Comparative Analysis of Algorithms for Finding the Maximum Independent set of Graphs on Quantum and Traditional Computer

Vyacheslav Korolyov and Oleksandr Khodzinskyi

V.M. Glushkov Institute of Cybernetics of Ukrainian National Academy of Sciences, 801 box, 1 build., 40 Glushkov prosp., Kyiv, 03680, Ukraine

Abstract

Quantum computers allow to obtain several times faster solution of some NP-complex problems of combinatorial optimization problems in comparison with classical clusters based on electronic computers, but inferior in the dimension of problems due to the small number of qubits and modern technological capabilities. The basis for solving some combinatorial optimization problems on quantum computers is the model of quadratic binary optimization with constraints. Creating algorithms for solving combinatorial optimization problems for hybrid quantum-classical computing systems allows to speed up the results and get more accurate solutions without restrictions on the dimensionality of problems. Examples of solving the problem of finding the maximum independent set of a graph on quantum computers from IBM and D-Wave are given. Universal and special approaches to calculations on quantum computers are considered. An example of solving the problem of finding the largest independent set on a quantum-classical cloud service from D-Wave is given The criterion for the correctness of solving problems of combinatorial optimization problems is usually taken as the minimum energy of the system, which, according to the authors, is sufficient, but not necessary condition for the existence of an optimal solution of the problem.

Keywords [1]

Quantum computer, quantum computer mathematics, qubit, maximum independent set of a graph

1. Introduction

Modern quantum computers (QCs) have moved from the stage of laboratory samples to computing services provided through cloud services on the Internet and are already able to solve relatively small-scale problems. For example, QCs allow you to solve combinatorial optimization problems for graphs with up to several dozen vertices. Performing the decomposition of the optimization problem on electronic computers with the subsequent transfer of subtasks to QC allows you to increase the total number of vertices of the input graph to several hundred. Therefore, the main attention of the developers of the QC is focused on the problems of increasing the number of qubits, controlling their states, increasing the number of connections between qubits, calibration of quantum computing systems, correction of computational errors.

According to Google experts and scientists specializing in quantum physics [1], to perform universal calculations on the QC must wait for the creation of systems consisting of more than a million quantum bits (qubits), because at this stage of technology, unstable physical qubits must be combined in logical qubits to correct computational errors caused by thermal noise and the limited

II International Scientific Symposium «Intelligent Solutions» IntSol-2021, September 28–30, 2021, Kyiv-Uzhhorod, Ukraine
EMAIL: korolev.academ@gmail.com (A. 1); okhodz@gmail.com (A. 2)
ORCID: 0000-0003-1143-5846 (A. 1);

lifetime of controlled quantum states of the created computing system (Decoherence Time). But even now, hybrid computers with quantum coprocessors can solve special optimization problems and model physical systems or chemical compounds at a rate that exceeds or compares traditional approaches. Today, all QC cloud services are built on superconducting technologies and operate at temperatures close to absolute zero, and photonics, which can operate at room temperature, QC developers promise to make available online in the next 2-3 years. At the time of writing, there are already several quantum computers [2]. IBM has several quantum computers with 1-65 qubits that are available for public use and paid access through cloud services. Google has a quantum processor with 72 qubits, called "Bristlecone", Intel has a quantum computer "Horse Ridge 2" with 128 qubits. The new companies: D-Wave (5436 qubits) and Rigetti (28 qubits), IonQ (32 qubits) provide paid access and time-limited free access to quantum computing. The most productive today is the Chinese photonic QC Jiuzhang, which contains 72 qubits. The Canadian startup QC Xanadu (24 qubits) also works based on photons.

2. General scheme of operations for quantum computing

Quantum superposition is the coexistence of qubit states that cannot be realized simultaneously from the classical point of view. Execution of quantum calculations is based on a sequence of four operations [3, 4, 5]:

1) Formulation of the optimization problem according to the quantum model of calculations;
2) Creation of quantum superposition;
3) Performance of calculations in superposition and transformations of qubits;
4) Reduction of the noise of measurement and reading of data.

In all quantum algorithms there are the following steps [5-15]:

1. Preparation of superposition of input data for calculation of function;
2. Application of the function itself (algorithm or quantum oracle);
3. Transformation of the obtained states so that the probability of the desired result for us was close to one (calibration of equipment, noise reduction, and multiple runs of the problem-solving process).

There is no single method for comparing quantum computers of different structures and physical principles of operation. The number of qubits may seem the most important criterion, but the number of connections between them is also important (Figure 1) because qubits do not connect to a common data bus, as in electronic computers (computers), the number of supported quantum logic elements, queue depth tasks, operating frequency, etc. One of the main difficulties in building quantum computers is to preserve quantum states for as long as possible, which is often achieved by low temperatures and control of computation time. The slightest disturbance of the physical state of the system can destroy quantum states, and then quantum calculations will have errors that will need to be corrected.

The connections between qubits in terms of computer mathematics mean the ability to transfer data between them or apply quantum valves to their states to perform operations.

3. Solving combinatorial optimization problems using quantum computers

The amount of computation required to solve NP-complex optimization problems [3, 4, 16, 17, 18] increases rapidly with an increasing dimension of the problem. Such problems cannot be solved by direct search, even with unlimited computing resources. Approximate algorithms, in particular, metaheuristics, are used to solve NP-complex optimization problems [17, 18]. The use of quantum calculations allows to accelerate the solution of NP-complex optimization problems and to obtain solutions of increased accuracy.

3.1. Solving the problem of a maximum independent set

Formulation of the problem. Given a simple undirected graph $G(V, E)$, where V is the set of vertices, E is the set of edges, n is the number of vertices, m is the number of edges, $V = \{1,2, ..., n\}$, $E = \{(u_1, v_1), (u_2, v_2), ..., (u_m, v_m)\}$; $u_i, v_i \in V$; $i = 1, ..., m$. Each vertex i is associated with a variable x_i,

which can take the value 0 or 1: $x = (x_1, ..., x_n)$, $x_i \in \{0,1\}$. The function $f(x) = \sum_{i=1}^{n} x_i$ and the constraint on the value of the vector $x = (x_1, ..., x_n)$: $(i, j) \in E \Rightarrow x_i + x_j \leq 1$ are also given.

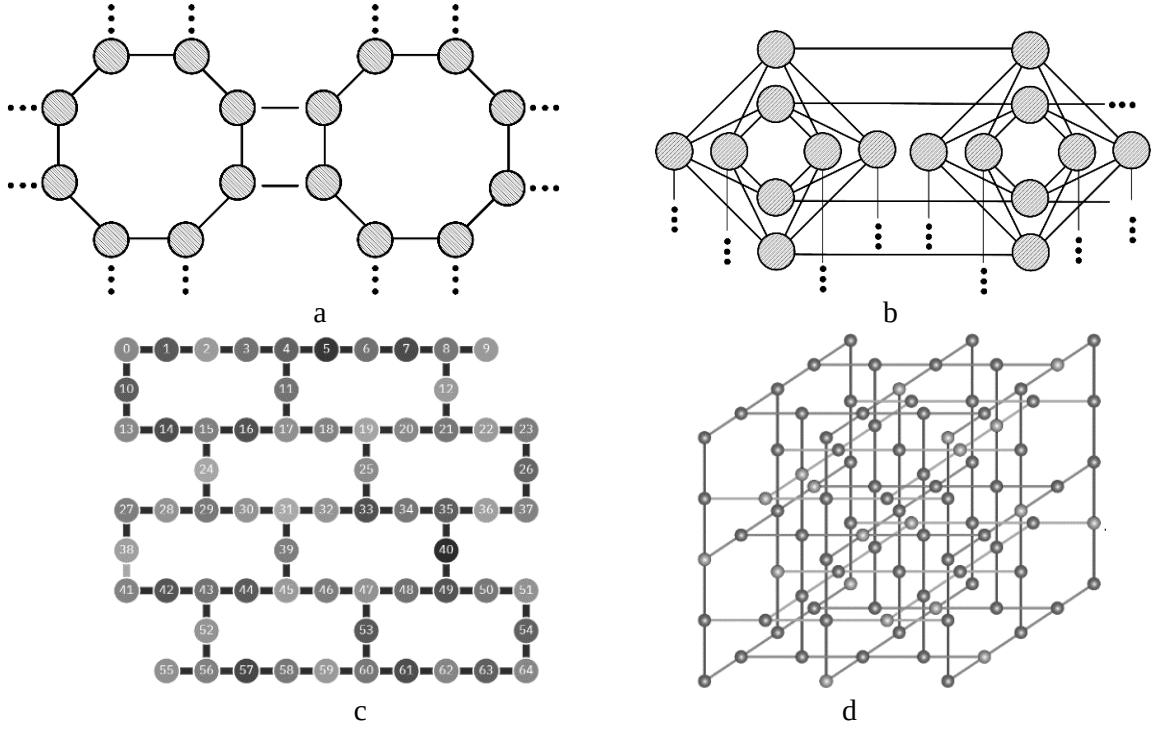


Figure 1. Qubit connection structures for quantum processors from Rigetti (a), D-wave (b), IBM (c), Xanadu (d)

Thus, the problem of the maximum independent set of a graph (MISG) is formulated as follows:

find $max\ f(x) = \sum_{i=1}^{n} x_i$

for constraints $(i, j) \in E \Rightarrow x_i + x_j \leq 1\ x_i \in \{0,1\}$, $i = 1, ..., n$.

Examples of independent sets for graphs are shown in Figure 2.

This problem, like most others in combinatorial optimization, is NP-complex [5, 18]. In such cases, optimization problems quickly become unsolvable methods of direct search, even with large computing resources. It will take several centuries to solve this class of large-scale problems as computers grow in line with Moore's laws. Therefore, heuristic approaches and empirical methods are usually applied to the problems of computationally complex optimization problems. The development of heuristic algorithms can take a long time to obtain satisfactory answers, in addition, they do not guarantee an optimal solution.

Under ideal conditions, quantum computers make it possible to obtain the exact optimal solution of problems, but due to physical and technological limitations, in practice, they choose an acceptable solution with minimal energy. Visual verification of problem-solving procedures on graphs with a small number of vertices is only partially suitable because quantum physics allows variation of the initial results with the same input parameters. The study aims to determine the requirements for the construction of criteria for evaluating the quality of solutions of combinatorial optimization problems for the stages of decomposition and final solutions of the problem, which can be used to build new meaningful problem statements, improve computational procedures and automate their quality.

Many optimization problems can be solved on universal or specialized QCs. To solve the problem of finding the largest independent set of graphs on the QC, as will be shown below, the model of quadratic unconstrained binary optimization (QUBO) is well suited [19, 20], so other possible formulations of this optimization problem for calculation on the QC go beyond this consideration.

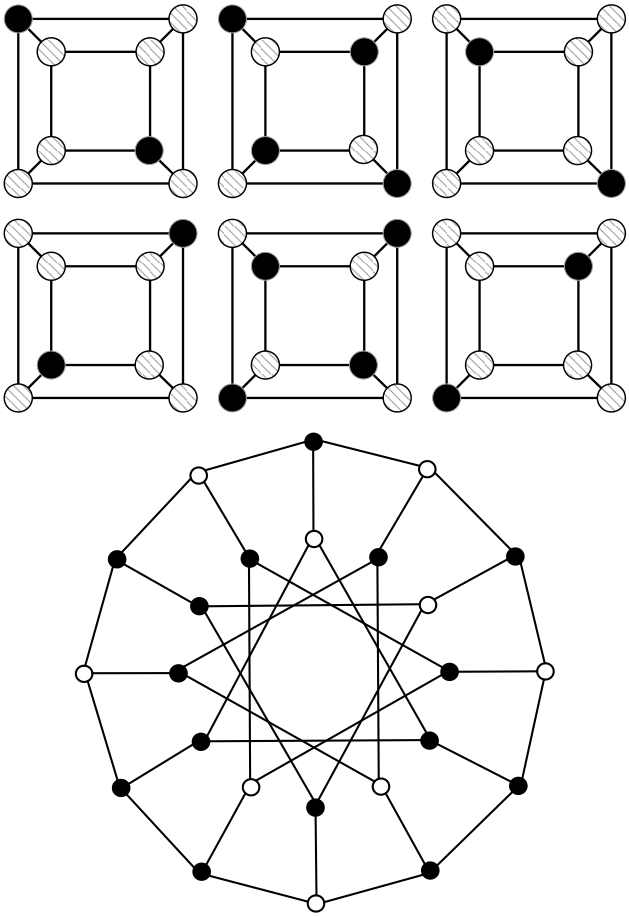


Figure 2. Examples of independent sets of a graph. Black circles correspond to independent set.

Many combinatorial optimization problems on graphs use the QUBO problem, which is determined using the upper diagonal matrix Q of dimension NxN, the upper diagonal matrix of real weights, x - vector, binary variables, as minimizes the function:

$$f(x) = \sum_i Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j$$

where the diagonals are linear coefficients Q_i, i, and the elements Q_i, j are quadratic coefficients, or

$$\min_x f(x) = \min_{x \in \{0,1\}^n} x^T Q x.$$

Since the problem of finding an independent set is a problem with constraints, the problem is reduced to a general form of the form [18]:

$$\min \textit{(objective function)} + \gamma \textit{ [constraint]}.$$

The value of γ characterizes the degree of rigidity of the constraints, is technically realized by adjusting the parameters of quantum computers, and is checked by repeating the calculations. For this problem of finding the largest independent set of graphs on the QC of D-wave, it was experimentally established [20] that the Lagrange parameter γ should be set in the range from 75% to 150% of the predicted value of the objective function depending on the need for softer or harder constraints.

To formulate a problem for a quantum computer, it is necessary to construct an objective function that is a reflection of the physical state of the system as a function of binary variables representing qubits. In quantum computing, the vertices of a graph are represented by qubits, and the edges are represented by connections between qubits. In most cases, the lower the energy state of a quantum system that reflects the objective function, the closer the solution of the problem is to the optimal one. The resulting code of the algorithm for solving the optimization problem in the

formulation of QUBO software services of the developer, such as IBM or D-Wave, converts a sequence of instructions according to the Ising model, which can be transferred to the QC.

Quantum annealing is a software and hardware computational procedure designed to solve NP-complex combinatorial optimization problems. Modern quantum annealers solve the problems of quadratic unlimited binary optimization - QUBO, also known as the Ising spin model:

$$E_{\text{Ising}}(s) = \sum_{i=1}^{N} h_i s_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{i,j} s_i s_j,$$

where E_{Ising} is the value of the Ising energy for the quantum system, s is the spin number that takes the value "+1" or "-1", hi - linear coefficients that are responsible for the displacement of qubits; Ji, j are the quadratic coefficients that are responsible for the strength of the connection between a pair of qubits.

3.2. Substantive statement of the problem of binary combinatorial optimization to find the maximum independent set of vertices of the graph

To present the problem of finding the largest independent set as a problem of unlimited binary optimization, we define the set of vertices of the graph in terms of binary variables. Assign to each vertex of the graph (Figure 3) a binary variable x_i:

$$x_i = \begin{cases} 0, \text{if the vertex is not in the set} \\ 1, \text{if the vertex is in the set} \end{cases}$$

Then for the set of vertices of the graph (Fig. 4) S = {2, 3, 6} we have:

$$x_1 = 0;\ x_2 = 1;\ x_3 = 1;\ x_4 = 0;\ x_5 = 0;\ x_6 = 1;\ x_7 = 0.$$

Since you need to find the largest independent set consisting of variables corresponding to the vertices of the graph, so the objective function will be the sum of the maximum value:

$$Obj = \max \sum_{i=1}^{N} x_i$$

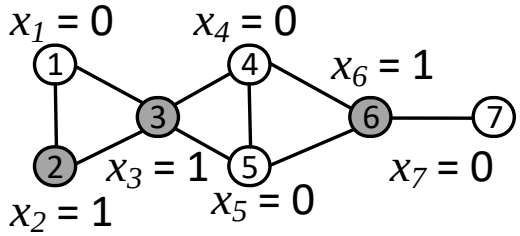


Figure 3. Denoting the set of vertices of a graph by binary variables

The maximum independent set of vertices of the graph means that there are no edges between them, i.e. it is necessary to describe the constraints on the connection of vertices. This can be represented by multiplying the binary variables corresponding to the vertices of the graph (Figure 4) between the corresponding edges [19].

If you add up all the products for the edges, you can get the number of "bad" edges. Then the constraints for the edges can be written as follows:

$$\sum_{(u,v)\in E} x_u \cdot x_v = 0$$

where *u*, *v* are the vertices of graph *E*, because the purpose of the optimization problem is the absence in the largest independent set of vertices with common edges.

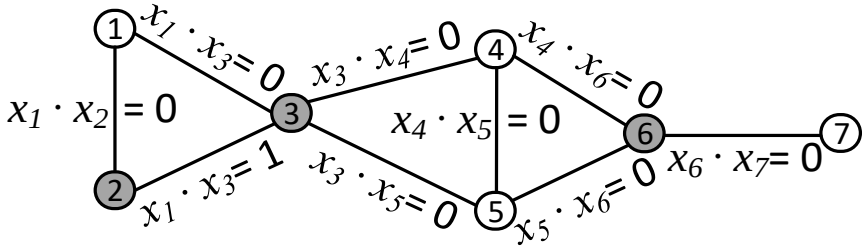


Figure 4. Illustration of the representation of constraints on vertex connections

The method of quantum annealing requires a minimum of energy, but the optimization problem under consideration is a maximization problem, so you need to invert the sign of the objective function. The expression for constraints on vertex connections remains unchanged because the zero value of the sum of the products of binary variables that correspond to the condition that there are no connections between the vertices of the graph satisfies the minimum of the function; a value greater than zero that corresponds to a connection will increase the value of the function. As a result, we have QUBO:

$$\min\left[\left(-1\cdot\sum_{i=1}^{N}x_i\right)+\gamma\left(\sum_{(u,v)\in E}x_u\cdot x_v\right)\right]$$

Consider solving the problem of the largest independent set on a universal quantum computer from IBM [2, 3, 6, 7, 20] and a specialized quantum computing device from D-wave [2, 3, 19, 21, 22] for solving minimization problems by the method of quantum annealing [20, 21, 22]. Quantum computer manufacturers have developed software libraries for the Python programming language [6-15], which provide a solution to the problem of finding the largest independent set for a graph. To get a solution to the problem, you need to provide a description of the graph at the input of the program and select a quantum computer from the available online, and set the calculation parameters, such as the number of runs of the problem. The results of the experiments are presented in the table. 1. A graph of four vertices is used to compare the results of solutions on the D-wave QC and IBM, which at the time of computational experiments provided free access to the QC of five qubits.

Table 1 shows the values of the objective function, the options for solving the problem - independent vertices, and the energy values of the system for the corresponding options.

Consider solving the problem of finding the largest independent set on the QC of the firm D-wave for a graph (Figure 5,6) consisting of seven vertices. It is clear that for a graph there are several options for solving the problem of finding the maximum independent set.

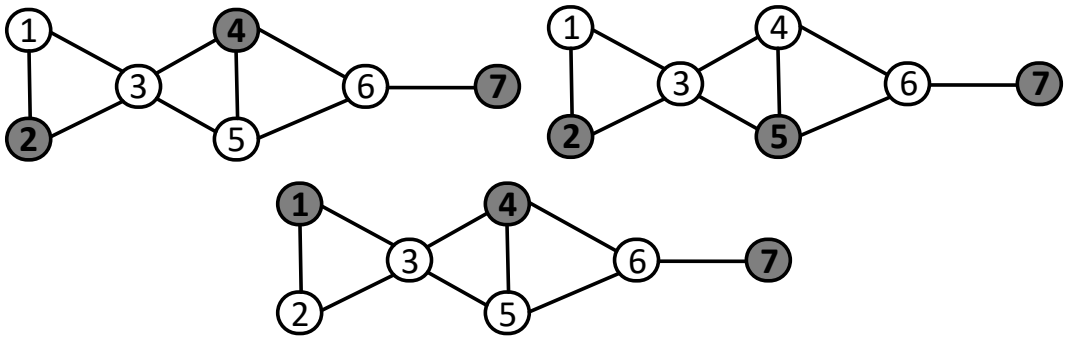


Figure 5. Several solutions to the problem of finding MISG

Figure 7 shows the display of the MISG problem in the form of a binary logic problem with seven variables for the QC. Each circle is a variable, and each line is the product of two variables. Both individual qubits and qubit chains can correspond to variables.

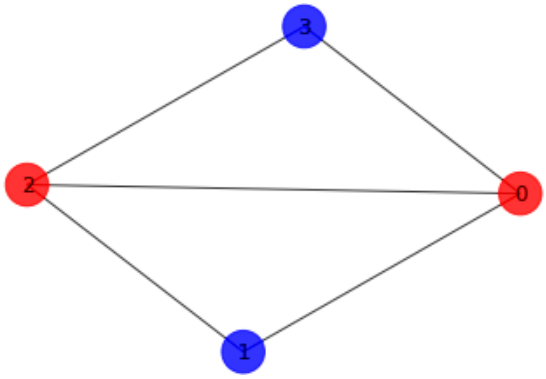
Figure 9 shows the result of embedding the MISG problem in the QC processor. Embedding is the process of displaying the QUBO problem in the form of qubits and chains of connections connecting qubits. Circles denote qubits and lines denote connections.

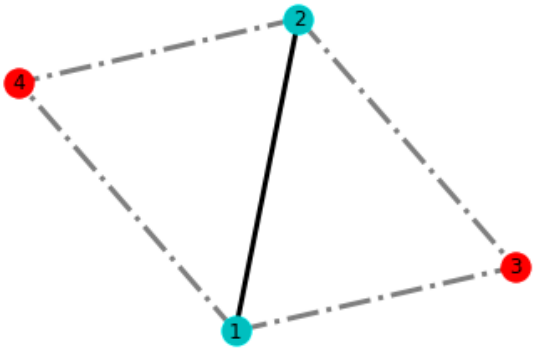
The histogram of the frequency of occurrence of solutions (Solution Occurrences) is ordered by the value of energy (Energy / Source) (Figure 9). Lower energy values correspond to better solutions.

Table 1.

The results of solving the problem of the largest independent set on quantum computers

IBM	D-wave			
energy: -1.5	Set 0	Set 1	Energy	Cut Size
max-cut objective: -4.0	[3, 4]	[1, 2]	-3.0	4
solution: [0. 1. 0. 1.]	[1, 2]	[3, 4]	-3.0	4
solution objective: 4.0	[1, 3]	[2, 4]	-1.0	3
	[1, 4]	[2, 3]	-1.0	3
	[2, 3]	[1, 4]	-1.0	3





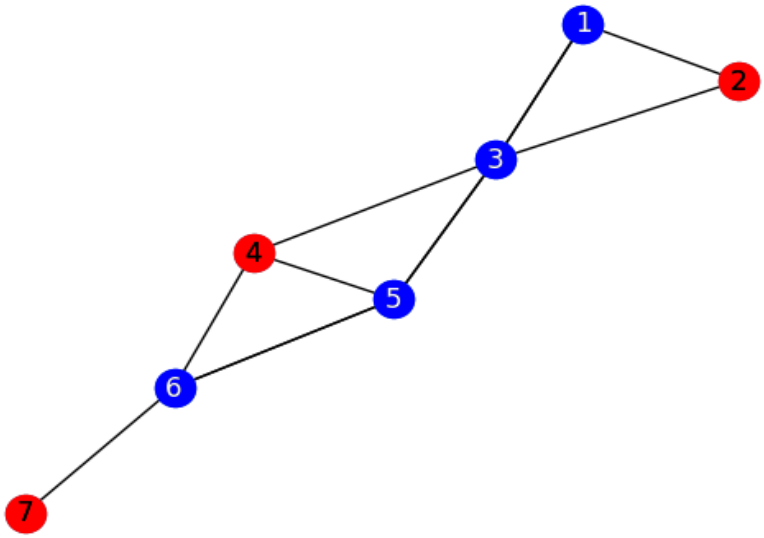


Figure 6. Search result of MISG {2,4,7} on D-wave QC

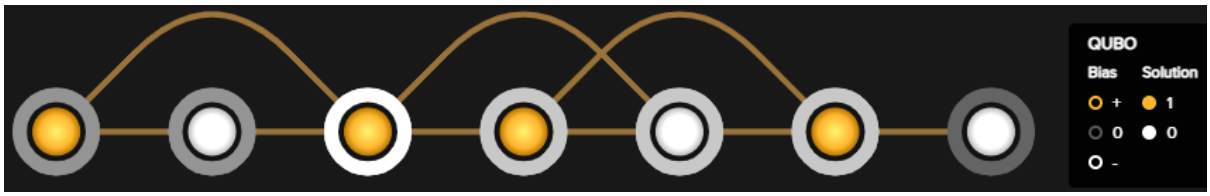


Figure 7. Logical problem performed on the QC and mapping a graph of seven vertices into the QUBO model

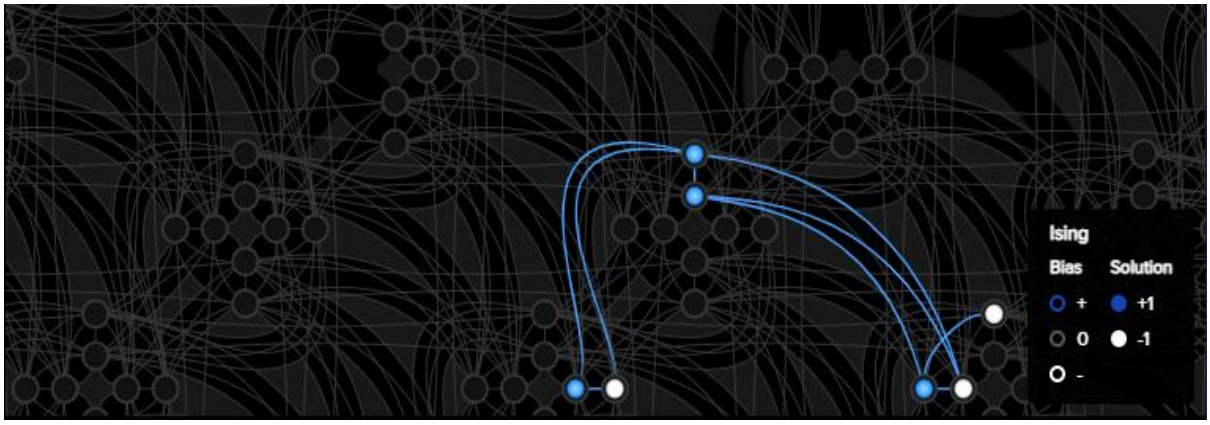


Figure 8. Display of a graph of seven vertices on the QC D-wave

3.3. Solving the problem of searching for medium-scale MISG

To decompose combinatorial optimization problems on graphs with more than 40 vertices, D-Wave, a manufacturer of quantum computers, offers [21, 22]:

- algorithms for decomposition of problems;
- ways to combine quantum and classical computers;
- procedures for managing workflow parameters to obtain calculation results.

Decomposition is performed by selecting a subtask with variables that most affect the energy of the problem and can fit in the QC. The variables selected based on the greatest energy impact may be unrelated (there are no common edges in the subgraphs) and do not describe the local structure of the problem. D-Wave software [21, 22] allows you to configure the mode of traversing the graph in width (breadth-first selection - BFS) or priority (priority-first selection - PFS), can detect the features of patterns representing local structures within the problem. In BFS mode, the graph traversal begins with the vertex that has the greatest energy effect, from which the graph traversal continues to the directly connected vertices, then to the vertices directly related to them, and so on, with the graph traversal ordered by the vertex index. In PFS mode, the graph bypass selects the vertex with the strongest energy impact among the unselected vertices directly connected to any already selected vertex. For the problem of finding the largest independent set of a graph, a decomposition algorithm is constructed, which traverses the graph in width starting from the variable with the largest energy impact and selecting approximately 50 variables for each subtask.

Here are some visual examples of solving the problem of finding the largest independent set for a graph of 300 vertices (Figure 11-13). It is visually difficult to assess the quality of both the solution of the problem and the subtasks created by the decomposition services.

3.4. Comparison of the solution time of the problem on a hybrid (quantum-classical cloud) and classical computers

D-Wave's modern quantum clouds allow you to solve quadratic binary optimization (QUBO) problems for graphs containing up to 1 million vertices and up to 2 million weighted edges. D-Wave's technical report states [21] that hybrid computers outperform classic Amazon cloud servers in solving tens of thousands to a million vertices of large QUBO problems. The solution time for D-Wave hybrid computers was up to 3 seconds versus 20 minutes for Amazon servers, computation time may vary due to the number of vertices and edges, and the degree of the sparseness of the graph.

For graphs of medium size with the number of vertices from 100 to 3200, the results of numerical experiments are summarized in Table 2 [22], which compares the solution time of the problem of finding the maximum independent set of a graph by the method of quantum annealing and simulated annealing on a high-performance workstation.

Similar results were obtained by the authors of the article to solve the problem of finding the maximum independent set of a graph on a hybrid computer by quantum annealing and a personal computer of average performance for a Python program looking for solutions by branches and boundaries. The D-wave hybrid computer solved the problem ten times faster, and for graphs with

more than 450 vertices, the solution time on a classic computer was already more than an hour, so no further research was performed.

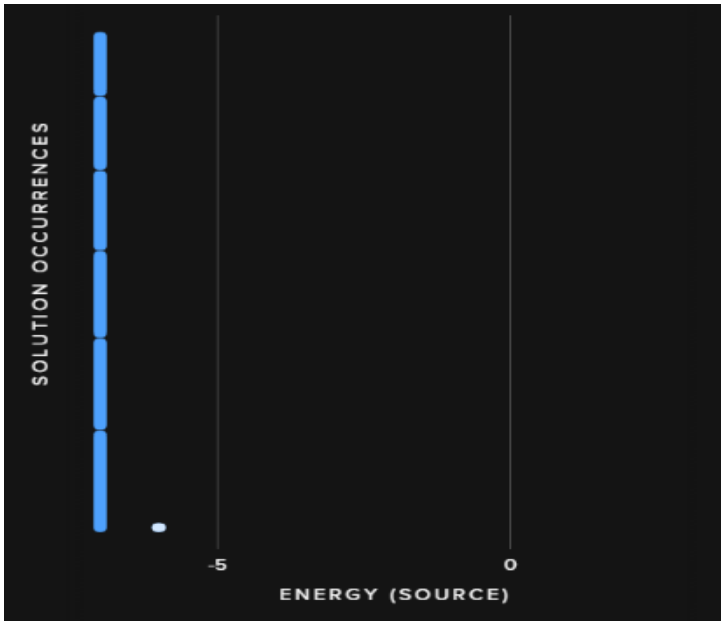


Figure 9. Histogram of probabilities for solving the PNNMG problem for 10,000 starts, below the energy value "7". You can also see the erroneous unlikely solution of the problem with the energy value "-6".

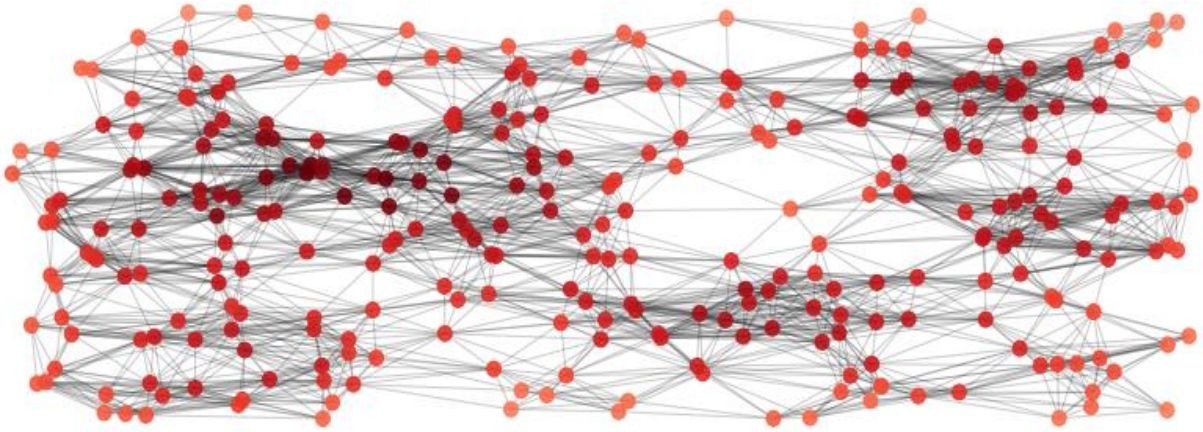


Figure 10. Example of a test random graph with 300 vertices

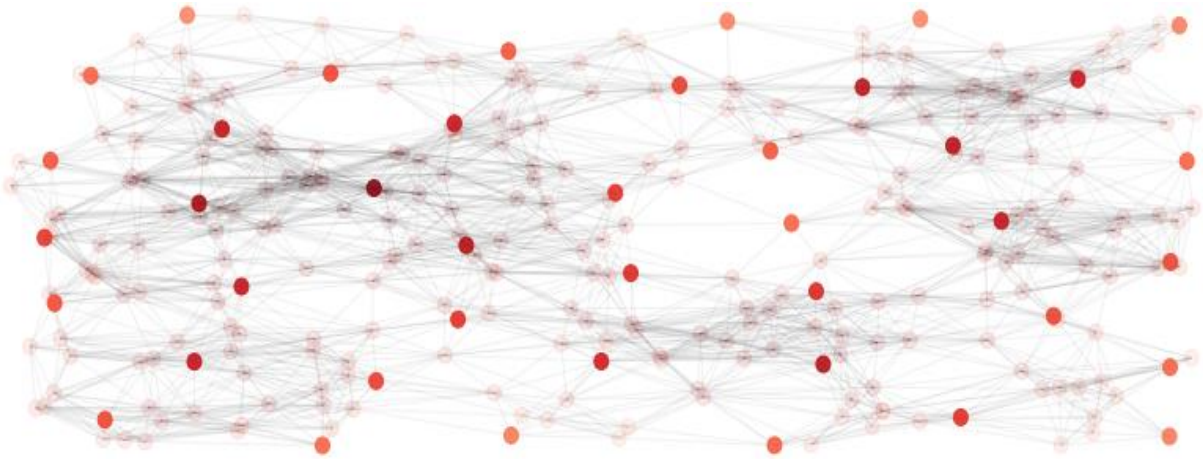


Figure 11. Solving the problem of finding the MISG for a test graph of 300 vertices

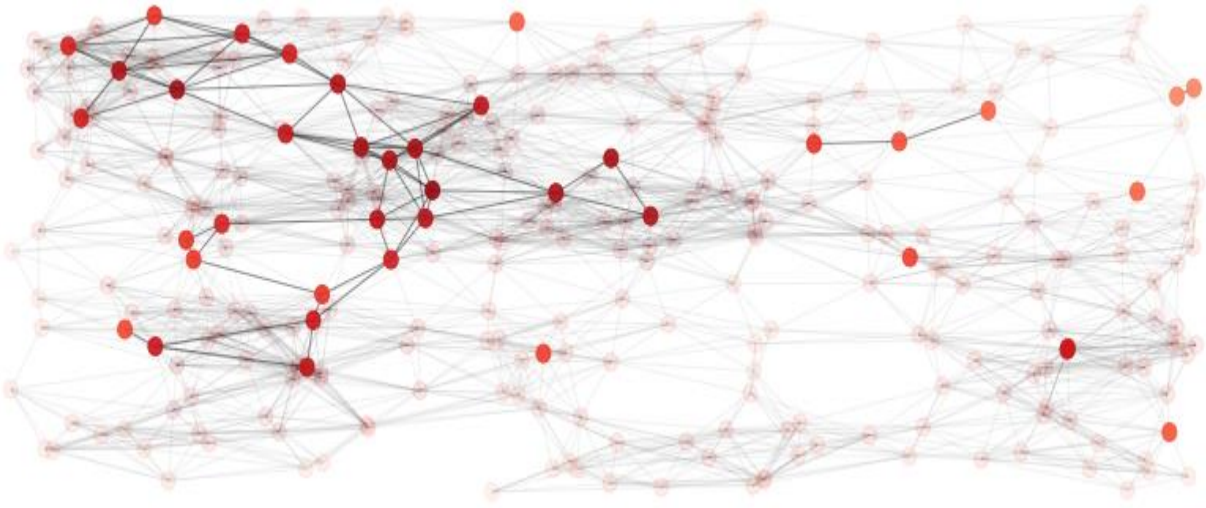


Figure 12. Example of decomposition on subtasks for graphs with 40 vertices for the problem of finding the MISG (another input graph) according to the heuristic algorithm

Table 2.
Comparison of the solution time of the problem of finding the maximum independent set on a quantum-classical cloud and a laptop by the annealing method

The number of vertices of the graph	Allocated time for a solution, sec	Total computing time of the hybrid computer, sec	Quantum computer (QC) operating time, sec	Calculation time on a classic workstation, sec
100	3	2,9997	0,37	0,536
200	3	2,999	0,29	0,78
400	3	2,998	0,16	1,86
800	3	2,99	0,20	6,77
1600	5	4,998	0,20	38,2
3200	8	7,98	0,12	247,0

4. Resume

Modern quantum-classical (hybrid) computing servers allow you to solve combinatorial optimization problems of medium-scale with the speed and dimension required by commercial structures. It is proposed to consider two approaches to the calculation of combinatorial optimization problems on quantum computers: universal, using quantum valves, and specialized, based on the parameterization of physical processes. An example of solving the problem of finding the maximum independent set for quantum computers from IBM and D-wave and a quantum-classical cloud server is given. The results of numerical experiments on solving combinatorial optimization problems show the need to develop criteria for decomposition of problems and the quality of solving both subtasks and problems in general.

It took about 30 seconds to calculate the solution of the problem of finding the maximum independent set on a personal computer, and 3 seconds for a hybrid computer with a quantum coprocessor for a random graph of 450 vertices and about 17,000 edges. The algorithm for solving on a personal computer is a method of branches and boundaries, which is programmed in Python; Hybrid computer solution procedure: heuristic decomposition of the problem into subgraphs for which the quantum annealing problem is solved.

5. References

[1] Lucero E. Unveiling our new Quantum AI campus. URL: https://blog.google/technology/ai/unveiling-our-new-quantum-ai-campus/

[2] List of quantum processors. URL: https://en.wikipedia.org/wiki/List_of_quantum_processors

[3] Korolyov V.Yu., Khodzinskyi O.M. “Rozv″yazuvannya zadach kombinatornoyi optymizatsiyi na kvantovykh komp'yuterakh” [Solving combinatorial optimization problems on quantum computers]. Cybernetics and Computer Technologies. 2020. 2. C. 5–13. (in Ukrainian) https://doi.org/10.34229/2707-451X.20.2.1

[4] Korolyov V.Yu., Ogurtsov M.I., Khodzinskyi O.M. “Bahatorivneve derzhavne vpiznavannya ob″yektiv ta analiz zastosovnosti post-kvantovykh kryptohrafichnykh alhorytmiv dlya zakhystu informatsiyi” [Multilevel state object recognition and analysis of the applicability of post-quantum cryptographic algorithms for information security.] Cybernetics and Computer Technologies. 2020. 3. C. 74–84. (in Ukrainian) https://doi.org/10.34229/2707-451X.20.3.7

[5] Nielsen M.A., Chuang I.L. Quantum Computation and Quantum Information. Cambridge: University Press, 2012. https://doi.org/10.1017/CBO9780511976667

[6] Moran C.C. Mastering Quantum Computing with IBM QX. Birmingham: Packt, 2019.

[7] Norlen H. Quantum Computing in Practice with Qiskit® and IBM Quantum Experience. Birmingham: Packt, 2020.

[8] Vos J. Quantum Computing for Developers: A Java-based introduction. New York: Manning, 2020.

[9] Johnston E.R., Harrigan N., Gimeno-Segovia M. Programming Quantum Computers. Essential Algorithms and Code Samples. Sebastopol: O’Reilly, 2019.

[10] Kommadi B. Quantum Computing Solutions. Solving Real-World Problems Using Quantum Computing and Algorithms. New York: Apress, 2020. https://doi.org/10.1007/978-1-4842-6516-1

[11] Kaiser S.C., Granade C.E. Learn Quantum Computing with Python and Q#. A hands-on approach. New York: Manning, 2021.

[12] Easttom C. Quantum Computing Fundamentals. Boston: Addison-Wesley, 2021.

[13] Senekane M. Hands-On Quantum Information Processing with Python: Get Up and Running with Information Processing and Computing Based on Quantum Mechanics Using Python. Birmingham: Packt, 2021.

[14] Loredo R. Learn Quantum Computing with Python and IBM Quantum Experience: A hands-on introduction to quantum computing and writing your own quantum programs with Python. Birmingham: Packt, 2020.

[15] Ganguly S., Cambier T. Quantum Computing with Silq Programming. Birmingham: Packt, 2021.

[16] Gary M., Johnson D. Computing machines and difficult-to-solve problems; Mir: Moscow, 1982. (in Russian)

[17] Talbi E.-G. Metaheuristics: from design to implementation; Wiley: Hoboken, 2009.

[18] Hulyanitsky L.F., Mulesa O.Yu. “Prykladni metody kombinatornoyi optymizatsiyi” [Applied methods of combinatorial optimization]; Publishing center "Kyiv University", 2016. (in Ukrainian)

[19] Maximum Cut. URL: https://github.com/dwave-examples/maximum-cut (accessed 06/21/2021).

[20] Asfaw A. Learn Quantum Computation using Qiskit. URL: https://qiskit.org/textbook/ch-applications/qaoa.html (accessed 06/21/2021).

[21] D-Wave Hybrid Solver Service: An Overview. URL: https://www.dwavesys.com/sites/default/files/14-1039A-A_D-Wave_Hybrid_Solver_Service_An_Overview.pdf (accessed 06/21/2021).

[22] Using the D-Wave Quantum Hybrid Solver Service | Webinar. URL: https://www.youtube.com/watch?v=1dnUhGUO3zQ (accessed 06/21/2021).