Quantum-inspired optimization for wavelength assignment

Aleksey S. Boev, Sergey R. Usmanov*, Alexander M. Semenov,
Maria M. Ushakova, Gleb V. Salahov, Alena S. Mastiukova*,
Evgeniy O. Kiktenko and Aleksey K. Fedorov*

Russian Quantum Center, Skolkovo, Moscow, Russia

Problems related to wavelength assignment (WA) in optical communications networks involve allocating transmission wavelengths for known transmission paths between nodes that minimize a certain objective function, for example, the total number of wavelengths. Playing a central role in modern telecommunications, this problem belongs to NP-complete class for a general case so that obtaining optimal solutions for industry-relevant cases is exponentially hard. In this work, we propose and develop a quantum-inspired algorithm for solving the wavelength assignment problem. We propose an advanced embedding procedure to transform this problem into the quadratic unconstrained binary optimization (QUBO) form, having a improvement in the number of iterations with price-to-pay being a slight increase in the number of variables ("spins"). Then, we compare a quantum-inspired technique for solving the corresponding QUBO form against classical heuristic and industrial combinatorial solvers. The obtained numerical results indicate on an advantage of the quantum-inspired approach in a substantial number of test cases against the industrial combinatorial solver that works in the standard setting. Our results pave the way to the use of quantum-inspired algorithms for practical problems in telecommunications and open a perspective for further analysis of the use of quantum computing devices.

1 Introduction

Optimization is a tool with applications across various technologies [1]. However, solving complex real-world optimization problems is computationally intensive even in the case of using advanced, specialized hardware. Quantum computers are widely believed to be useful for solving computationally difficult optimization problems beyond the capability of existing computing devices to use quantum optimization [2–6]. A general approach consists in encoding a cost function in a quantum Hamiltonian [7] so that its low-energy state is obtained starting from a generic initial state. Among existing methods to achieve such dynamics, quantum annealing offers physical implementations of a non-trivial size [8]. Quantum annealing is by now explored for analysis of various areas, such as chemistry calculations [9, 10], lattice protein folding [11, 12], genome assembly [13, 14], solving polynomial systems of equations for engineering applications [15] and linear equations for regression [15], portfolio optimization [16–19], forecasting crashes [20], finding optimal trading trajectories [21], optimal arbitrage opportunities [22], optimal feature selection in credit scoring [23], foreign exchange reserves management [24], traffic optimization [25–27], scheduling [28–33], railway conflict management [32, 33], and many others [5]. Advances also include the recent experimental

demonstration of a super-linear quantum speedup in finding exact solutions for the hardest maximum independent set graphs [34].

Although quantum optimization algorithms suggest an intriguing possibility to solve computationally difficult problems beyond the capability of classical computers, existing conceptual and technical limitations make it challenging to use it for solving problems of industry-relevant sizes. Attempts to simulate quantum computations classically resulted in a new class of algorithms and techniques known as quantum-inspired techniques [35, 36]. As soon as these algorithms are compatible with currently existing (classical) hardware, analyzing their limiting capabilities and advantages over classical approaches is required toward their use in practice. Specifically, a way to solve combinatorial optimization problems *via* simulating the coherent Ising machine (SimCIM) has been proposed [35]. The SimCIM algorithm is able to solve optimization problems that are formulated in the quadratic unconstrained binary optimization (QUBO)/Ising form, which can be performed for various practically relevant cases [7]. The SimCIM approach has demonstrated capabilities to outperform the bona fide coherent Ising machine and existing classical methods for certain G-Set graphs. However, one of the arising questions is related to the need to tune hyperparameters [35]. For a wide range of the benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization, namely, the D-Wave hybrid solver service, Toshiba simulated bifurcation machine, Fujitsu digital annealer, and simulated annealing on a personal computer, see also [37].

The design of an optical communication network is a specific industrial avenue, in which combinatorial optimization is ubiquitous. Examples of tasks include finding optimal transmission and reservation paths, frequency allocation, network throughput maximization, and many others [38, 39]. A notable example is the routing and wavelength assignment (RWA) problem, which consists in allocating transmission wavelengths and finding transmission paths between nodes that minimize the total number of wavelengths. Conventional techniques, such as linear programming and mixed integer programming, are useful for most of the cases; however, the combinatorial nature and hardness of the problems make it extremely challenging to apply these techniques for large-scale problems. It is then reasonable to assume that the telecommunication industry may benefit from the use of a quantum-inspired algorithm in the near-term horizon and quantum computing in the future [40, 41].

In this study, we consider the variant of the RWA problem. To explain more precisely, we focus on the wavelength assignment task for known routes which we further refer to as the wavelength assignment (WA) problem. This problem is generally NP-hard, so its solution is computationally challenging for large sizes. We propose an original way to transform the WA problem to the QUBO form, which makes it compatible with the quantum-inspired optimization algorithm and, in principle, quantum annealing hardware. For solving this problem, we develop a technique based on the SimCIM quantum-inspired optimization solver [35] with the use of the Lagrange multipliers for minimizing the number of hyperparameters. Our numerical results indicate on an advantage of the quantum-inspired solver in a number of test cases against the industrial combinatorial solver working on the standard settings.

2 Wavelength assignment (WA) problem

Let us consider a network connecting a number of endpoints with optical links (see an example in Figure 1). Several endpoints that are interconnected by optical links sequentially comprise a path between a transmitter and a receiver. A single optical link can be shared between several paths, given that each path is assigned different wavelengths. Each path is indicated by the path ID, which uniquely identifies a pair of transmitting/receiving nodes, sequence of interconnecting nodes, and the wavelength ID.

The WA problem implies allocation of the wavelength IDs for paths that are pre-computed and known *a priori* in such a way to meet the target objective, for example, the number of the used wavelengths is minimized[1]. Formally, WA is considered to be correct if and only if it satisfies the following requirements: 1) each path should use a single wavelength and 2) several paths sharing the same edge should have different wavelengths.

The problem of finding correct wavelength allocation under given constraints is equivalent to the coloring problem [7] in a transformed graph $G = (V, E)$, where nodes V and edges E represent paths and their intersections in fibers, correspondingly (two nodes from V are connected if and only if the corresponding paths have an intersection within the optical network). Let N_V and N_E denote numbers of vertices and edges of G, respectively. Later, we interchangeably use the terms wavelengths and colors since the underlying problems are formally identical. The example of the correspondence of network paths to graph coloring mapping is shown in Figure 2.

In order to define a particular coloring of a graph G with at most W colors, we introduce two collections of auxiliary variables. The first variable is x that consists of $N_V W$ binary variables:

$$x_{vi} = \begin{cases} 1, & \text{if vertex } v \text{ is assigned wavelength } i, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

The second one, denoted by w, consists of W binary variables:

$$w_i = \begin{cases} 1, & \text{if } i-\text{th wavelength is assigned,} \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Using x and w, the problem of finding a correct allocation with a minimum number of the used wavelengths not exceeding some maximal numbers $W \geq 1$ can be formulated as an integer programming (IP) problem of the following form:

$$\sum_{i=1}^{W} w_i \to \min, \quad \text{s.t.,} \quad (3)$$

$$\sum_{i=1}^{W} x_{vi} = 1 \quad \forall v \in V, \quad (4)$$

$$x_{ui} + x_{vi} \leq w_i \quad \forall i \in \{1, \dots, W\}, \forall (u, v) \in E. \quad (5)$$

One can see that the constraint Eq. 4 assures that each vertex is assigned to exactly one wavelength, while the constraint Eq. 5 indicates that two adjacent vertices are not assigned the same wavelength.

This problem is generally NP-hard, so its solution is computationally challenging for large sizes. As it is shown in the following sections, the QUBO reduction makes the problem

---

1 We note that other objectives for optimization are also possible, such as total throughput or network resiliency.
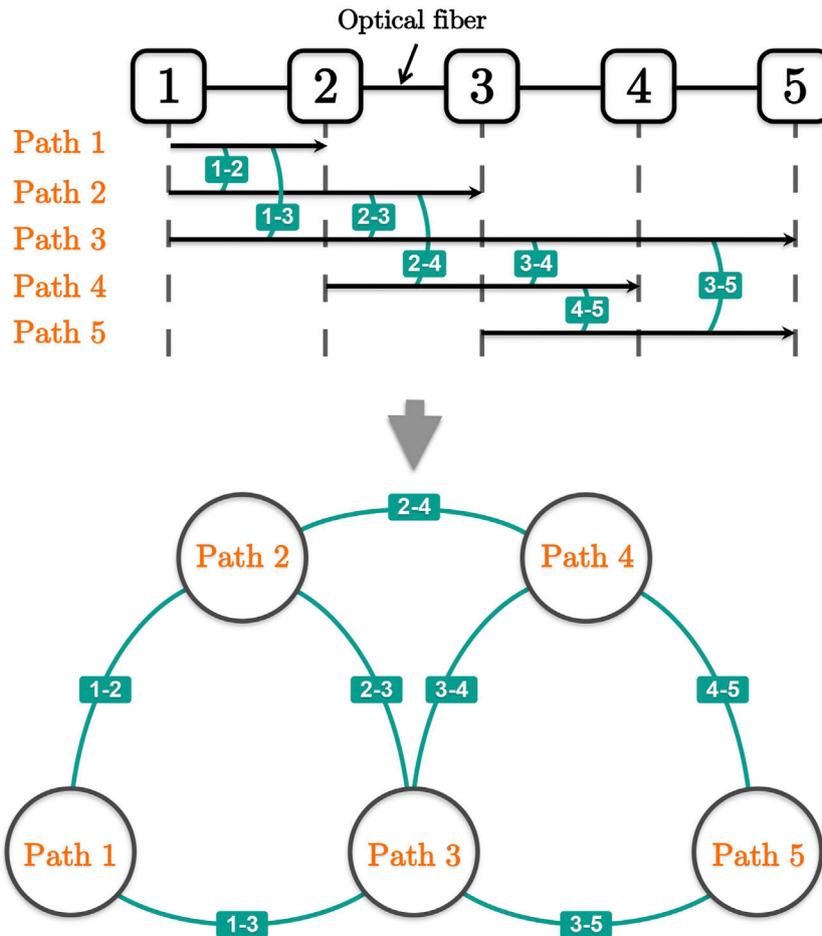
FIGURE 1
Illustration of the approach. A linear network with generated requests and paths consisting of five nodes, four edges, and five traffic paths is considered: Solid lines represent original edges, and the arrows represent traffic paths. One can reduce the WA problem to a graph coloring problem with a simple graph transformation (bottom of the figure): each traffic path is now considered a vertex; if two traffic paths share (at least) one fiber, they are connected by an edge.

compatible with quantum-inspired algorithms that can shift tractability boundaries to higher problem sizes. Although such reduction usually involves additional overheads in the problem size due to auxiliary variables, the overheads can be compensated by the computational advantage of quantum-inspired solvers, leading to better overall results.

3 Results

3.1 Transforming the WA problem to a QUBO form

In order to make the WA problem compatible with the SimCIM quantum-inspired optimization algorithm [35], we first consider a transformation, allowing one to convert the IP problem Eqs 3–5 into a QUBO form as follows:

$$\boldsymbol{s}^T Q \boldsymbol{s} \rightarrow \min \qquad (6)$$

for a certain binary vector $\boldsymbol{s}$ and the symmetric real matrix Q. This problem is equivalent to finding a configuration of binary-state particles ("spins") that minimizes the energy:

$$\mathcal{H}(\boldsymbol{s}) = \boldsymbol{s}^T Q \boldsymbol{s}, \qquad (7)$$

where the Ising Hamiltonian $\mathcal{H}$ consists of only single-order terms (energies of individual spins in an external magnetic field) and pair-wise interactions between spins. Although spin variables usually are considered to take values ± 1, the transition to a binary form is quite straightforward [13].

A known way [7] to transform a graph coloring problem to the QUBO form is to set $\boldsymbol{s} := \boldsymbol{x}$ (here we treat $\boldsymbol{x}$ as a $N_V W$-dimensional vector) and use the Hamiltonian of the form:

$$\mathcal{H}(\boldsymbol{x}) = \mathcal{H}_1(\boldsymbol{x}) + \mathcal{H}_2(\boldsymbol{x}), \qquad (8)$$

where

$$\mathcal{H}_1(\boldsymbol{x}) = \sum_{v=1}^{N_V} \left(1 - \sum_{i=1}^{W} x_{vi}\right)^2, \qquad (9)$$

$$\mathcal{H}_2(\boldsymbol{x}) = \sum_{(u,v) \in E} \sum_{i=1}^{W} x_{ui} x_{vi}. \qquad (10)$$

One can see that $\mathcal{H}_1(\boldsymbol{x}) > 0$ in the case where a single node is assigned with two distinct colors, while $\mathcal{H}_2(\boldsymbol{x}) > 0$ when two adjacent vertices are assigned the same color. If minimization routine provides some $\boldsymbol{x}$ such that $\mathcal{H}(\boldsymbol{x}) = 0$, then $\boldsymbol{x}$ defines a correct coloring with at
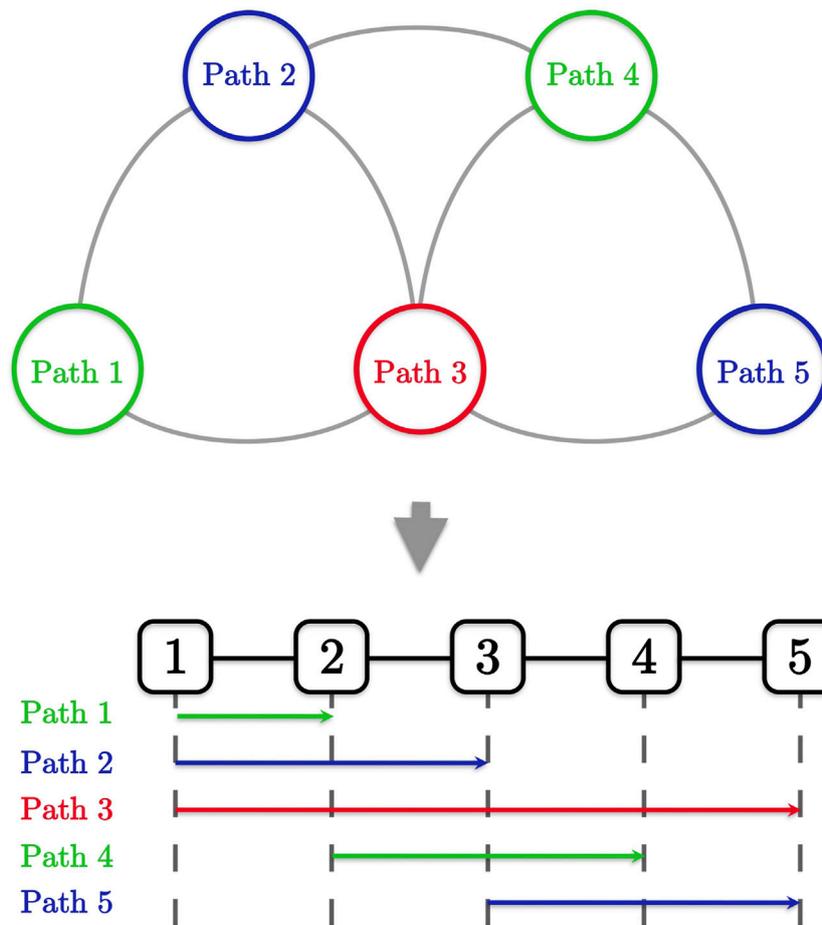
FIGURE 2
Example of a graph coloring problem and its representation to the network graph with requests.

most W colors. Therefore, an ability to solve the QUBO problem corresponding to the Hamiltonian (Eq. 8) guarantees one to solve a decision problem of whether it is possible to color a graph with at most W colors. Since it is always possible to color a graph with $W = N_V$ colors, a minimal number of colors can be obtained, for example, by using a standard binary search with at most $\lceil \log_2(N_V) \rceil$ iterations. We note that this approach is quite sensitive to possible imperfections of QUBO problem solutions, especially at first iterations of the binary search. An alternative way is to decrease W by a unit at each step that, however, results in a possible increase of iteration numbers up to $\mathcal{O}(W_{\text{start}})$, where W_{start} is the initial upper bound for a color number.

3.2 Improving QUBO transformation for quantum-inspired annealing

We propose an improved approach for solving a graph coloring problem by developing an alternative transformation into a QUBO form. In our approach, we pursue two major goals. The first is decreasing the number of QUBO problems to be solved. The second is making the whole algorithm robust against the possibility of finding not optimal, but some suboptimal solutions for a particular

QUBO problem. We note that these points are of particular importance in the framework of using (quantum-inspired) annealing for solving QUBO problems.

The main idea of our approach is to consider an extended $N_V(W+1)$-dimensional binary vector $\boldsymbol{s} := (\boldsymbol{w}, \boldsymbol{x})$ and take the target Hamiltonian in the following form:

$$\mathcal{H}(\boldsymbol{w}, \boldsymbol{x}) = c_0 \mathcal{H}_0(\boldsymbol{w}) + c_1 [\mathcal{H}_1(\boldsymbol{x}) + \mathcal{H}_2(\boldsymbol{x})] \\ + c_2 \mathcal{H}_3(\boldsymbol{w}, \boldsymbol{x}), \tag{11}$$

where

$$\mathcal{H}_0(\boldsymbol{w}) = \sum_{i=1}^{W} w_i, \tag{12}$$

$$\mathcal{H}_3(\boldsymbol{w}, \boldsymbol{x}) = \sum_{(u,v) \in E} \sum_{i=1}^{W} (1 - w_i)(x_{ui} + x_{vi}), \tag{13}$$

and c_i are positive coefficients satisfying a particular constraint (see more details in Section 5.1). Minimization of this Hamiltonian provides us the solution vector $(\boldsymbol{w}, \boldsymbol{x})$ such that the optimal number of wavelength is encoded in $\boldsymbol{w}$ by non-zero values. We note that the term $\mathcal{H}_0(\boldsymbol{w})$ grows with the total number of used wavelengths; $\mathcal{H}_1(\boldsymbol{x})$ and $\mathcal{H}_2(\boldsymbol{x})$ have the same form as in Eq. 8;

$\mathcal{H}_3(\boldsymbol{w}, \boldsymbol{x})$ is responsible for the relationship $w_i \geq x_{vi}$, which becomes positive when the relation is violated. Both terms $\mathcal{H}_2(\boldsymbol{x})$ and $\mathcal{H}_3(\boldsymbol{w}, \boldsymbol{x})$ correspond to inequalities in Eq. 5 in the IP form (see Section 5).

The complete algorithm of solving a graph coloring problem (WA problem) is shown in Algorithm 1. The algorithm uses a subroutine make_qubo(G, W) that generates the corresponding QUBO matrix Q with respect to the Hamiltonian (Eq. 11), given the input graph G and the target number of the wavelengths W. The QUBO problem is then solved with the subroutine solve_qubo(Q), which finds the optimal spin vector $\boldsymbol{s} = (\boldsymbol{w}, \boldsymbol{x})$ using the quantum-inspired SimCIM approach for the QUBO matrix Q, as defined in [35]. In order to check the validness of the obtained solution, we use check_coloring$(G, \boldsymbol{x})$ that validates the fulfillment of Eqs 4, 5.

Require: W is the initial upper bound on the number of
  wavelengths
Require: make_qubo$(G, W) \rightarrow Q$
Require: solve_qubo$(Q) \rightarrow (\boldsymbol{w}, \boldsymbol{x})$
Require: check_coloring$(G, \boldsymbol{x}) \rightarrow$ true/false
  returns true if coloring is correct

1: $\boldsymbol{x}^{\text{opt}} := \boldsymbol{0}$ ▷ initializing solution variable
2: $W' := W$ ▷ current number of colors
3: **while** $W' \geq 1$ **do**
4:     $Q := $ make_qubo(G, W')
5:     $(\boldsymbol{w}, \boldsymbol{x}) := $ solve_qubo(Q)
6:     **if** check_coloring$(G, \boldsymbol{x}) = $ true **then**
7:         $W' := \sum_{i=1}^{W'} w_i - 1$
8:         $\boldsymbol{x}^{\text{opt}} := \boldsymbol{x}$
9:     **else**
10:         break
11: **return** $\boldsymbol{x}^{\text{opt}}$

Algorithm 1. Solving graph coloring problem with improved transformation:

One can see that, if solve_qubo(Q) provides an optimal solution, then the whole problem is solved in the first iteration. However, even in the case when the obtained solution is suboptimal, the updated problem with the reduced upper bound W becomes easier to solve, and the algorithm converges with a few numbers of iterations.

3.3 Numerical results

In this study, we solve the WA problem and obtain results with the use of 1) the proposed technique based on quantum-inspired optimization SimCIM [35] (with the improved approach, see Section 5), 2) industry-grade commercial Gurobi optimization software, and 3) the open-source mixed-integer programming solver—GLPK. We note that in the case of the quantum-inspired optimization with SimCIM, we solve the problem in the QUBO form (Eq. 11), whereas in the case of Gurobi and GLPK, we use the IP formulation of graph coloring [see (Eqs 3–5)]. Additionally, we include the results obtained via the largest-degree-first (LDF) heuristics used as the baseline since it allows one to instantly produce feasible coloring without numerical optimization. We also ran the experiments for original QUBO transformation proposed in [7] and compared them to our proposed QUBO in the Table A1.

Our numerical experiments have been performed on a synthetic dataset of 900 randomly generated graphs with varying node numbers and edge probabilities (for details, see Section 5.3). The main characteristics that we are interested in are time-to-solution (TTS)

TABLE 1 Numerical results obtained with the largest-degree-first (LDF) heuristics, open-source mixed-integer programming solver (GLPK), Gurobi optimization software, and SimCIM quantum-inspired optimization on the number of colors averaged by the number of nodes.

Number of nodes	LDF	GLPK	Gurobi	SimCIM
10	4.46	4.34	4.34	4.34
20	6.82	6.36	6.36	6.36
30	9.03	8.03	8.02	8.02
40	10.92	—	**9.38**	9.39
50	12.80	—	**10.88**	10.96
60	14.83	—	**12.28**	12.44
70	16.62	—	**13.70**	14.01
80	18.41	—	**15.34**	15.56
90	20.10	—	17.21	**17.02**
100	22.01	—	19.64	**18.54**
Average number of colors				
(Lower is better)				

The best result is highlighted in bold.

and the number of colors in the obtained solution. The total run time has been limited by 300 s, and the best solutions have been compared. Results are averaged over 90 runs for each graph size (for details, see Table 1). For all numerical experiments, we use the same hardware set, which is based on Xeon E-2288G 3.7 GHz CPU, 128GB RAM, and GeForce GTX1080 8 GB graphics card.

Our results indicate that the quantum-inspired technique SimCIM demonstrates a behavior comparable with Gurobi in the case of small nodes (10–30 nodes). Moreover, the run time of SimCIM is better for large-scale (90 and 100 nodes) graphs, as it is indicated in Table 2. Such a trend can be explained. As the number of nodes increases, the number of inequalities in the ILP formulation of the problem grows rapidly. The number of inequalities is equal to the product of the number of edges by the number of colors available for coloring the vertices of the graph. So, the complexity of the problem for the ILP solver increases rapidly with the number of nodes. GLPK shows a stable result up to 30 nodes and becomes unstable further after a timeout interruption without any solution with more than 10 percent instances. We note that the comparison between our quantum-inspired approach and Gurobi is conducted in the common setting, so its additional tuning for obtaining better results is also possible. At the same time, we find it interesting that the quantum-inspired technique shows comparable or superior results in harder, industry-relevant, combinatorial optimization problem.

3.4 Other potential applications

Although our goal was to demonstrate the applicability of a quantum-inspired graph coloring algorithm for the wavelength assignment problem, our approach can be applied to a variety of problems, in particular from the field of scheduling [42].

Assuming that we have the set of jobs to schedule, every job requires one time slot and some jobs cannot be executed at the same

TABLE 2 Mean solution time depending on the number of nodes for GLPK, Gurobi, and SimCIM. The best result is highlighted in bold.

Number of nodes	GLPK	Gurobi	SimCIM
10	1.77	.002	.19
20	103.97	.02	.45
30	195.39	.12	4.95
40	—	**.79**	8.90
50	—	**14.63**	16.82
60	—	38.89	28.51[a]
70	—	66.01	61.58[a]
80	—	102.14	69.00[a]
90	—	144.23	**79.87**
100	—	127.33	**123.13**
Average time (seconds)			
(Lower is better)			

[a]Cases, where the average number of colors is higher.

time due to some interference with each other; we need to determine the minimal time when every job will be finished or how many time slots they will occupy. One can build the graph so that vertices correspond to the jobs, and two vertices are connected if these jobs cannot be executed at the same time. The colors of vertices represent the time slots to assign; so, a graph has k number of colors if the jobs can be executed in k time slots.

Using our approach, we take the proposed Hamiltonian in Eq. 11 and redefine its variables so that the following expression is obtained:

$$x_{vi} = \begin{cases} 1, & \text{if vertex } v \text{ is assigned time slot } i, \\ 0, & \text{otherwise,} \end{cases} \qquad (14)$$

and

$$w_i = \begin{cases} 1, & \text{if } i - \text{th time slot is assigned,} \\ 0, & \text{otherwise.} \end{cases} \qquad (15)$$

That way, the job-scheduling problem can be solved using quantum-inspired annealing analogously to the WA problem.

The same approach can be implemented for tasks from other fields, such as computer register allocation [43], storage of chemicals [44], and printed circuit board testing [45].

4 Conclusion

A search for new approaches to solving practically relevant optimization problems is a clear goal for many industial applications since even minor improvement on a large scale may generate serious economic impact. In this domain, much attention is paid to quantum computing, which is believed to be useful for such class of problems. At the current technological level, practical quantum advantage, for example, in optimization is still needed to be achieved. An interesting part of this research is the understanding of the physical origin of the potential advantages of quantum computing technologies. Attempts to simulate quantum computation classically resulted in a new class of algorithms and

methods known as quantum-inspired methods, which are ready to be tested for industry-relevant problems.

In this work, we have considered the industry-relevant problem in the field of telecommunications. We have demonstrated a way to make it compatible with the quantum and quantum-inspired techniques. Interestingly, our numerical results have indicated on an advantage of the quantum-inspired solver in a number of test cases against the industrial combinatorial solver working on the standard settings.

One may expect that the additional tuning of the industry-grade commercial optimization solver may result in a substantial improvement of its performance. At the same time, studying the origins of the advantages of the quantum-inspired approach, which are largely beyond the scope of the present proof-of-concept demonstration, would allow its further progress as well.

We would like to note that our comparison is limited by the upper bound of 100 nodes since it allows us to run all solvers in equivalent hardware setup using the CPU mode on a single core. Further analysis of larger graphs requires running the SimCIM solver on a GPU card, which gives the significant acceleration factor not directly available in conventional MIP algorithms, which are heavily dependent on graph-processing routines. As for the multi-core CPU execution environment, some MIP solvers can benefit from such a set-up by running various optimization strategies and hyperparameters simultaneously. Such speed up quickly reaches the saturation point at the level of 8–16 cores (with around 2x improvement in accordance with Gurobi experiments, see slide 26 [46]) and demonstrates no substantial improvement at higher concurrency levels. On the other hand, a quantum-inspired approach exploits parallelism on the level of starting optimization points, which demonstrates slower, but stable performance increase at the higher levels of concurrency (100 ~ 1,000 parallel units of execution). Thus, we conduct our benchmarks exclusively using the CPU mode on a single core to avoid bias toward either the solution approach. In order to maintain fairness of comparison for larger graphs, our benchmark routine should be further revised to account for heterogeneous (CPU/CPU multi-core vs. GPU/multi-GPU) computing environments.

5 Methods

5.1 Hamiltonian of the wavelength assignment problem

The main step in solving an optimization problem using the quantum and quantum-inspired annealing method is to map the problem of interest to the energy Hamiltonian (so-called Ising Hamiltonian) so that the quantum device could find the ground state that corresponds to the optimum value of the objective function. In this study, we formulate a mapping of the graph coloring problem into the QUBO form given by Eq. 6. There is a well-known transformation of the graph $G = (V, E)$ coloring decision model [7] that shows the possibility of coloring with a constant number of colors W, but we represent novel QUBO transformation that could minimize the number of colors and implement the original problem statement (Eqs 3–5).

The objective function $\sum_{i=1}^{W} w_i$ could be exactly mapped to the QUBO form:

$$\mathcal{H}_0(\boldsymbol{w}) = \sum_{i=1}^{W} w_i, \tag{16}$$

where $\boldsymbol{w} = (w_1, ..., w_W)$ is a binary vector indicating colors used in coloring. The constraint $\sum_{i=1}^{W} x_{vi} = 1$ for every $v \in V$ after mapping takes the form

$$\mathcal{H}_1(\boldsymbol{x}) = \sum_{v=1}^{N_V} \left(1 - \sum_{i=1}^{W} x_{vi}\right)^2, \tag{17}$$

where N_V is the number of nodes in G.

The situation with the second constraint $x_{ui} + x_{vi} \le w_i$ for every $i \in \{1, ..., W\}$ and $(u, v) \in E$ appears to be more complicated. One can see that it involves three variables and thus cannot be directly embedded into a two-body Hamiltonian. However, we can use the following trick. One can easily check that for arbitrary $a, b, c \in \{0, 1\}$, the following equivalence holds:

$$a + b \le c \Leftrightarrow \begin{cases} ab = 0, \\ (1 - c)(a + b) = 0. \end{cases} \tag{18}$$

This fact allows us to embed the conditions $x_{ui} + x_{vi} \le w_i$ into two Hamiltonians:

$$\mathcal{H}_2(\boldsymbol{x}) = \sum_{(u,v) \in E} \sum_{i=1}^{W} x_{ui} x_{vi}, \tag{19}$$

$$\mathcal{H}_3(\boldsymbol{w}, \boldsymbol{x}) = \sum_{(u,v) \in E} \sum_{i=1}^{W} (1 - w_i)(x_{ui} + x_{vi}). \tag{20}$$

The resulting Hamiltonian consists of all components' sum:

$$\mathcal{H}(\boldsymbol{w}, \boldsymbol{x}) = c_0 \mathcal{H}_0(\boldsymbol{w}) + c_1 [\mathcal{H}_1(\boldsymbol{x}) + \mathcal{H}_2(\boldsymbol{x})] \\ + c_2 \mathcal{H}_3(\boldsymbol{w}, \boldsymbol{x}), \tag{21}$$

where c_0, c_1, and c_2 are positive constants standing for a positive penalty value. We note that the sum $\mathcal{H}_1(\boldsymbol{x}) + \mathcal{H}_2(\boldsymbol{x})$ is exactly matched with the classical decision problem [7] and responsible for the correct coloring of the graph. Therefore, $\mathcal{H}_1(\boldsymbol{x})$ and $\mathcal{H}_2(\boldsymbol{x})$ are grouped with the same penalty coefficient c_1. Coefficients c_0, c_1, and c_2 should be set manually using the following criteria: the penalty value c_1 should be high enough to the keep the final solution from violating constraints. At the same time, a very big penalty value can overwhelm the target function, making it difficult to distinguish solutions of different qualities. We establish inequalities for constraint coefficients that show the equivalence of IP and QUBO models of a problem.

5.1.1 Proposition (QUBO penalty coefficient selection)

Consider an IP problem given by Eqs 3–5 for a maximal color number W and a graph $G = (V, E)$ with N_E edges. If the IP problem has a solution, then the corresponding QUBO problem, given by Hamiltonian (Eq. 21) with penalty coefficients satisfying

$$c_1 > 2 N_E W c_2 + W c_0, \tag{22}$$

$$c_2 > W c_0, \tag{23}$$

has a solution, equivalent to the solution of the IP problem.

5.1.2 Proof

First, let us rewrite the Hamiltonian (Eq. 21) in the following form:

$$\mathcal{H}(\boldsymbol{w}, \boldsymbol{x}) = c_0 \mathcal{A}(\boldsymbol{w}) + c_1 \mathcal{B}(\boldsymbol{x}) + c_2 \mathcal{C}(\boldsymbol{w}, \boldsymbol{x}), \tag{24}$$

where

$$\begin{aligned} \mathcal{A}(\boldsymbol{w}) &\coloneqq \mathcal{H}_0(\boldsymbol{w}), \\ \mathcal{B}(\boldsymbol{x}) &\coloneqq \mathcal{H}_1(\boldsymbol{x}) + \mathcal{H}_2(\boldsymbol{x}), \\ \mathcal{C}(\boldsymbol{w}, \boldsymbol{x}) &\coloneqq \mathcal{H}_3(\boldsymbol{w}, \boldsymbol{x}). \end{aligned} \tag{25}$$

Note that $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ can take non-negative integer values only. Let $(\boldsymbol{w}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{I}})$ and $(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}})$ be solutions of the IP and QUBO problems correspondingly. Our goal is to prove the following: (i)

$$\mathcal{B}(\boldsymbol{x}_{\mathrm{Q}}) = \mathcal{C}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}) = 0, \tag{26}$$

i.e., $(\boldsymbol{x}_{\mathrm{Q}}, \boldsymbol{w}_{\mathrm{Q}})$ defines a correct coloring, and (ii)

$$\mathcal{A}(\boldsymbol{w}_{\mathrm{Q}}) = \sum_{i=1}^{W} (\boldsymbol{w}_{\mathrm{I}})_i, \tag{27}$$

i.e., the solution of the QUBO problem coincides with the one of the IP problem.

First, let us see that Eq. 22 assures $\mathcal{B}(\boldsymbol{x}_{\mathrm{Q}}) = 0$. The proof of this part is by a contradiction. Let us suppose that $\mathcal{B}(\boldsymbol{x}_{\mathrm{Q}}) \ge 1$. Consider the difference of energy functions:

$$\begin{aligned} \Delta \mathcal{H} &\coloneqq \mathcal{H}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}) - \mathcal{H}(\boldsymbol{w}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{I}}) \\ &= c_0 [\mathcal{A}(\boldsymbol{w}_{\mathrm{Q}}) - \mathcal{A}(\boldsymbol{w}_{\mathrm{I}})] + c_1 [\mathcal{B}(\boldsymbol{x}_{\mathrm{Q}}) - \mathcal{B}(\boldsymbol{x}_{\mathrm{I}})] \\ &\quad + c_2 [\mathcal{C}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}) - \mathcal{C}(\boldsymbol{w}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{I}})]. \end{aligned} \tag{28}$$

The correctness of the IP solution implies $\mathcal{B}(\boldsymbol{x}_{\mathrm{I}}) = 0$, and so $\mathcal{B}(\boldsymbol{x}_{\mathrm{Q}}) - \mathcal{B}(\boldsymbol{x}_{\mathrm{I}}) \ge 1$. The differences in terms with $\mathcal{A}$ and $\mathcal{C}$ can be lower-bounded by the corresponding extreme values:

$$\mathcal{A}(\boldsymbol{w}_{\mathrm{Q}}) - \mathcal{A}(\boldsymbol{w}_{\mathrm{I}}) \ge -W, \tag{29}$$

$$\mathcal{C}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}) - \mathcal{C}(\boldsymbol{w}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{I}}) \ge -2 N_E W. \tag{30}$$

In this way, Eq. 28 transforms into the following:

$$\Delta \mathcal{H} \ge -c_0 W + c_1 - 2 c_2 N_E W > 0, \tag{31}$$

given constraint (Eq. 22). However, this result contradicts with the fact that $(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}})$ provides the minimal energy. Therefore, $\mathcal{B}(\boldsymbol{x}_{\mathrm{Q}}) = 0$, and

$$\mathcal{H}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}) = c_0 \mathcal{A}(\boldsymbol{w}_{\mathrm{Q}}) + c_2 \mathcal{C}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}). \tag{32}$$

We then prove that $\mathcal{C}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}})$ is zero as well. Indeed, if $\mathcal{C}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}) \ge 1$, then

$$\begin{aligned} \Delta \mathcal{H} &= c_0 [\mathcal{A}(\boldsymbol{w}_{\mathrm{Q}}) - \mathcal{A}(\boldsymbol{w}_{\mathrm{I}})] \\ &\quad + c_2 [\mathcal{C}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}) - \mathcal{C}(\boldsymbol{w}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{I}})] \\ &\ge -c_0 W + c_2 > 0, \end{aligned} \tag{33}$$

provided $\mathcal{C}(\boldsymbol{w}_{\mathrm{I}}, \boldsymbol{x}_{\mathrm{I}}) = 0$ and the second constraint (Eq. 23). Thus, $\mathcal{H}(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}}) = c_0 \mathcal{A}(\boldsymbol{w}_{\mathrm{Q}})$.

Finally, $\mathcal{A}(\boldsymbol{w}_{\mathrm{Q}}) = \mathcal{A}(\boldsymbol{w}_{\mathrm{I}})$ since otherwise, either there exists a solution for the QUBO problem that is better than $(\boldsymbol{w}_{\mathrm{Q}}, \boldsymbol{x}_{\mathrm{Q}})$ or $(\boldsymbol{x}_{\mathrm{I}}, \boldsymbol{w}_{\mathrm{I}})$ is not the true solution for the IP problem.

Therefore, the optimal solution to the QUBO problem appears to be equivalent to the optimal solution to the corresponding IP problem.

5.2 Wavelength assignment QUBO transformation

In this study, we demonstrate how to construct an operator matrix Q of our QUBO model for the WA problem. Recall that we take the

TABLE 3 Characteristics of the graph coloring dataset; the total number of instances is 900.

Number of nodes	Number of edges		QUBO matrix size	
	Min	Max	Min	Max
10	9	43	44	110
20	23	176	84	315
30	39	399	124	589
40	74	714	205	943
50	118	1,123	255	1,377
60	168	1,625	366	1,891
70	231	2,209	426	2,556
80	301	2,879	486	3,321
90	372	3,652	546	4,004
100	470	4,501	707	4,848

binary vector of the QUBO problem in the form $s = (w, x)$, i.e., enumerate $K = (N_V+1)W$ and binary variables s_k and link them to our model variables as follows:

$$s_k = \begin{cases} w_k, & k = 1, \ldots, W, \\ x_{ui}, & k = uW + i, \end{cases} \tag{34}$$

where $u = 1, \ldots, N_V, i = 1, \ldots, W$.

The goal is to find the vector s that minimizes the quadratic form $s^T Q s$, and we show that it is equivalent to minimizing energy of the Hamiltonian (Eq. 11). Let us denote A the adjacency matrix of the network graph $G = (V, E)$ so that $a_{uv} = 1$ if $(u, v) \in E$, and $a_{uv} = 0$ otherwise. We note that the sum of the vth column of A equals the degree of the vertex v, and the sum of all vertex degrees is $2N_E$. We rewrite the operator (Eq. 11) terms $\mathcal{H}_0(w), \mathcal{H}_1(x), \mathcal{H}_2(x)$, and $\mathcal{H}_3(w, x)$ as follows:

$$\mathcal{H}_0(w) = \sum_{i=1}^{W} w_i^2, \tag{35}$$

$$\mathcal{H}_1(x) = \sum_{v=1}^{N_V} \left(1 - \sum_{i=1}^{W} x_{vi}\right)^2$$

$$= \sum_{v=1}^{N_V} \left(\left(\sum_{i=1}^{W} x_{vi}\right)^2 - 2\sum_{i=1}^{W} x_{vi}\right) + N_V \tag{36}$$

$$= \sum_{v=1}^{N_V} \left(\sum_{i,j=1}^{W} x_{vi} x_{vj} - 2\sum_{i=1}^{W} x_{vi}^2\right) + N_V,$$

$$\mathcal{H}_2(x) = \sum_{(u,v)\in E} \sum_{i=1}^{W} x_{ui} x_{vi} = \sum_{u,v=1}^{N_V} \sum_{i=1}^{W} a_{uv} x_{ui} x_{vi}, \tag{37}$$

$$\mathcal{H}_3(w, x) = \sum_{(u,v)\in E} \sum_{i=1}^{W} (1 - w_i)(x_{ui} + x_{vi})$$

$$= \sum_{i=1}^{W} (1 - w_i) \sum_{v=1}^{N_V} d_v x_{vi} \tag{38}$$

$$= -\sum_{v=1}^{N_V} \sum_{i=1}^{W} d_v w_i x_{vi} + \sum_{v=1}^{N_V} d_v \sum_{i=1}^{W} x_{vi}.$$

In expanding the expression for $\mathcal{H}_1(x)$, we exploit the fact that since x_{vi} is binary, then $x_{vi}^2 = x_{vi}$. Also, we note that if $\mathcal{H}_1(x) = 0$, then the last term in $\mathcal{H}_3(w, x)$ equals $2N_E$.

Considering the equalities (Eqs 35–38) for Hamiltonian terms $\mathcal{H}_0(x), \mathcal{H}_1(x), \mathcal{H}_2(x)$ and $\mathcal{H}_3(w, x)$, we construct a QUBO operator as a block matrix as follows:

$$Q = \begin{pmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{pmatrix}, \tag{39}$$

where

$$Q_{11} = c_0 E_W, \tag{40}$$

$$Q_{12} = -\frac{c_2}{2} D \otimes E_W, \quad Q_{21} = Q_{12}^T, \tag{41}$$

$$Q_{22} = c_1 E_{N_V} \otimes (I_W - 2E_W) + c_1 A \otimes E_W. \tag{42}$$

Here, E_W denotes the identity matrix of size W, I_W denotes a matrix with all elements equal to 1 of those of size W, and $D = (d_1, ..., d_{N_V})$ is a row vector of graph vertex degrees. We also employ the fact that the terms of the form

$$\sum_{u,v=1}^{N_V} \sum_{i,j=1}^{W} c_{uv} h_{ij} x_{ui} x_{vj}, \tag{43}$$

for some coefficients $c_{uv} = c_{vu}$ and $h_{ij} = h_{ji}$ can be represented by a quadratic form defined by the Kronecker product $C \otimes H$, where C and H are matrices of c_{uv} and h_{ij}, respectively. Matrix Q is constructed so that the Q_{11} submatrix corresponds to the term $\mathcal{H}_0(x)$ of the Hamiltonian (Eq. 11), the Q_{12} submatrix is for $\mathcal{H}_3(w, x)$, and Q_{22} is for $\mathcal{H}_1(x) + \mathcal{H}_2(x)$.

It is worth emphasizing that it is the structure of the encoding problem parameters into the spin vector, given by Eq. 34, that allows us to represent submatrices Q_{12}, Q_{21}, and Q_{22} in the form of Kronecker products. This feature of QUBO submatrices significantly speeds up their assembly using standard mathematical packages, e.g., numpy and scipy.

5.3 Dataset generation

We generate datasets that are used in binomial graphs [47], or Erdös–Rényi graphs, which have two parameters for generation: the number of nodes N_V and the probability of an edge occurrence p. Each of possible $N = N_V \cdot (N_V - 1)/2$ edges is chosen with probability p. The number of edges N_E is drawn randomly from the binomial distribution:

$$P(N_E = x) = \binom{N}{x} p^x \cdot q^{(N-x)}. \tag{44}$$

To take into account sparse and dense graphs, various probability p options from .1 to .9 with an interval of .1 have been chosen; the number of graph nodes has been varied 10 to 100 with a step of 10. For each pair (n, p), 10 connected graphs have been generated with different seed parameters. We note that disconnected graphs are not included the dataset. The overall characteristics of the dataset are given in Table 3.

5.4 Setting penalty values

Optimal penalty values guarantee the fulfillment of constraints for an optimal solution, but large values of c_1 and c_2 reduce the contribution of the initial objective function to the total energy and

significantly increase the time to find the optimal solution. Our approach to solve this problem is as follows:

1. Set the minimum possible penalty values c_1 and c_2 using trial runs so that the contribution of the objective function is sufficient.
2. Use all SimCIM iterations to select feasible solutions.
3. Take the feasible solution with the lowest energy.

The following penalty values were set for the tests:

$$c_0 = 1, \ c_1 = 10 + pN_V, \ c_2 = 2.5. \tag{45}$$

5.5 Quantum-inspired annealing using SimCIM

SimCIM [35] is an example of a quantum-inspired annealing algorithm, which works in an iterative manner. SimCIM can be used for sampling low-energy spin configurations in the classical Ising model, and the Hamiltonian can be written as follows:

$$\mathcal{H} = \sum_i h_i s_i + \sum_{<i,j>} J_{ij} s_i s_j, \tag{46}$$

where J represents the spin–spin interaction, h represents the external field, and s_i are the individual spins on each of the lattice sites. The Ising Hamiltonian can be directly transformed to a QUBO problem [13], and then, quantum annealing can be applied to any optimization problem, which can be expressed into the Quadratic Unconstrained Binary Optimization (QUBO) form. The SimCIM algorithm treats each spin value as a continuous variable $s_i \in [-1, 1]$. Each iteration of the algorithm starts with calculating the mean field of the following form:

$$\Phi_i = \sum_{j \neq i} J_{ij} s_j + h_i, \tag{47}$$

which acts on each spin by all other spins. Then, the gradients for the spin values are calculated as follows:

$$\Delta s_i = p_t s_i + \zeta \Phi_i + N(0, \sigma), \tag{48}$$

where p_t is a dynamic parameter dependent on the SimCIM annealing process, the overall feed forward factor is ζ, and $N(0, \sigma)$ is a random variable sampled from the Gaussian distribution with zero mean and standard deviation σ. Then, the spin values are updated according to $s_i \leftarrow \phi(s_i + \Delta s_i)$, where $\phi(x)$ is the activation function:

$$\phi(x) = \begin{cases} x \text{ for } |x| \leq 1; \\ x/|x|, \text{ otherwise}. \end{cases} \tag{49}$$

After multiple updates, the spins will tend to either -1 or $+1$, and the final discrete spin configuration is obtained by taking the sign of each s_i.

In our implementation, we added several improvements to the SimCIM algorithm defined in the original paper [35]. In particular, we normalized the value of the Gaussian noise to a gradient norm and introduced gradient quantization, which made the solver more stable near optimum points.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

Author contributions

The development of a quantum-based algorithm to solve the RWA problem was made by AB and SU. Also, an improved embedding procedure for this problem in the form of quadratic unbounded binary optimization (QUBO) was implemented by AS, MU, and GS. In addition, the part of this work related to the analysis of the quantum-inspired optimization algorithm was made by AM, EK, and AF. All authors made a significant contribution to the work on obtaining the results and writing the article.

Funding

Acknowledgments

Conflict of interest

Owing to the employments and consulting activities of authors, the authors have financial interests in the commercial applications of quantum computing. The authors do not have any non-financial competing interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary Material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fphy.2022.1092065/full#supplementary-material

References

1. Paschos VT. *Paradigms of combinatorial optimization*. 2nd ed. London: Hoboken: John Wiley & Sons (2014). ISTE.

2. Farhi E, Goldstone J, Gutmann S, Sipser M. Quantum computation by adiabatic evolution (2000). Available at: https://arxiv.org/abs/quant-ph/0001106.

3. Das A, Chakrabarti BK. Colloquium: Quantum annealing and analog quantum computation. *Rev Mod Phys* (2008) 80:1061–81. doi:10.1103/RevModPhys.80.1061

4. Albash T, Lidar DA. Adiabatic quantum computation. *Rev Mod Phys* (2018) 90:015002. doi:10.1103/RevModPhys.90.015002

5. Fedorov AK, Gisin N, Beloussov SM, Lvovsky AI. Quantum computing at the quantum advantage threshold: A down-to-business review (2022). Available at: https://arxiv.org/abs/2203.17181.

6. Farhi E, Goldstone J, Gutmann S. A quantum approximate optimization algorithm (2014). Available at: https://arxiv.org/abs/1411.4028.

7. Lucas A. Ising formulations of many NP problems. *Front Phys* (2014) 2:5. doi:10.3389/fphy.2014.00005

8. King AD, Raymond J, Lanting T, Isakov SV, Mohseni M, Poulin-Lamarre G, et al. Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets. *Nat Commun* (2021) 12:1113. doi:10.1038/s41467-021-20901-5

9. Streif M, Neukart F, Leib M. Solving quantum chemistry problems with a d-wave quantum annealer. In: Feld S, Linnhoff-Popien C, editors. *Quantum Technology and optimization problems*. Cham: Springer International Publishing (2019). p. 111–22.

10. Chermoshentsev DA, Malyshev AO, Esencan M, Tiunov ES, Mendoza D, Aspuru-Guzik A, et al. Polynomial unconstrained binary optimisation inspired by optical simulation (2021). 2106.13167.

11. Perdomo-Ortiz A, Dickson N, Drew-Brook M, Rose G, Aspuru-Guzik A. Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific Rep* (2012) 2:571. doi:10.1038/srep00571

12. Babej T, Ing C, Fingerhuth M. Coarse-grained lattice protein folding on a quantum annealer (2018). 1811.00713.

13. Boev AS, Rakitko AS, Usmanov SR, Kobzeva AN, Popov IV, Ilinsky VV, et al. Genome assembly using quantum and quantum-inspired annealing. *Scientific Rep* (2021) 11:13183. doi:10.1038/s41598-021-88321-5

14. Sarkar A, Al-Ars Z, Bertels K. QuASeR: Quantum Accelerated de novo DNA sequence reconstruction. *PLOS ONE* (2021) 16:e0249850. doi:10.1371/journal.pone.0249850

15. Chang CC, Gambhir A, Humble TS, Sota S. Quantum annealing for systems of polynomial equations. *Scientific Rep* (2019) 9:10258. doi:10.1038/s41598-019-46729-0

16. Orús R, Mugel S, Lizaso E. Quantum computing for finance: Overview and prospects. *Rev Phys* (2019) 4:100028. doi:10.1016/j.revip.2019.100028

17. Mugel S, Kuchkovsky C, Sanchez E, Fernandez-Lorenzo S, Luis-Hita J, Lizaso E, et al. Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks (2020). 2007.00017.

18. Grant E, Humble TS, Stump B. Benchmarking quantum annealing controls with portfolio optimization. *Phys Rev Appl* (2021) 15:014012. doi:10.1103/PhysRevApplied.15.014012

19. Herman D, Googin C, Liu X, Galda A, Safro I, Sun Y, et al. A survey of quantum computing for finance (2022). Available at: https://arxiv.org/abs/2201.02773.

20. Orús R, Mugel S, Lizaso E. Forecasting financial crashes with quantum computing. *Phys Rev A* (2019) 99:060301. doi:10.1103/PhysRevA.99.060301

21. Rosenberg G, Haghnegahdar P, Goddard P, Carr P, Wu K, de Prado ML. Solving the optimal trading trajectory problem using a quantum annealer. *IEEE J Selected Top Signal Process* (2016) 10:1053–60. doi:10.1109/JSTSP.2016.2574703

22. Rosenberg G. Finding optimal arbitrage opportunities using a quantum annealer (2016). Available at: https://1qbit.com/.

23. Andrew Milne MR, Goddard P. Optimal feature selection in credit scoring and classification using a quantum annealer (2017). Available at: https://1qbit.com/.

24. Vesely M. Application of quantum computers in foreign exchange reserves management (2022). Available at: https://arxiv.org/abs/2203.15716.

25. Neukart F, Compostella G, Seidel C, von Dollen D, Yarkoni S, Parney B. Traffic flow optimization using a quantum annealer. *Front ICT* (2017) 4:29. URL. doi:10.3389/fict.2017.00029

26. Inoue D, Okada A, Matsumori T, Aihara K, Yoshida H. Traffic signal optimization on a square lattice with quantum annealing. *Scientific Rep* (2021) 11:3303. doi:10.1038/s41598-021-82740-0

27. Hussain H, Javaid MB, Khan FS, Dalal A, Khalique A. Optimal control of traffic signals using quantum annealing. *Quan Inf Process* (2020) 19:312. doi:10.1007/s11128-020-02815-1

28. Venturelli D, Marchand DJJ, & Rojo G Quantum annealing implementation of job-shop scheduling (2016). 1506.08479.

29. Ikeda K, Nakamura Y, Humble TS. Application of quantum annealing to nurse scheduling problem. *Scientific Rep* (2019) 9:12837. doi:10.1038/s41598-019-49172-3

30. Sadhu A, Zaman S, Das K, Banerjee A, & Khan F Quantum annealing for solving a nurse-physician scheduling problem in Covid-19 clinics (2020).

31. Stollenwerk T, Michaud V, Lobe E, Picard M, Basermann A, Botter T, Image acquisition planning for Earth observation satellites with a quantum annealer (2020). 2006.09724.

32. Domino K, Koniorczyk M, Krawiec K, Jałowiecki K, & Gardas B Quantum computing approach to railway dispatching and conflict management optimization on single-track railway lines (2021). 2010.08227.

33. Domino K, Koniorczyk M, Krawiec K, Jałowiecki K, Deffner S, Gardas B, Quantum annealing in the nisq era: Railway conflict management (2021). 2112.03674.

34. Ebadi S, Keesling A, Cain M, Wang TT, Levine H, Bluvstein D, et al. Quantum optimization of maximum independent set using rydberg atom arrays. *Science* (2022) 376:1209–15. doi:10.1126/science.abo6587

35. Tiunov ES, Ulanov AE, Lvovsky AI. Annealing by simulating the coherent ising machine. *Opt Express* (2019) 27:10288–95. doi:10.1364/oe.27.010288

36. Killoran N, Bromley TR, Arrazola JM, Schuld M, Quesada N, Lloyd S. Continuous-variable quantum neural networks. *Phys Rev Res* (2019) 1:033063. doi:10.1103/PhysRevResearch.1.033063

37. Oshiyama H, Ohzeki M. Benchmark of quantum-inspired heuristic solvers for quadratic unconstrained binary optimization. *Scientific Rep* (2022) 12:2146. doi:10.1038/s41598-022-06070-5

38. Resende MGC. Combinatorial optimization in telecommunications. In: *Applied optimization*. New York: Springer US (2003). p. 59–112. doi:10.1007/978-1-4613-0233-9_4

39. Vesselinova N, Steinert R, Perez-Ramirez DF, Boman M. Learning combinatorial optimization on graphs: A survey with applications to networking. *IEEE Access* (2020) 8:120388–416. doi:10.1109/ACCESS.2020.3004964

40. Martin V, Brito JP, Escribano C, Menchetti M, White C, Lord A, et al. Quantum technologies in the telecommunications industry. *EPJ Quan Tech* (2021) 8:19. doi:10.1140/epjqt/s40507-021-00108-9

41. Harwood S, Gambella C, Trenev D, Simonetto A, Bernal Neira D, Greenberg D. Formulating and solving routing problems on quantum computers. *IEEE Trans Quan Eng* (2021) 2:1–17. doi:10.1109/tqe.2021.3049230

42. Marx D. Graph colouring problems and their applications in scheduling. *Periodica Polytechnica Electr Eng (Archives)* (2004) 48:11–6.

43. Chaitin GJ, Auslander MA, Chandra AK, Cocke J, Hopkins ME, Markstein PW. Register allocation via coloring. *Computer languages* (1981) 6:47–57. doi:10.1016/0096-0551(81)90048-5

44. Ott J, Tan D, Loveless T, Grover WH, Brisk P. Chemstor: Using formal methods to guarantee safe storage and disposal of chemicals. *J Chem Inf Model* (2020) 60:3416–22. doi:10.1021/acs.jcim.9b00951

45. Garey M, Johnson D, So H. An application of graph coloring to printed circuit testing. *IEEE Trans circuits Syst* (1976) 23:591–9. doi:10.1109/tcs.1976.1084138

46. Glockner G. Parallel and distributed optimization with gurobi optimizer (2015). Available at: https://assets.gurobi.com/pdfs/2015-09-21-Parallel-and-Distributed-Optimization-with-the-Gurobi-Optimizer.pdf.

47. Batagelj V, Brandes U. Efficient generation of large random networks. *Phys Rev E* (2005) 71:036113. doi:10.1103/PhysRevE.71.036113

Appendix

TABLE A1 Comparison of the proposed QUBO transformation for the graph coloring problem to original QUBO transformation described in [7].

Number of nodes	Original QUBO transformation		Proposed QUBO transformation	
	Number of colors	Run time	Number of colors	Run time
10	4.34	.28	4.34	.19
20	6.47	.62	6.36	.45
30	8.24	7.67	8.02	4.95
40	10.31	14.22	9.39	8.90
50	12.41	26.28	10.96	16.82
60	14.53	42.01	12.44	28.51
70	16.52	63.89	14.01	61.58
80	18.03	98.50	15.56	69.00
90	19.74	106.61	17.02	79.87
100	20.65	140.41	18.54	123.13
Average result				
(Lower is better)				

Experiments were performed on the same dataset of 900 randomly generated graphs with the use of SimCIM. Results show that the proposed QUBO runs faster, giving on average lower or the same number of colors.