Quantum Information Processing (2023) 22:418
https://doi.org/10.1007/s11128-023-04170-3

High-speed train timetable optimization based on space–time network model and quantum simulator

Hui-Zhang Xu[1] · Jun-Hua Chen[1] · Xing-Chen Zhang[1] · Te-Er Lu[1] · Tian-Ze Gao[1] · Kai Wen[2] · Yin Ma[2]

Received: 10 June 2023 / Accepted: 30 October 2023

Abstract
Timetable scheduling is a combinatorial optimization problem that presents formidable challenges for classical computers. This paper introduces a pioneering methodology for addressing the high-speed train timetabling problem through quantum computing. Initially, a comprehensive binary integer programming model, grounded in the space–time network, is proposed (M1). To manage the intricacy of model M1, a knapsack problem reformulation is employed to establish a simplified binary integer programming model (M2). Both M1 and M2 are subsequently converted into quadratic unconstrained binary optimization (QUBO) models to harness the potential of quantum computing. Several techniques, including the Gurobi solver, simulated annealing, and the coherent Ising machine (CIM) quantum simulator, are deployed to solve the model across four distinct scenarios of varying complexity. The findings

✉ Jun-Hua Chen
cjh@bjtu.edu.cn

Hui-Zhang Xu
19114059@bjtu.edu.cn

Xing-Chen Zhang
xczhang@bjtu.edu.cn

Te-Er Lu
teerlu@bjtu.edu.cn

Tian-Ze Gao
tianzegao@bjtu.edu.cn

Kai Wen
wenk@boseq.com

Yin Ma
may@boseq.com

[1] School of Traffic and Transportation, Beijing Jiaotong University, Beixiaguan, Haidian, Beijing 100044, China

[2] Beijing QBoson Quantum Technology Co., Ltd., Jiuxianqiao, Chaoyang, Beijing 100015, China

Published online: 22 November 2023

indicate that CIM quantum simulator outperforms the simulated annealing method in terms of solution quality for medium-scale problems.

Keywords High-speed railway · Timetable scheduling · Quantum computing · Quadratic unconstrained binary optimization · Space–time network

1 Introduction

Timetable scheduling is a combinatorial optimization problem that poses significant difficulties for classical computers. Due to the large scale and complexity of such NP-hard problems, classical algorithms typically either require unrealistic running times or fail to produce optimal solutions. In light of these limitations, quantum computing offers an attractive alternative for more efficient solutions to combinatorial optimization problems.

The concept of quantum computing and its related ideas can be traced back to 1980 when physicist Paul Benioff proposed a quantum Hamiltonian model of the Turing machine [1], and mathematician Yuri Manin pointed out the potential of quantum computers to surpass their classical counterparts. In 1986, Feynman introduced the prototype concept of quantum circuits [2]. The development of Shor's quantum algorithm for integer factorization (later known as the "Shor's algorithm") in 1994 [3], followed by the Grover search algorithm [4], represented important progress in quantum algorithms and computing research. Based on categorization criteria proposed by Peng [5] and Cooper [6], existing quantum algorithms can be classified into two primary categories, as illustrated in Fig. 1: algorithms that run on quantum computers and algorithms that use the principles of quantum mechanics but run on classical computers [7–10].

In the category of algorithms that run on quantum computers, there are quantum annealing algorithms as well as algorithms that operate under the framework of CIM (coherent Ising machine) [11]. Quantum annealing algorithm [12–15] is implemented on quantum annealing computers, with one of the most popular and widely used ones

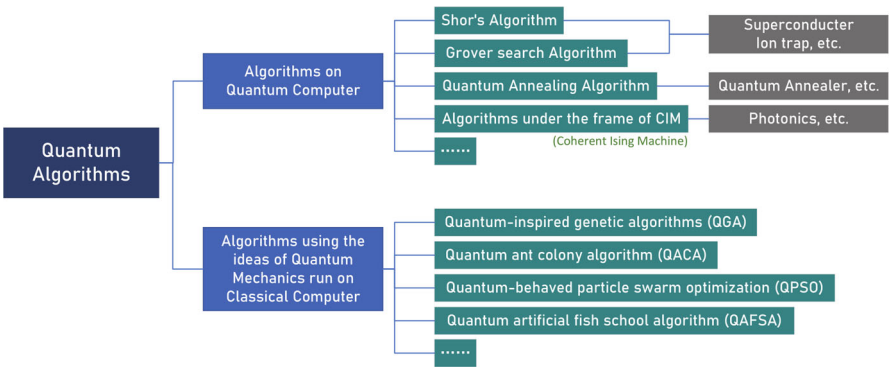


Fig. 1 Category of quantum algorithms

designed by D-Wave [16–19]. Algorithms running on CIM [20–22]. The CIM based on photonic quantum systems exhibits technical advantages such as room temperature operation, coherent optical encoding, and full connectivity [23, 24]. These two algorithms are generally considered to be well-suited and efficient for solving optimization problems [25]. Additionally, it is worth noting that while quantum computers are not yet highly mature, quantum simulators can be used to address these problems. The CIM simulator is a software implementation of the physical CIM system, designed to simulate and evaluate the performance and effectiveness of CIM in solving various problems. The CIM simulator utilizes numerical methods to simulate the operation process of CIM, providing a convenient means to explore and study optimization and computing tasks in different problem domains. In this research, the method of utilizing quantum simulators is employed to solve the problem of train timetable scheduling.

In recent years, there is a line of research applying quantum computing to the train scheduling problems. Domino et al. [26] introduced a quadratic unconstrained binary optimization (QUBO) model for the railway dispatching problem and solve the real-life case from the Polish railway network using D-Wave quantum annealers. Two years later, Domino et al. [27] proposed higher-order binary optimization (HOBO) formulation to use quantum annealing solving the railway rescheduling problem. The proof of concept implementation was demonstrated on the D-Wave quantum processing unit and D-Wave hybrid solver. Grozea et al. [28] focused on rolling stock optimization including maintenance tasks and compared constraint programming with quantum annealing methods.

It is observed that the complicated constraints that are hard to deal with always exist in train scheduling problems. To address this issue, in paper [29], for vehicle routing problem with pickup and delivery services with time windows (VRPPDTW), the state–space–time networks were constructed, and then a multi-commodity network flow programing model was established. The presence of multi-dimensional decision variables posed computational challenges when dealing with large-scale real-world data sets. Therefore, the problem was reformulated by Lagrangian relaxation (LR) approach, i.e., relaxing one complex constraint into the objective function and introducing Lagrangian multiplier $\lambda(p)$ alongside to construct the dualized Lagrangian function. In this paper, we reformulate the problem into QUBO model, which means all constrains are put into the objective function and the penalty coefficient are added.

This paper aims to optimize train space–time paths for the high-speed train timetable problem by providing daily schedule sets from historical schedules [30, 31]. Based on quantum computing technics, we contributed the following advancements to the research on the high-speed train timetable optimization:

(1) A simplified knapsack problem model (Sect. 2.2) is proposed to test the arithmetic power of quantum computing.
(2) To fit the requirements of quantum simulators of solving the timetable scheduling problem, a QUBO model (Sect. 2.3) is introduced and transformed.
(3) In order to compare the effectiveness of quantum simulators, two quantum algorithms including simulated annealing and CIM are utilized to solve the proposed QUBO model in different problem scales.

The structure of this paper is as follows: Sect. 2 presents our proposed integer programming flow model and binary integer programming model, which are based on the candidate-train-set and their corresponding QUBO models. In Sect. 3, various solution methods, including the CIM simulator, are discussed along with the presentation of four illustrative examples of different scales. Finally, the conclusions are provided in Sect. 4.

2 Mathematical model

Four models on timetable scheduling are constructed: ①Model ST (Space–time Network-Based Integer Programming Model), ②Model ST-QUBO (QUBO model of Model ST), ③Model KP (Knapsack Problem Reformulation [30] of Space–time Network Based Model), ④Model KP -QUBO (QUBO model of Model KP).

2.1 Space–time network-based integer programming model

2.1.1 Problem description

Based on graph theory, an optimization model using the space–time network is established for high-speed train timetabling [32]. To facilitate problem analysis, we abstract the train running time and state using the space–time network and discretize time into a single node. Passenger-oriented train timetables typically have a time accuracy of 1 min, and a day is divided into 1440 time nodes with 1 min as the unit, enabling the construction of space–time nodes and ensuring the accuracy requirements for describing the running state of high-speed trains. The space–time network represents the running states of the trains and the occupation of resources, with the horizontal axis representing the time dimension and the vertical axis representing the spatial dimension. The problem at hand is a typical large-scale combinatorial optimization problem.

2.1.2 Model assumptions

To enhance the alignment between the optimization model and real-world scenarios, and to increase its practical value, the following assumptions are made.

(1) Independence of upstream and downstream systems:

It is assumed that the upstream and downstream systems are completely independent. This means that in the model construction, only one direction of train operation is considered, while the optimization for the other direction can be conducted in a similar manner.

(2) Homogeneity of trains:

The assumption is made that trains are homogeneous. Currently, trains are primarily categorized into two grades: high-grade trains with fast running speeds and low-grade trains with relatively slower speeds. Trains within the same grade share the same

standards for factors such as interval running speed, dwelling time, and additional time required for train starting and stopping.

(3) Known order of train operation:

To ensure the balance of passenger transportation services provided by the operation department, this paper roughly arranges the departure sequence of trains based on the actual train operation chart's departure times. However, it should be noted that the actual departure time and sequence might differ from the operational chart.

(4) Availability of adequate resources for train movement:

Furthermore, it is assumed that sufficient resources, such as the number of train sets and crews, are available to facilitate the optimization of train-stopping schemes and timetable preparation.

2.1.3 Symbol definition

To ensure clarity and understanding, we will start by defining the sets, parameters, decision variables, and their corresponding symbols that are pertinent to the collaborative optimization model of the train stopping scheme and operation diagram. The definitions are presented as follows (Tables 1, 2).

2.1.4 Objective functions

Minimize the total train travel time costs In actual railway transportation operations, the primary goal of railway authorities is to minimize operating costs without compromising the quality of passenger service. To evaluate the quality of the train timetable, the total train travel time is considered as it reflects both the turnaround time and the utilization of transportation resources. Hence, minimizing the total train travel time serves as the initial optimization objective within the model. The cost associated with the total train travel time can be represented by the sum of the time costs for all trains within the selected arc.

$$Z_1 = \sum_{k \in K} \sum_{(i,j,t,t') \in A^k} c^k_{i,j,t,t'} \cdot x^k_{i,j,t,t'} \tag{1}$$

Minimize the total train departure delay at the departure station In order to ensure a balanced service to passengers, it is desired for trains to adhere to their desired departure time as much as possible. So the second optimization objective of this paper is to choose the minimum total departure delay of trains at the departure station.

$$Z_2 = \sum_{k \in K} \sum_{(i,j,t,t') \in A^{\text{owait}}} \left(t_{k'} - t^E_k \right) \cdot x^k_{i,j,t,t'} \tag{2}$$

Minimize the number of train stops Each additional stop for high-speed trains results in increased starting and stopping times, as well as an extended total stopping time. This not only diminishes the high-speed and convenient advantages of these trains, impacting passenger travel experience but also leads to wasted operational capacity

Table 1 Sets, indices, and parameters

Symbol	Description
T	Set of time
N	Set of stations
$N^k \in N$	Set of stations on the path of train k
K	Set of trains
L	Set of the interval, including physical interval and virtual interval set
A	Set of space–time extended arcs
i, i', j, j'	Stations, $i, i', j, j' \in N$
t, t', τ, τ'	Timepoints, $t, t', \tau, \tau' \in T$
k, k'	Trains, $k, k' \in K$
A^k	Set of space–time arcs on the path of train k
$A_i^{k,+}$	Set of spacetime arcs on the path of train k starting from A
$A_i^{k,-}$	Set of spacetime arcs on the path of train k that ends at A
A^{owait}	The origin-waiting arc of a train at its departure station
A^{drive}	The driving arc of the train
A^{dwell}	The dwell arc of the train
$\Omega_{(i,i',j,j')}$	Set of arcs that conflict with the driving arc i, i', j, j'
o_k, d_k	The starting and ending stations of train k, $o_k, d_k \in N_k$
t_k^E	The desired departure time of train k from its departure station o_k
t_k^d	The desired arrival time of train k from its departure station d_k
ΔT	Maximum departure delay allowed at the train's origin station, min
t_i^{min}	Minimum stopping time required at station i, min
t_i^{max}	Maximum stopping time required at station i, min
$T_{i,j}^k$	Travel time of train k between stations (i, j), min
$c_{(i,i',j,j')}^k$	The cost of arc (i, i', j, j') for train k
α_1	Weighting factor for travel time and number of stops
α_2	Weighting factor for delay time
t_k'	The actual departure time of train k at o_k
G_h, G_l	Upper and lower limits of the number of stops for high-grade trains
D_h, D_l	Upper and lower limits of the number of stops for low-grade trains
$\underline{f^i}$	The minimum service times of station i
$A_{i,j}$	Inter-station service accessibility index, i.e., the minimum number of trains stopping at stations i and j at the same time
R_i	Maximum number of stops allowed at station i in the studied direction
h_{dep}^{min}	Minimum departure interval of two adjacent trains, min
h_{arr}^{min}	Minimum arrival interval of two adjacent trains, min
g_k	When train k is a high-class train, the value is 1, otherwise it is 0

Table 2 Decision variables

Symbol	Description
$x^k_{i,i',j,j'}$	Binary variable(= 1 if train k occupies arc (i, i', j, j'); = 0 otherwise)
y_{ki}	Binary variable(= 1 if train k stops at station i; = 0 otherwise)
$s^k_{i,j}$	Binary variable(= 1 if train k stops at both station i and j; = 0 otherwise)

on the timetable. Therefore, the train stopping settings should minimize the number of unnecessary stops while still meeting the passenger flow exchange demand between stations, thereby ensuring a more rational and balanced distribution of stops. Therefore, the third optimization objective of the model is to minimize the number of stops for all trains.

$$Z_3 = \sum_{i \in N} \sum_{k \in K} y_{ki} \tag{3}$$

In order to construct a unified optimization objective, this paper introduces the corresponding weight coefficients α_1, α_1 and α_2 to the three optimization objectives. In particular, the values of α_1 and α_2 belong to [0,1] and satisfy the condition $\alpha_1 + \alpha_2 = 1$. The above problem with three optimization objectives is converted into a single-objective optimization problem, and the weighted sum of the above three costs is used as the optimization objective to construct. The objective function of Model ST is as follows.

$$\begin{aligned} minZ = & \sum_{k \in K} \sum_{(i,j,t,t') \in A^k} \alpha_1 \cdot c^k_{i,j,t,t'} \cdot x^k_{i,j,t,t'} + \alpha_1 \sum_{i \in N} \sum_{k \in K} y_{ki} \\ & + \alpha_2 \cdot \sum_{k \in K} \sum_{(i,j,t,t') \in A^{\text{ovait}}} \left(t_{k'} - t^E_k \right) \cdot x^k_{i,j,t,t'} \end{aligned} \tag{4}$$

2.1.5 Constraints

The constraints of Model ST are shown as follows.

Constraint on the number of stops of a single train In order to meet the demand of passenger boarding and alighting, and at the same time to ensure the requirements of running speed of different classes of trains, it is necessary to determine the number of train stops. The requirements on the number of stops for different classes of trains are different, so the upper and lower limits of the number of stops for different classes of trains should be scientifically determined and constructed as follows.

$$g_k \cdot G_l + (1 - g_k) \cdot D_l \le \sum_{i \in N^k} y_{ki} \le g_k \cdot G_h + (1 - g_k) \cdot D_h, \quad \forall k \in K \tag{5}$$

Station service frequency constraint In order to meet the passenger demand between stations, each station has the requirement of minimum number of stops. The number of trains stopping at a station needs to meet the service frequency constraint of that station, which is constructed as follows:

$$\sum_{k \in K} y_{ki} \geq \underline{f^i}, \quad \forall i \in N \tag{6}$$

Train stopping time constraints In the space–time network, the selection of station stop arcs in the space–time path of the train and their space–time length describe the stopping operation of the train and the corresponding stopping time. If $y_{ki} = 1$, train k stops at station i (not the train departure and arrival station), the stopping time of train k at station i should be no less than the minimum stopping time t_i^{min} to ensure that the train can successfully complete the necessary technical operations, such as passenger boarding and alighting, crew shift change, etc., during the stopping. Based on the correlation analysis of the train stopping scheme and the operation diagram decision variables before, the following is constructed.

$$\sum_{t'} \sum_{t} x^k_{i,i,t,t'} \geq y_{ki} \cdot t_i^{min}, \quad \forall i \in N_k \quad \text{and} \quad i \notin (\{o_k\} \cup \{d_k\}) \tag{7}$$

$$\sum_{t'} \sum_{t} x^k_{i,i,t,t'} \leq y_{ki} \cdot t_i^{max}, \quad \forall i \in N_k \quad \text{and} \quad i \notin (\{o_k\} \cup \{d_k\}) \tag{8}$$

Considering only a minimum stopping time constraint in the model cannot guarantee that when $y_{ki} = 0$, $x^k_{i,i,t,t'}$ is also 0, i.e., no station stopping arc is selected, so in this paper, by adding a constraint of maximum train stopping time (Cons.8), the two constraints together form a standard constraint of train stopping time, which makes the optimization result of the train operation diagram more scientific, while the maximum stopping time constraint can also reduce the feasible domain of the model and improve the solving efficiency of the model.

Flow balancing constraints The flow balance constraint is the basic constraint in solving the path selection optimization problem using graph theory. In practice, the flow of trains at each origin, intermediate and destination space–time nodes in the operation path should always be balanced, i.e., trains in the space–time network can only choose one space–time arc segment from the origin to the destination, and the entire space–time path of the train should be guaranteed to be continuous and uninterrupted at the intermediate nodes. The space–time path of each train is unique and the construction constraint is as follows.

$$\sum_{(i,j,t,t') \in A_i^{k,+}} x^k_{i,j,t,t'} - \sum_{(j,i,t',t) \in A_i^{k,-}} x^k_{j,i,t,t} = \begin{cases} 1, & i = o_k, \quad t = t_k^E \\ -1, & i = d_k, \quad t = t_k^d \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

Since the arrival time of the train is difficult to determine before optimization, but all trains must complete the transportation service within the operation time, this paper

treats the virtual terminal as the end point in the whole space–time network of train operation. In addition, in order to ensure the provision of transportation services as balanced as possible, trains have their determined desired departure time, which is used as the time attribute of the train departure waiting arc in the space–time network in this paper.

Constraint on the departure time expectation of the starting station In order to provide balanced transportation services to passengers as much as possible, each train has a corresponding desired departure time t_k^E at the departure station. In practice, if trains operate strictly according to this time, and the time is not set reasonably enough, coupled with factors such as mutual interference between trains of different speeds, it may lead to conflicts between trains or cause waste of operating chart capacity. Therefore, this paper allows trains to fluctuate somewhat at the departure station compared to the desired departure moment, and at the same time, in order to ensure the necessary preparation activities before train departure, trains are only allowed to delay departure, with the delay time set to T.

In this paper, the desired departure moment of the train at the origin station is regarded as the arrival moment of the train at that station (from the virtual origin station), and thus the constraint can be expressed as the stopping time of the train at the origin station is not greater than the maximum delay of the train departure at the origin station, which is constructed as follows.

$$\sum_{t'}\sum_{t} x_{i,i,t,t'}^{k} \le \Delta T, \quad \forall\left(i,i,t,t'\right) \in A^{\text{owait}}, \quad \forall k \in K \tag{10}$$

Station stopping capacity constraint Since a station has only a limited number of mainline and arrival and departure line resources, the station's stopping capacity constraint limits the maximum number of trains stopping at the station at the same time, and its value should be less than or equal to the maximum train stopping capacity of the station.

In the space–time network, the number of trains occupying the resources of a station at the same time can be expressed by the number of stopping arcs at the station at that time. In this paper, the stopping time of trains is discretized into intervals of 1 min, and then the station stopping capacity constraint is constructed as follows.

$$\sum_{k\in K}\sum_{t'} x_{i,i,t,t'}^{k} \le R_i, \quad \forall\left(i,i,t,t'\right) \in A^{\text{dwell}}, \quad \forall i \in N/1, |N| \tag{11}$$

Train safety interval constraint (departure and arrival) In order to ensure the safety of trains, the continuous arrival and continuous departure of trains need to meet certain interval time criteria, so the constraint can be specifically subdivided into constraints on the departure and arrival intervals of trains at stations. In this paper, we define the incompatible arc set $\Omega_{(i,i',j,j')}$ to express the constraints on the departure and arrival time interval between two trains, for the interval running arc $(i, j, t, t') \in A^{\text{drive}}$ of

the space–time network, its incompatible arc set $\Omega_{(i,i',j,j')}$ is defined as

$$\Omega_{(i,j,t,t')} = \left\{ \left(i, j, \tau, \tau'\right) : |t - \tau| < h_{dep}^{min} \bigcup \left|t' - \tau'\right| < h_{arr}^{min} \right\}. \tag{12}$$

The parameters h_{dep}^{min} and h_{arr}^{min} denote the minimum departure and arrival intervals between two trains, respectively. Equation (11) indicates that if arc $\left(i, j, t, t'\right) \in A^{\text{drive}}$ is used by a train, then for any arc $\left(i, j, \tau, \tau'\right) \in A^{\text{drive}}$, as long as the arrival between the $arc(i, j, t, t')$ or departure interval is less than the minimum, all belong to the incompatible arc set $\Omega_{(i,i',j,j')}$, which cannot be used by other trains, and thus construct the train safety interval constraint as follows.

$$\sum_{k \in K} \sum_{(i,i',j,j') \in \Omega_{(i,i',t,t')}} x_{i',j',\tau,\tau'}^{k} \leq 1, \quad \forall \left(i, j, t, t'\right) \in A^{\text{drive}} \tag{13}$$

Interval crossing constraint In order to improve the efficiency of the utilization of the resources of the operation chart, generally the high-grade trains will cross the low-grade trains to improve the passing capacity of the interval, but since the interval does not have the crossing condition, the crossing cannot occur in the interval. When the trains in the same direction interval run at different speeds, each train may clash although they satisfy the train safety interval constraint, and then the constraint needs to be considered.

$$x_{i,j,\tau,\tau'+T_{i,j}^{k}}^{k} + \sum_{t \in \left[\tau, \tau+\tau_{i,j}^{k}\right]} \sum_{t' \left[\tau, \tau+\tau_{i,j}^{k}\right]} x_{i,j,t,t'}^{k'} \leq 1, \quad \forall i, j, k, k', \tau \tag{14}$$

Because of the existence of the train safety interval constraint, interval crossing of trains in the same direction occurs when and only when the running time of two trains satisfies $T_{i,j}^{k} - T_{i,j}^{k'} > h_{dep}^{min} + h_{arr}^{min}$.

Maintenance gap constraint China's high-speed railroads specify the maintenance for 00:00–6:00, in this time period, train operation is prohibited, in the space–time network can be expressed as the skylight range of space–time arc occupancy variables take the value of 0, the construction of the following.

$$x_{i,j,t,t'}^{k} = 0, \forall k \in K, \forall i, j \in N, \forall 0 \leq t, t' \leq 360 \tag{15}$$

Constraints on the value of decision variables

$$x_{i,j,t,t'}^{k} \in 0, 1, \quad \forall k \in K, \quad \left(i, j, t, t'\right) \in A \tag{16}$$

$$y_{ki} \in 0, 1, \quad \forall k \in K, \quad i \in N \tag{17}$$

Model ST involves a wide range of influencing factors, and in the process of solving them, all of them will explode in combination as the scale of the problem expands,

[33] i.e., it is impossible to solve to an exact solution in polynomial time. This model is hence still complicated for quantum computing in the solution process, and the solution results are almost always infeasible. To solve this problem, we propose a simplified model to test the arithmetic power of quantum computing.

2.2 Knapsack problem reformulation of space–time network-based model

We use the space–time network model and historical timetable data to generate train running line, then we propose a 0–1 integer programming model based on the knapsack problem to simplify the model, which fixes the decision variables such as train arrival and departure times and train stops in the original model and uses the historical train trajectory selection as the core decision variable.

2.2.1 Symbol description

The definition of the sets, parameters, decision variables, and their notation involved in the binary integer programming model based on knapsack problem is shown in Tables 3 and 4.

Table 3 Sets, index, and parameters

Symbol	Description
T	Set of time
V	Set of stations
K	Set of train running lines, each of which includes both temporal and spatial dimensions
$\mathbb{N}$	Set of incompatible arcs for trains
N_k	Set of train running lines that conflict with train running line k, $N_k \in \mathbb{N}$
$t_{k,v}$	Index of time, $t \in T$, $k,\ k' \in K$, $v \in V$
v	Index of station, $v \in V$
$k,\ k'$	Index of train running lines, $k,\ k' \in K$
V_k	Set of stations of train running line k, $v \in V_k$
h^{min}	The minimum headway time

Table 4 Decision variables

Symbol	Description
x_k	Binary variable, when the kth running line is selected, $x_k \in 1$, else,$x_k \in 0$

2.2.2 Objective function

With the objective of optimizing the maximum number of train running lines to be drawn, the objective function is as Eq. (18).

$$\max Z = \sum_{k \in K} x_k \tag{18}$$

2.2.3 Constraints

This paper mainly considers the train headway constraint. In order to describe this constraint, firstly the incompatible arc set is defined as shown in ①, and then the constraint expression is given as shown in ②.

①Set of incompatible arcs.

In order to ensure the safety of trains, the continuous arrivals and departures of trains need to meet minimum headway constraint. In this paper, for each train running line $k \in K$, a set of incompatible arcs N_k is defined in Eq. (19).

$$N_k = \left\{k : \ |t_{k'v} - t_{kv}| < h^{min}\right\} \forall v \in V, \quad \forall k' \in K, \quad k \neq k' \tag{19}$$

②Expression for the train safety interval constraint.

In the final selection of running arcs to form the timetable, it is required that no more than one running arc is selected from each incompatible arc set, and thus the train safety headway constraint is constructed as Eq. (20).

$$x_k \cdot \sum_{k' \in N_k} x_{k'} = 0, \quad \forall N_k \in \mathbb{N} \tag{20}$$

2.3 QUBO representation of the models

2.3.1 QUBO representation of model ST

While the QUBO formulation does not inherently support constraints, it is possible to incorporate constraints into the optimization process. These constraints can be effectively handled by translating them into penalty terms that impact the objective function. By reformulating constraints as quadratic equations and adding them to the objective function, we can ensure that they influence the optimization process. For detailed derivation rules regarding the transformation of non-QUBO models into QUBO models, please refer to Appendix.

For the inequality constraints existed in the Model ST, the slack positive integer variables s_p need to be introduced to transform inequalities into equalities. And $s_p = \sum_{q=0}^{r_s} 2^q v_q$ where r_s is an integer and v_q is binary variable [34]. Each penalty should be multiplied by a positive constant λ to have comparable magnitude with the objective

function. A simpler example of such transformation is shown in Appendix. And the QUBO formulation result is shown as follows.

2.3.2 QUBO representation of model KP

Binary integer programming model based on knapsack problem mentioned in Sect. 2.2 is formulated as a QUBO problem. And the corresponding QUBO model is constructed as follows.

To transform the integer programming model described in Sect. 2.2 into a QUBO model, a three-step process is employed. First, we introduce a parameter, denoted as λ, which serves as the penalty coefficient for handling constraints. Second, we modify the constraint equation [Eq. (20)] by multiplying it with the penalty coefficient, thereby transforming the inequalities into equations. Subsequently, these modified equations are integrated into the objective function. The resulting objective function is expressed as Eq. (21).

$$\min \left[-\sum_{k \in K} x_k + \lambda \cdot x_k \cdot \sum_{k' \in N_k} x_{k'} \right] \tag{21}$$

2.4 Comparisons among the proposed models

The relationship of the four proposed models is shown in Fig. 2.

In this paper, Fig. 2 illustrates the integration of models introduced in Sects. 2.2 and 2.3. We present Model ST, the space–time network model for integer programming,

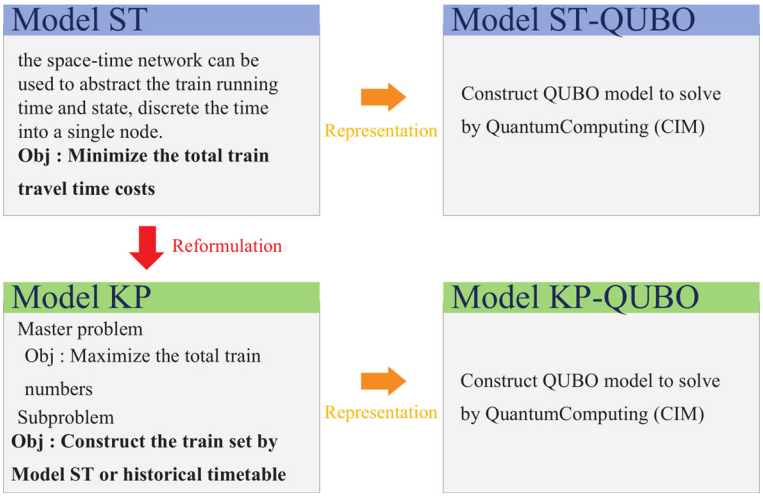


Fig. 2 Relationship of the four proposed models

alongside its corresponding QUBO model, referred to as Model ST-QUBO. Similarly, model KP, the alternative set solution model for integer programming, is displayed alongside its corresponding QUBO model, Model KP-QUBO. This visual representation facilitates a clear understanding of the relationships between these models and their formulations, offering a valuable tool for those interested in exploring the optimization of space–time networks using integer programming and QUBO models.

A comprehensive comparison among the four proposed models is shown in Table 5.

3 Algorithms and analysis

We have used Model ST and Model ST-QUBO to solve the train scheduling problem. However, due to the complexity of this type of model and limited capacity of quantum simulator, the solving results were not satisfactory. Therefore, we decide to focus on Model KP and Model KP-QUBO for analysis. Two different algorithms are applied to solve these two models. For the general Model KP, we use the classical optimization solver Gurobi, whose result serves as a control group. For Model KP-QUBO, we apply SA (simulated annealing) and Kaiwu-SDK. The process of using a coherent Ising machine (CIM) to solve the QUBO model involves inputting the q_{ij} from the QUBO into the CIM, and then the CIM returns the value of decision variables. And Kaiwu-SDK is a software development kit developed by QBoson company for solving QUBO problems using CIM.

3.1 Numerical calculations

3.1.1 Defining penalty value λ

To assess the feasibility of the models, we initially employ the Gurobi solver and then investigate how the penalty term λ affects the QUBO problem, taking the example of 100 trains as an instance. Within the QUBO model of the problem, there exists a single penalty term λ, and as λ assumes different values, the solution outcome of the QUBO problem undergoes changes. Theoretically, a larger value of λ should result in a better solution. However, it is crucial to note that a higher λ value also leads to longer solution times. Therefore, considering the attainment of a reasonable solution time while ensuring that the penalty term value and the original objective function value are within the same order of magnitude, we choose $\lambda = 100$ as the penalty term for subsequent quantum solver experiments.

3.1.2 Numerical studies on SA

We employ simulated annealing (SA) to tackle this QUBO problem, utilizing 100 quantum bits with 168 different coefficients. The SA solution function includes several input parameters that require adjustment based on the quality of the solution obtained during the solving process, aiming to uncover the optimal solution.

Table 5 Comparisons among the four proposed models

	Model M1-ST	Model M2-ST-QUBO	Model M3-KP	Model M4-KP-QUBO
Model type	Arc routing model (MILP)	Arc routing model (QUBO)	Path-based model (MILP)	Path-based model (QUBO)
Main decision variables	$x_{i,j,t,t'} \in \{0,1\}, \forall (i,j,t,t') \in A$	$x_{i,j,t,t'} \in \{0,1\}, \forall (i,j,t,t') \in A$	$x_i \in \{0,1\}, \forall i \in Z$	$x_i \in \{0,1\}, \forall i \in Z$
Side constraints	Flow balance constraint and others	None	Headway constraint	None
Features or challenges	A large number of additional decision variables are introduced due to intersection expansion	Hard to find a feasible solution	Only one set of side constraint	Hard to find the optimal solution

T_{init}: initial temperature of simulated annealing.
α: temperature drop rate of simulated annealing.
T_{min}: minimum threshold temperature for simulated annealing.
$Iters_{\text{per}}$: number of iterative runs per cycle.

T_{init} and $Iters_{\text{per}}$ exert a substantial influence on the solution time and results, while the remaining parameter α is fixed at 0.99 and T_{min} is set to 10^{-3}. Our tests indicate that increasing T_{min} yields a slight enhancement in solution quality without causing a notable change in solution time. Conversely, elevating the value of $Iters_{\text{per}}$ significantly impacts the solution and can lead to improved solution quality. We observe that when T_{min} falls within the range of 500–1000 and $Iters_{\text{per}}$ also lies within the same range, a balance can be struck between solution quality and computational time.

3.2 Numerical studies on CIM

Kaiwu-SDK function has the following parameters, which also need to be adjusted according to the quality of the solution obtained during the solution process in order to find the optimal solution.

$Pump$: pump rate.
$Noise$: noise power.
$Laps$: number of laps per run.
D_t: time step per lap.
$Normalization$: minimum negative eigenvalue normalization factor of the matrix.
$iterations$: number of independent runs.

After testing, a set of parameter adjustment scheme has been summarized as follows:

The pump parameter is positively correlated with the solution quality in the interval of [0.5, 2] and does not affect the running time too much.

The noise parameter is adjusted in the interval of [0.001,0.5], increasing appropriately can increase the randomness of the solution and make it easier to obtain the optimal solution.

The iterations parameter is positively correlated with the quality of the solution within a certain range, but it will significantly affect the running time and should be choosed according to the application scenario.

3.3 Numerical cases

To test the effectiveness of different solution algorithms, four different sizes of alternative sets are given for testing. The tests are conducted on a computer with a Core i7-12700H with 32 GB of RAM, and the runtime on that computer is used as a benchmark for algorithm efficiency comparisons. When CIM finds the better solution, we compare the solution found by SA.

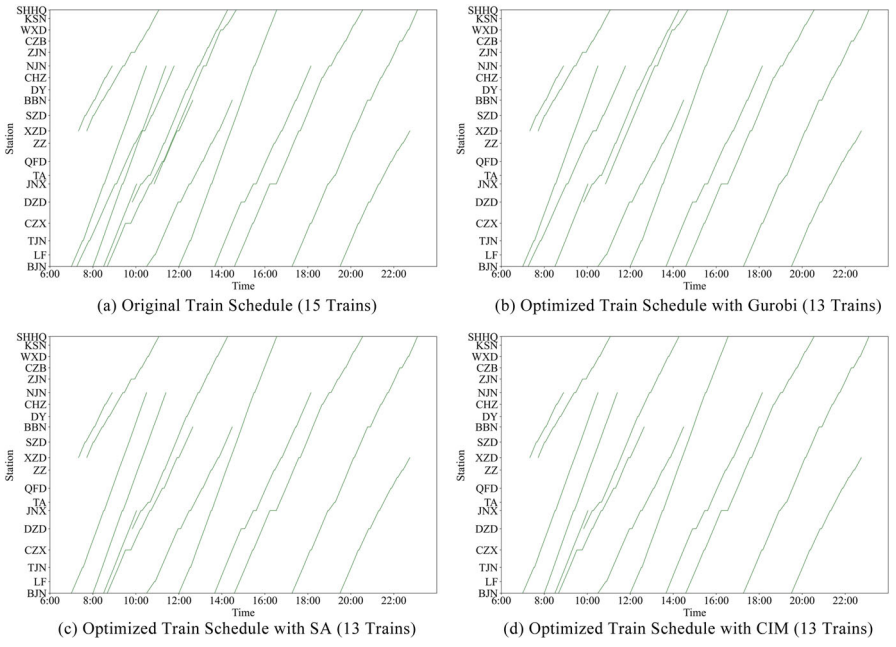
3.3.1 15-Train case

The case of 15 trains is used to test the correctness of the algorithm and the effect of running it under small-scale conditions. We formulate an alternative set of 15 trains,

Table 6 Results of 15-train case

Algorithms	SA	CIM-Kaiwu
Result	13 trains	13 trains
Parameters	$T_{\text{init}} = 200$	Pump Rate = 1.2
	$\alpha = 0.90$	Noise Power = 0.224
	$Iters_{\text{per}} = 200$	
Time consumed	4.1 s	4.1 s

2 of which have significantly conflicting relationships with other trains. The optimal solution in the case is 13 trains. In the test, all algorithms can find the correct result, where CIM uses parameter random strategy and SA is fixed parameter, the parameters are shown in the table. In the resulting figures, the horizontal axis of each subplot represents the timeline of the train schedule, while the vertical axis represents all the stations along the route. The movement of all trains is depicted by a green line, representing their respective trajectories. Additionally, we have included the optimal results achieved by the corresponding solver in the title of each subplot. The subsequent explanations of the charts remain the same as mentioned above (Table 6; Fig. 3).



(a) Original Train Schedule (15 Trains)

(b) Optimized Train Schedule with Gurobi (13 Trains)

(c) Optimized Train Schedule with SA (13 Trains)

(d) Optimized Train Schedule with CIM (13 Trains)

Fig. 3 Train schedule of 15-train case

3.3.2 30-Train case

The case of 30 trains is used to test the correctness of the algorithm and the effect of running it under medium-scale conditions. The optimal solution in the case is 25 trains.

We noticed that the solution of the SA and CIM quantum algorithm is not optimal (Table 7; Fig. 4).

Table 7 Results of 30-train case

Algorithms	SA	CIM-Kaiwu
Result	21 trains	23 trains
Parameters	$T_{\text{init}} = 200$	Pump Rate = 1.6
	$\alpha = 0.90$	Noise Power = 0.30
	$Iters_{\text{per}} = 200$	
Time consumed	6.8 s	6.8 s

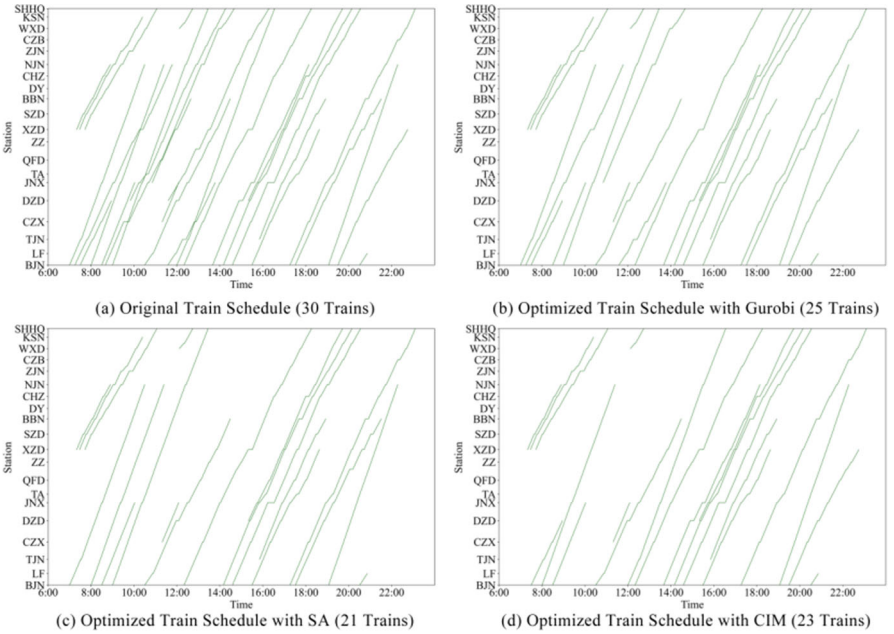
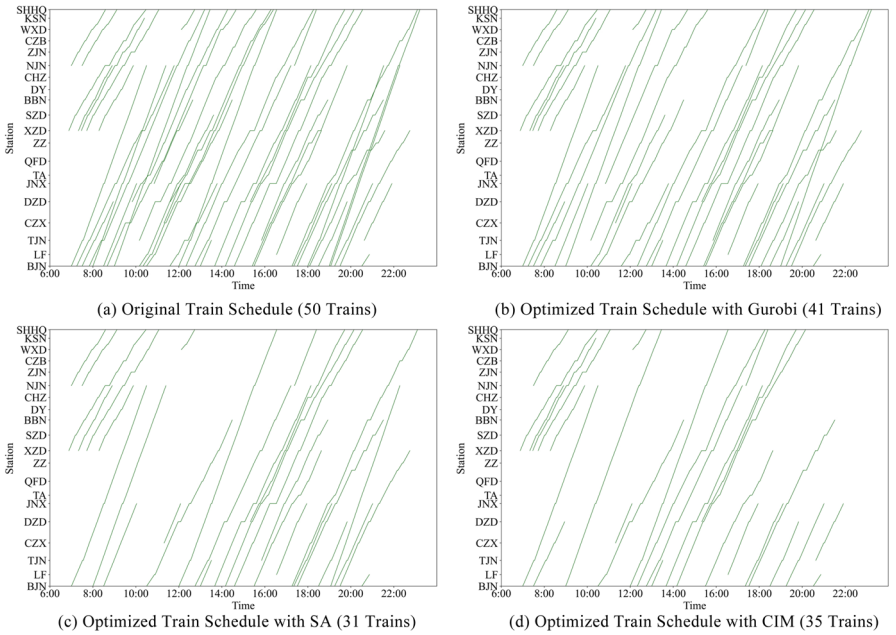


Fig. 4 Train schedule of 30-train case

Table 8 Results of 50-train case

Algorithms	SA	CIM-Kaiwu
Result	31 trains	35 trains
Parameters	$T_{\text{init}} = 200$	Pump Rate = 1.5
	$\alpha = 0.90$	Noise Power = 0.11
	$Iters_{\text{per}} = 200$	
Time consumed	10.1 s	10.1 s



(a) Original Train Schedule (50 Trains)

(b) Optimized Train Schedule with Gurobi (41 Trains)

(c) Optimized Train Schedule with SA (31 Trains)

(d) Optimized Train Schedule with CIM (35 Trains)

Fig. 5 Train schedule of 50-train case

3.3.3 50-Train case

The case of 50 trains is used to test the correctness of the algorithm and the effect of running it under medium-scale conditions. The optimal solution in the case is 41 trains (Table 8; Fig. 5).

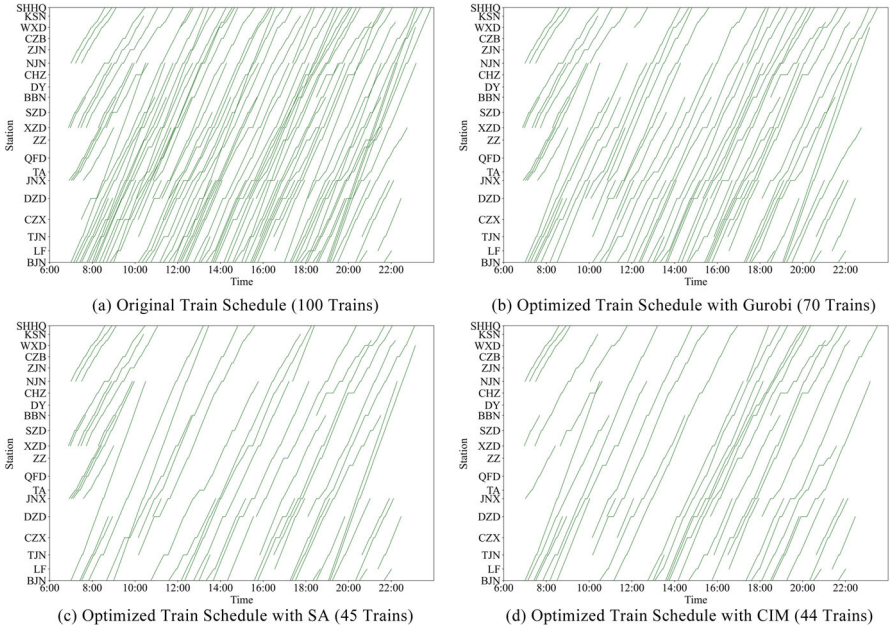
3.3.4 100-Train case

The case of 100 trains is used to test the correctness of the algorithm and the effect of running it under large-scale conditions. The optimal solution in the case is 70 trains (Table 9; Fig. 6).

It is noteworthy that the quality of solutions obtained through CIM and SA quantum algorithms tends to diminish as the problem size increases. In the case of large-scale

Table 9 Results of 100-train case

Algorithms	SA	CIM-Kaiwu
Result	45 trains	44 trains
Parameters	$T_{\text{init}} = 200$	Pump Rate = 1.5
	$\alpha = 0.90$	Noise Power = 0.11
	$Iters_{\text{per}} = 200$	
Time Consumed	23.3 s	23.3 s



(a) Original Train Schedule (100 Trains)

(b) Optimized Train Schedule with Gurobi (70 Trains)

(c) Optimized Train Schedule with SA (45 Trains)

(d) Optimized Train Schedule with CIM (44 Trains)

Fig. 6 Train schedule of 100-train case

problems, CIM seems to produce infeasible solutions. In general, both methods demonstrate the capability to achieve optimal solutions with similar computation times in small-scale test cases. However, for medium-scale cases such as the 30 Train Case and 50 Train Case, the CIM method exhibits advantages in terms of solution quality over the SA method. The overall solution performance is depicted in the graph below (Fig. 7).

4 Conclusions

Quantum simulators have the potential to provide significant speed-ups for combinatorial optimization problems. To utilize these simulators, the first step is to convert the problem from its conventional notation into the QUBO form, which is the format

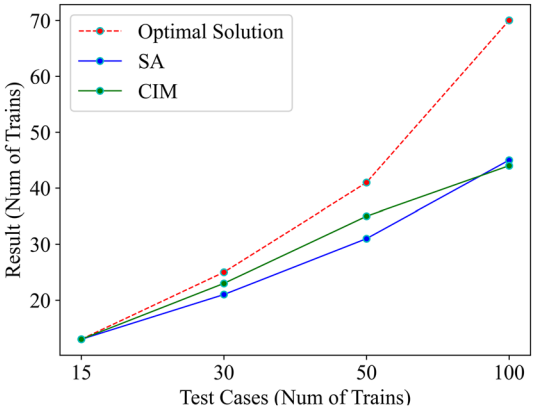


Fig. 7 Comparison of running results

required by quantum simulators. This paper presents the QUBO representation for the railway timetabling problem, which involves determining the optimal train schedule based on a given set of conflicting trains.

However, there are three primary reasons why the current solving speed and outcomes of quantum simulators are not yet ideal. Firstly, the number of qubits remains a bottleneck for current quantum devices [35], and thus it is preferable to use the smallest possible number of qubits for problem modeling. In addition, the values of the model parameters (such as pump rate, noise power, and times of simulated annealing) also affect the results. Furthermore, finding appropriate penalty values for QUBO problems is a challenging task. While we have used experiments and manual calibration to determine the penalty values for our railway timetabling problem, recent research has proposed several algorithms for penalty determination. For instance, the cross-entropy optimization [36] and another algorithm [37] tested successfully on the particular Fujitsu digital annealer. These methods may offer a more efficient and automated approach for determining penalty values in QUBO problems.

Looking ahead, as quantum devices continue to improve, we anticipate more efficient solutions for larger problem instances. This progress holds the potential to significantly enhance the practicality and applicability of quantum devices for solving complex optimization challenges. Moreover, the problem transformation into the QUBO framework is not exclusive to railway timetabling; its adaptability extends to a myriad of combinatorial optimization problems in transportation field, such as routing, scheduling, assignment and rostering, namely vehicle routing problem, automated guided vehicle (AGV) scheduling, electric bus scheduling, railway crew assignment and rostering, and beyond. In fact, this method could also be used in other fields related to combinatorial optimization, such as telecommunications (e.g., wireless sensor network deployment), finance and economics (e.g., portfolio optimization), healthcare (e.g., nurse scheduling), and so on. The foundational principles and techniques presented in this paper lay the groundwork for addressing a diverse range of real-world optimization problems using quantum computing methodologies.

Acknowledgements This research was funded by China State Railway Group Co., Ltd. (Grant No. N2022X030 and K2022X014), Natural Science Horizontal Project (Grant No. T22L00120) and Young Scientists Fund of the National Natural Science Foundation of China (Grant No. 72001021)

Declarations

Conflict of interest The authors declare no conflict of interest.

Availability of data and materials The data that support the findings of this study are available from the corresponding author upon reasonable request.

Appendix

Table of different quantum computers

See Appendix Table 10.

Derivation of QUBO models from a general non-QUBO model

The methods of reformulating general models into QUBO models can be found in [23]. Here, we only summarize the key points in two cases/approaches.

Creating QUBO models using known penalties

Many of the constrained models can be reformulated as a QUBO model by introducing quadratic penalties into the objective function. For a minimization problem, these penalties are formulated so that they equal zero for feasible solutions and equal some positive penalty amount for infeasible solutions. And for certain types of constraints, during the process of transforming a given constrained problem into a QUBO model, the corresponding quadratic penalties could be known in advance. Examples of such penalties for some commonly encountered constraints are shown in the Table 11. Note that in the table, all variables are binary and the parameter P is a positive, scalar penalty value. This value needs be set large enough to assure that the penalty term could play a role in penalizing the infeasible solutions and thus make itself indeed equivalent to the classical constraint. But in practice an acceptable value for P is usually easy to discover.

Table 10 Table of different quantum computers

	Superconductors	Ion trap	Semiconductor quantum dots	Topology	Quantum annealer
Coherent time	About 50 μ s	More than 1000 s	About 100 μ s	Theoretically could be as long as possible	0.05 μ s
Research groups	Google, IBM, Intel, Quantum Circuits, Rigetti, Origin Quantum, Zhejiang University, Nanjing University, Beijing Academy of Quantum Information Sciences	IonQ, Honeywell, NIST, Cambridge Quantum, Tsinghua University, University of Science and Technology of China	Intel, Princeton University, Delft University of Technology, Origin Quantum, University of Science and Technology of China	Microsoft, Delft University of Technology, Tsinghua University, Peking University, Institute of Physics Chinese Academy of Sciences	D-Wave
Advantages	① High degree of control in manipulation and preparation	① Long coherence time	① Good scalability	①Small number of physical qubits needed	Could be put into commercial use more easily
	② Efficient readout	② High implementation of both single- and two-qubit gates	② Potential to scale up to extremely large number of qubits operations	② Extremely high fidelities on the physical qubits	

Table 10 (continued)

	Superconductors	Ion trap	Semiconductor quantum dots	Topology	Quantum annealer
	③ Possibility to tailor circuit properties and implement tunable qubit frequencies and coupling strengths	③ Efficient qubits reproducibility			
		④ Small number of calibration steps required at the beginning of the computation			
Disadvantages/ Challenges	Extremely strict physical environment	Scaling to larger numbers of trapped ions has been slow and it is hard to implement large-scale quantum computation	① Short coherence time	Significant materials, fabrication, and measurement challenges	Only able to solve specific optimization questions
			② Reliable and high-fidelity two-qubit gates need developed		

① In this table, we only list some typical-related research groups and main advantages and disadvantages. We extract the contents from Kjaergaard et al. [38], Bruzewicz et al. [39], NASEM [40]

② As for an introduction to the pros and cons of these different quantum computers, we must point out that the field of quantum computing is constantly evolving, so many advantages and disadvantages may change as research progresses

Table 11 A few known constraint/penalty pairs [41]

Table 11 (continued)

Index	Classical constraint	Equivalent penalty
1	$x + y \leq 1$	$P(xy)$
2	$x + y \geq 1$	$P(1 - x - y + xy)$
3	$x + y = 1$	$P(1 - x - y + 2xy)$
4	$x \leq y$	$P(x - xy)$
5	$x_1 + x_2 + x_3 \leq 1$	$P(x_1x_2 + x_1x_3 + x_2x_3)$
6	$x = y$	$P(x + y - 2xy)$

Creating QUBO models using a general purpose approach

It turns out that for the general cases when QUBO formulation does not arise naturally or useable penalties are not known in advance, some useable penalties could be specified by adopting the procedures given below and an example will be provided to illustrate it.

But let's consider the general 0/1 problem first. The problem appears in this form:

$$\min y = x^T Cx$$
$$Ax = b,\ x \text{ binary}$$

Specifically, for a positive scalar P, we add a quadratic penalty $P(Ax - b)^t(Ax - b)$ to the objective function to get

$$y = x^T Cx + P(Ax - b)^t(Ax - b) = x^T Cx + x^T Dx + c = x^T Qx + c$$

where the matrix D and the additive constant c result directly from the matrix multiplication indicated. Dropping the additive constant, the equivalent unconstrained version of the constrained problem becomes

$$QUBO : \min\ x^t Qx,\ x \text{ binary}.$$

Now the example where the constraints are not equalities but inequality is provided and the procedure is shown to deal with such situation.

$$\min y = f(x)$$

Subject to $4x_1 + 5x_2 - x_3 \leq 6$ and x is a binary variable.

***Procedure*:**

A. Introduce a slack variable s to convert the inequality into the equality

$$4x_1 + 5x_2 - x_3 + s = 6$$

b. Find out the upper bound and lower bound of the slack variable

$$s_{max} = 6 - min(4x_1 + 5x_2 - x_3) = 6 - (0 + 0 - 1) = 7$$
$$s_{min} = 6 - max(4x_1 + 5x_2 - x_3) = 6 - (0 + 5 + 0) = 1$$

c. Use a series of binary variables v_q to express the slack variable s

$s = 2^0 * v_1 + 2^1 * v_2 + 2^2 * v_3 = v_1 + 2v_2 + 4v_3$ where v_1, v_2, and v_3 are additional binary variables.

d. Get the new equality constrain and form the quadratic penalties

$P(4x_1 + 5x_2 - x_3 + v_1 + 2v_2 + 4v_3 - 6)^2$ where P is a positive, large enough, scalar penalty value.

e. Put the quadratic penalties into the original objective function

$$\min y = f(x) + P(4x_1 + 5x_2 - x_3 + v_1 + 2v_2 + 4v_3 - 6)^2.$$

References

1. Benioff, P.: The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by turing machines. J. Stat. Phys. **22**(5), 563–591 (1980)
2. Feynman, R.P.: Quantum mechanical computers. Found. Phys. **16**(6), 507–531 (1986)
3. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science, pp. 124–134. IEEE (1994)
4. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)
5. Peng, W., Fang, W.: Overview of intelligent optimization algorithms from the perspective of quantum **51**(01), 2–15 (2022)
6. Cooper, C.: Exploring potential applications of quantum computing in trans- portation modelling. IEEE Trans. Intell. Transp. Syst. (2021)
7. Di Martino, F., Sessa, S.: A novel quantum inspired genetic algorithm to initialize cluster centers in fuzzy c-means. Expert Syst. Appl. **191**, 116340 (2022)
8. Das, M., Roy, A., Maity, S., Kar, S.: A quantum-inspired ant colony optimization for solving a sustainable four-dimensional traveling salesman problem under type-2 fuzzy variable. Adv. Eng. Inform. **55**, 101816 (2023)
9. Fang, W., Sun, J., Ding, Y., Wu, X., Xu, W.: A review of quantum-behaved particle swarm optimization. IETE Tech. Rev. **27**(4), 336–348 (2010)
10. Yumin, D., Li, Z., Zong, F.: Quantum behaved particle swarm optimization algorithm based on artificial fish swarm. Math. Probl. Eng. **2014**, 592682 (2014)
11. Wen, J., Wang, Z., Huang, Z., Cai, D., Jia, B., Cao, C., Ma, Y., Wei, H., Wen, K., Qian, L.: Optical experimental solution for the multiway number partitioning problem and its application to computing power scheduling. Sci. China Phys. Mech. Astron. **66**(9), (2023)
12. Martŏnák, R., Santoro, G.E., Tosatti, E.: Quantum annealing of the traveling-salesman problem. Phys. Rev. E **70**(5), 057701 (2004)

13. Harris, R., Sato, Y., Berkley, A., Reis, M., Altomare, F., Amin, M., Boothby, K., Bunyk, P., Deng, C., Enderud, C.: Phase transitions in a programmable quantum spin glass simulator. Science **361**(6398), 162–165 (2018)
14. Brady, L.T., Baldwin, C.L., Bapat, A., Kharkov, Y., Gorshkov, A.V.: Opti- mal protocols in quantum annealing and quantum approximate optimization algorithm problems. Phys. Rev. Lett. **126**(7), 070505 (2021)
15. Rajak, A., Suzuki, S., Dutta, A., Chakrabarti, B.K.: Quantum annealing: an overview. Philos. Trans. R. Soc. A **381**(2241), 20210417 (2023)
16. Patton, R., Schuman, C., Potok, T.: Efficiently embedding qubo problems on adiabatic quantum computers. Quantum Inf. Process. **18**(4), 1–31 (2019)
17. Yarkoni, S., Raponi, E., Bäck, T., Schmitt, S.: Quantum annealing for industry applications: introduction and review. Rep. Progr. Phys. **85**(10), (2022)
18. Carugno, C., Ferrari Dacrema, M., Cremonesi, P.: Evaluating the job shop scheduling problem on a d-wave quantum annealer. Sci. Rep. **12**(1), 6539 (2022)
19. Bożejko, W., Pempera, J., Uchroński, M., Wodecki, M.: Distributed quantum annealing on d-wave for the single machine total weighted tardiness scheduling problem. In: International Conference on Computational Science, pp. 171–178. Springer (2022)
20. Mohseni, N., McMahon, P.L., Byrnes, T.: Ising machines as hardware solvers of combinatorial optimization problems. Nat. Rev. Phys. **4**(6), 363–379 (2022)
21. Otsuka, T., Li, A., Takesue, H., Inaba, K., Aihara, K., Hasegawa, M.: High-speed resource allocation algorithm using a coherent ising machine for noma systems. IEEE Trans. Veh. Technol. https://doi.org/10.1109/TVT.2023.3300920 (2023)
22. Zhang, T., Tao, Q., Liu, B., Han, J.: A review of simulation algorithms of classical ising machines for combinatorial optimization. In: 2022 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1877–1881. IEEE (2022)
23. McMahon, P.L., Marandi, A., Haribara, Y., Hamerly, R., Langrock, C., Tamate, S., Inagaki, T., Takesue, H., Utsunomiya, S., Aihara, K., et al.: A fully pro- grammable 100-spin coherent ising machine with all-to-all connections. Science **354**(6312), 614–617 (2016)
24. Inagaki, T., Haribara, Y., Igarashi, K., Sonobe, T., Tamate, S., Honjo, T., Marandi, A., McMahon, P.L., Umeki, T., Enbutsu, K., et al.: A coherent ising machine for 2000-node optimization problems. Science **354**(6312), 603–606 (2016)
25. Wen, J., Wang, Z., Huang, Z., Cai, D., Jia, B., Cao, C., Ma, Y., Wei, H., Wen, K., Qian, L.: Optical experimental solution for the multiway number partitioning problem and its application to computing power scheduling. Sci. China Phys., Mech. Astron. **66**(9), 290313 (2023)
26. Domino, K., Koniorczyk, M., Krawiec, K., Ja Lowiecki, K., Gardas, B.: Quantum computing approach to railway dispatching and conflict management optimization on single-track railway lines. arXiv preprint arXiv:2010.08227 (2020)
27. Domino, K., Kundu, A., Salehi, O., Krawiec, K.: Quadratic and higher-order unconstrained binary optimization of railway rescheduling for quantum computing. Quantum Inf. Process. **21**(9), 1–33 (2022)
28. 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)
29. Mahmoudi, M., Zhou, X.: Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: a dynamic programming approach based on state–space–time network representations. Transp. Res. Part B: Methodol. **89**, 19–42 (2016)
30. Li, W., Lee, J., Shroff, N.: A faster fptas for knapsack problem with cardinality constraint. Discret. Appl. Math. **315**, 71–85 (2022)
31. Li, W.: Performance analysis of modified SRPT in multiple-processor multitask scheduling. ACM Sigmetr. Perform. Eval. Rev. **50**(4), 47–49 (2023)
32. Yue, Y., Wang, S., Zhou, L., Tong, L., Saat, M.R.: Optimizing train stopping patterns and schedules for high-speed passenger rail corridors. Transp. Res. Part C: Emerging Technol. **63**, 126–146 (2016)
33. Magnanti, T.L., Wong, R.T.: Network design and transportation planning: models and algorithms. Transp. Sci. **18**(1), 1–55 (1984)
34. Pilon, G., Gugole, N., Massarenti, N.: Aircraft loading optimization–qubo models under multiple constraints. arXiv preprint arXiv:2102.09621 (2021)

35. Tsukamoto, S., Takatsu, M., Matsubara, S., Tamura, H.: An accelerator architecture for combinatorial optimization problems. Fujitsu Sci. Tech. J **53**(5), 8–13 (2017)
36. Roch, C., Impertro, A., Linnhoff-Popien, C.: Cross entropy optimization of constrained problem hamiltonians for quantum annealing. In: International Conference on Computational Science, pp. 60–73. Springer (2021)
37. Ayodele, M., Allmendinger, R., López-Ibá˜ nez, M., Parizy, M.: Multi-objective qubo solver: bi-objective quadratic assignment problem. In: Proceedings of the Genetic and Evolutionary Computation Conference, pp. 467–475 (2022)
38. Kjaergaard, M., Schwartz, M.E., Braumüller, J., Krantz, P., Wang, J.I.-J., Gustavsson, S., Oliver, W.D.: Superconducting qubits: current state of play. Ann. Rev. Condens. Matter Phys. **11**, 369–395 (2020)
39. Bruzewicz, C.D., Chiaverini, J., McConnell, R., Sage, J.M.: Trapped-ion quantum computing: progress and challenges. Appl. Phys. Rev. **6**(2), 021314 (2019)
40. National Academies of Sciences, Engineering, and Medicine.: Quantum computing: progress and prospects. Washington, DC. The National Academies Press. https://doi.org/10.17226/25196 (2019)
41. Glover, F., Kochenberger, G., Hennig, R., Du, Y.: Quantum bridge analytics I: a tutorial on formulating and using QUBO models. Ann. Oper. Res. **314**(1), 141–183 (2022)