, M. Training variational quantum algorithms is np-hard. *Phys. Rev. Lett.* **127**, 120502. https://doi.org/10.1103/PhysRevLett.127.120502 (2021).
36. Liu, X. et al. Leveraging special-purpose hardware for local search heuristics. *Comput. Optim. Appl.* **82**, 1–29. https://doi.org/10.1007/s10589-022-00354-2 (2022).
37. Avron, J. E. & Elgart, A. Adiabatic theorem without a gap condition. *Commun. Math. Phys.* **203**, 445–463. https://doi.org/10.1007/s002200050620 (1999).
38. Dattani, N., Szalay, S. & Chancellor, N. Pegasus: The second connectivity graph for large-scale quantum annealing hardware. *arXiv preprint* arXiv:1901.07636 (2019).
39. Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644. https://doi.org/10.1038/s42254-021-00348-9 (2021).
40. Blekos, K. et al. A review on quantum approximate optimization algorithm and its variants. *Phys. Rep.* **1068**, 1–66. https://doi.org/10.1016/j.physrep.2024.03.002 (2024).
41. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. *arXiv preprint* arXiv:1411.4028 (2014).
42. Powell, M. J. D. *A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation, 51–67* (Springer, Netherlands, 1994).
43. Fernández-Pendás, M., Combarro, E. F., Vallecorsa, S., Ranilla, J. & Rúa, I. F. A study of the performance of classical minimizers in the quantum approximate optimization algorithm. *J. Comput. Appl. Math.* **404**, 113388. https://doi.org/10.1016/j.cam.2021.113388 (2022).
44. Tan, B., Lemonde, M.-A., Thanasilp, S., Tangpanitanon, J. & Angelakis, D. G. Qubit-efficient encoding schemes for binary optimisation problems. *Quantum.* **5**, 454. https://doi.org/10.22331/q-2021-05-04-454 (2021).
45. Tene-Cohen, Y., Kelman, T., Lev, O. & Makmal, A. A variational qubit-efficient maxcut heuristic algorithm. *arXiv preprint* arXiv:2308.10383 (2023).
46. Sciorilli, M. et al. Towards large-scale quantum optimization solvers with few qubits. *Nat. Commun.* **16**, 476. https://doi.org/10.1038/s41467-024-55346-z (2025).
47. Glos, A., Krawiec, A. & Zimborás, Z. Space-efficient binary optimization for variational computing. *arXiv preprint* arXiv:2009.07309 (2020).
48. Stein, J. *et al.* Evidence that pubo outperforms qubo when solving continuous optimization problems with the qaoa. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, GECCO '23 Companion, 2254–2262. https://doi.org/10.1145/3583133.3596358 (ACM, 2023).
49. Montanaro, A. Quantum speedup of branch-and-bound algorithms. *Phys. Rev. Res.* **2**, 013056. https://doi.org/10.1103/PhysRevResearch.2.013056 (2020).
50. Chakrabarti, S., Minssen, P., Yalovetzky, R. & Pistoia, M. Universal quantum speedup for branch-and-bound, branch-and-cut, and tree-search algorithms. *arXiv preprint* arXiv:2210.03210 (2022).
51. MTA schedule. https://www.mta.maryland.gov/schedule/lightrail. (accessed 01 Dec 2023).
52. MDOT MTA transit information. https://www.mta.maryland.gov/schedule/stops/lightrail (accessed 01 Dec 2023).
53. King, A. D., Lanting, T. & Harris, R. Performance of a quantum annealer on range-limited constraint satisfaction problems. *arXiv preprint* arXiv:1502.02098 (2015).
54. Ullrich, O., Lückerath, D. & Speckenmeyer, E. Do regular timetables help to reduce delays in tram networks? it depends!. *Public Transp.* **8**, 39–56. https://doi.org/10.1007/s12469-015-0115-6 (2016).
55. Büchel, B. & Corman, F. Review on statistical modeling of travel time variability for road-based public transport. *Front. Built Environ.* **6**, 70. https://doi.org/10.3389/fbuil.2020.00070 (2020).
56. Baltimore Light RailLink passing time data collected with *Swiftly's GTFS-realtime API* (https://swiftly.zendesk.com/hc/en-us) and covers workdays from 11.1.24 - 31.1.24, excluding the 17th, 18th, and 26th due to issues with the Baltimore Light RailLink system.
57. Botelho, L. et al. Error mitigation for variational quantum algorithms through mid-circuit measurements. *Phys. Rev. A* **105**, 022441. https://doi.org/10.1103/PhysRevA.105.022441 (2022).

Acknowledgements

Author contributions

K.D., S.D.: conceptualization, K.D., E.D., R.R. B.G.: preparing experiments, K.D., E.D., R.R.: running experiments, K.D., E.D., R.R., B.G., S.D.: data analysis, K.D., E.D. S.D.: manuscript writing, K.D., E.D., B.G., S.D.: manuscript supervision, S.D., B.G.: funding acquisition. All authors reviewed the manuscript.

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-025-15545-0.

Correspondence and requests for materials should be addressed to K.D. or E.D.

Reprints and permissions information is available at www.nature.com/reprints.

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