Foundations for Workflow Application Scheduling on D-Wave System

Dawid Tomasiewicz[1], Maciej Pawlik[1], Maciej Malawski[1], and Katarzyna Rycerz[1]

Institute of Computer Science, AGH, al. Mickiewicza 30, 30-059 Kraków, Poland
tomasiewicz.dawid@gmail.com, {mapawlik,malawski,kzajac}@agh.edu.pl

Abstract. Many scientific processes and applications can be represented in the standardized form of workflows. One of the key challenges related to managing and executing workflows is scheduling. As an NP-hard problem with exponential complexity it imposes limitations on the size of practically solvable problems. In this paper, we present a solution to the challenge of scheduling workflow applications with the help of the D-Wave quantum annealer. To the best of our knowledge, there is no other work directly addressing workflow scheduling using quantum computing. Our solution includes transformation into a Quadratic Unconstrained Binary Optimization (QUBO) problem and discussion of experimental results, as well as possible applications of the solution. For our experiments we choose four problem instances small enough to fit into the annealer's architecture. For two of our instances the quantum annealer finds the global optimum for scheduling. We thus show that it is possible to solve such problems with the help of the D-Wave machine and discuss the limitations of this approach.

Keywords: D-Wave, QUBO, workflow scheduling, serverless

1 Introduction

The paradigm of workflows is commonly used for describing and preserving complex scientific processes and applications [13]. Workflows are usually represented as Directed Acyclic Graphs (DAG) [23]. Each vertex represents a task, while edges designate dependencies or data transfers between tasks. By using such a general representation it is possible to improve application portability and reusability. The abstract graph representation enables decoupling the application from a specific infrastructure and easily extract parts of the process with clear understanding of what the extracted part does and what its dependencies and outputs are. Some examples of scientific applications implemented as workflows include: Montage [4] – image mosaic software used to construct human-perceptible images of sky features from multiple images captured by telescopes; software used by the LIGO collaboration, designed to process data related to detecting gravitational waves [1]; software designed to predict the occurrence and effects of earthquakes based on geological data [23].

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

One of the key challenges related to managing and executing scientific workflows is scheduling. The general goal of scheduling is to create a plan of execution with respect to given parameters such as deadline, budget and computing resources. For the scope of this paper we assumed a simple form of workflow scheduling, where we operate on the serverless infrastructure discussed in [3]. In this case the serverless infrastructure is implemented as a set of cloud functions, provided by major cloud service providers, such as Google, Amazon, Azure and IBM. Cloud functions have the potential to be used for compute-intensive tasks, as proposed in [27], with particular emphasis on challenges which require high levels of infrastructure elasticity. Scheduling workflows for serverless infrastructures can be summarized as defining the runtime parameters of a function based on user-supplied deadline and budget. In this paper we assume that the serverless infrastructure consists of an infinite number of machine instances, while the number of machine types is finite, with each machine type associated with specific cost and performance.

In recent years, quantum computing has become popular due to its potential ability to solve problems that are beyond the capabilities of classical computing infrastructures. The research is still in its early stage, however there are many theoretical quantum algorithms already available, such as famous polynomial time Shor factorization [26] or $O(\sqrt{N})$ complexity Grover search in unsorted databases [16]. A good overview of the current status of theoretical quantum algorithms can be found in [18]. There are also various attempts to implement algorithms on the available quantum hardware: IBM-Q[1], Rigetti computing[2] or D-Wave[3]. Scheduling problems are assumed to be one of the challenges which might be efficiently solved by this new approach.

In this paper, we present a solution for the workflow scheduling problem with the use of the D-Wave quantum annealer. In particular, we propose a method of reformulating the problem as Quadratic Unconstrained Binary Optimization (QUBO) [21] required by D-Wave. To achieve this, we developed a Binary Integer Linear Programming (BILP) [19] formulation of the problem, which is then translated to QUBO in a similar way as shown in [14]. Finally, we discuss results obtained on the annealer. We attempt to find the optimal solution for selected instances of workflow scheduling problems, which are constrained by the deadline and have the lowest cost possible.

This paper is organized as follows. In Section 3 we provide the overview of quantum computation with the use of the quantum annealer. In Section 4 we present the precise formulation of the problem. In Section 5 we provide a complete description of transforming the workflow scheduling problem into a QUBO problem. First, the transformation to BILP is presented, which includes all the constraints necessary to be translated. Next comes BILP to QUBO problem translation, using methods from [14]. Finally, Sections 6 and 7 present full re-

[1] https://quantum-computing.ibm.com/
[2] https://www.rigetti.com/
[3] https://www.dwavesys.com/

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

sults with a detailed commentary, as well as discussion and suggestions for future work in this matter.

2 Related work

There is a significant body of knowledge available concerning the topic of scheduling workflows on cloud infrastructures. Due to its widespread adoption, most scheduling solutions operate on Infrastructure as a Service (IaaS) services. To the best of our knowledge, there is no other work directly addressing the problem of workflow scheduling on serverless infrastructures with help of quantum computing. In the presented case, the serverless infrastructure is represented by a Function as a Service (FaaS) type of service. For example, in [2] Arabnejad et al. propose a heuristic list scheduling algorithm with low computational complexity, designed for running workflows on IaaS. Work presented in this paper aims to provide similar features, albeit for FaaS and with help from quantum computing. In particular, we propose a method to create the final execution plan upfront in a short time and at low cost. One of the factors included in the presented algorithm is the sole cost of executing workflow tasks in the cloud. This problem was discussed in more detail in [30]. Zhou et al. addressed the problem of performance offered by cloud services with specific focus on the use case of running workflow applications. The matter of performance offered by serverless infrastructures was studied in [24], with focus on potential parallelism and application of FaaS as an infrastructure for large-scale scientific workflows. The specific topic of scheduling workflows on serverless infrastructures was addressed by Kijak et al. in [20], where they proposed a heuristic algorithm for scheduling workflows on cloud functions.

Although solving scheduling problems with quantum computers is a novelty, some examples of such work are available. In particular, a promising approach is to use the D-Wave quantum annealer based on a chimera graph [5]. A possible method of solving a shop job scheduling problem (JSP) [15] using D-Wave is shown in [29], however the proposed three-dimensional structure of the problem description is a strong limitation for scalability. In general, many NP-hard problems can be formulated as Ising problems [22] which can then be solved using a quantum annealer. For example [8] discusses the problem of finding maximum cliques in a graph. An important factor for solving problems with the use of quantum annealers is small machine size. In [25] the authors discuss heuristics for solving large-scale problems described in [8].

3 D-Wave computing overview

D-Wave 2000Q [5] is an adiabatic quantum computer that, unlike its universal counterparts (e.g. IBM-Q or Rigetti), cannot run algorithms implemented with the use of general quantum circuits. Therefore, in spite of its architecture offering a relatively large number of qubits, its usage is limited to certain types of optimisation problems. It is a fully analog machine, the result of which depends on the

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

value of the magnetic field applied to each qubit and the value of the connection between qubits (coupler). Programming the D-Wave 2000Q computer involves determining values of fields and associations. The aim of quantum annealing is to find the minimal energy state for the problem defined by a programmer. Annealing begins with an initial well-known quantum system for which the minimal energy state is known. Then it slowly switches to the quantum system related to the search problem. Ideally, during the whole process of switching, the system remains in its minimal energy state, so it should end up in the minimal energy state of the intended problem.

From a programmer's perspective, the problems to be solved must be formulated as an objective function using the Ising model [22] or, alternatively, a QUBO problem description [21]. Both approaches are isomorphic and can be transformed into each other in polynomial time. In this study, we choose to use QUBO formulation of the problem. In general, a QUBO problem consists of a matrix Q of size NxN, where N is the number of used binary variables and the size of the vector of binary variables that constitute the searched state. It can be converted without loss of generality to the upper triangular matrix as described in [14]. The actual problem solved by D-Wave is to find the final state of n binary variables $X = (x_1, x_2...x_n)$ that minimizes the objective function defined as

$$f(x) = \sum_i Q_{i,i} x_i + \sum_{i,j,i<j} Q_{i,j} x_i x_j. \tag{1}$$

Elements with the same indexes ($Q_{i,i}$) are responsible for "bias", i.e. the initial value of the magnetic field applied to the qubit, and elements with unequal indexes are responsible for the links between qubits (couplers). Minimizing such function is an NP-hard problem which makes it well suited for solving with help from the D-Wave 2000Q quantum computer.

The problem description does not limit the number of connections between qubits. However, the actual hardware is, in fact, a chimera graph (16x16 board of $K(4, 4)$ graphs, degree 6); thus, the problem graph must be mapped onto that architecture, which is achieved by minor-embedding. It is a process of mapping each logical binary variable present in a problem description to a group (called a chain) of physical qubits on the actual machine so all required connections are realized. For dense matrices there is probably no better method than minor-embedding based on complete graph embedding described in [11]. For sparse matrices it is reasonable to attempt heuristics to lower the qubit chain length (e.g. as described in [6]).

4 Workflow problem formulation

In its general form, the workflow application can be represented as a Directed Acyclic Graph, where each task is represented by a vertex. An example graph is depicted in Fig. 1. We assume that the serverless infrastructure consists of a finite number of machine types, albeit with an infinite number of instances for

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

each type. Each machine type has an associated cost. Each task has an associated running time for each machine type. Our model can be applied to any DAG.

The goal is to assign a machine type to each task with minimal total cost, while respecting the deadline. For simplicity, we will use the term *machine* instead of *machine type* from now on.

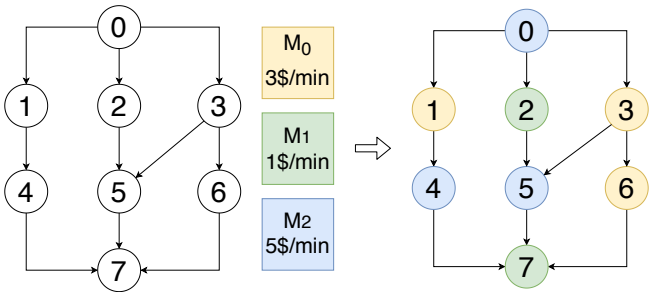

Fig. 1: Example workflow problem as a graph with task count t=8. We search for an assignment of each task to one of the machine types. For example, with machine count m=3 vector $K = [k_0, k_1, k_2]$ describes the cost of time unit execution on each machine while matrix $T = [\tau_{i,j}]_{8\times 3}$ describes the execution time of task i running on machine j

The formal definition of the problem addressed in this paper consists of:

- a time matrix $T = [\tau_{i,j}]_{t\times m}$ where t is the number of tasks, m is the number of machines and $\tau_{i,j}$ expresses the execution time of task with number i on the machine with number j;
- a machine cost per time unit vector $K = [k_i]_m$ measured in currency units per time unit. This matrix can be used to calculate the cost of running the workflow;
- a deadline d (an integer, limiting this value is crucial for the size of the resulting QUBO problem);
- a list Θ of paths from the vertex representing the first task to the vertex representing the final task (both inclusive) in a DAG representing the problem.

For the purpose of describing results, we apply the following terminology: a **correct** result meets all the constraints, the **(global) optimum** is a correct result which has the lowest cost possible for the problem instance, while a **wrong** result fails to meet at least one constraint.

5 Transformation to QUBO problem

The problem addressed in this paper is NP-hard [9]. We provide its translation to a QUBO problem which consists of two steps. First, we propose a transformation

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

of the problem to BILP, which is described in this section. The second step shows how to translate BILP to a QUBO problem using [14].

The general goal of solving a BILP problem (also called "0-1 Integer Programming" [19]) is to find, for a given vector $C = [c_i]_n$, n binary numbers $X = [x_1, ..., x_n]$, for which the function

$$f(x_1, ..., x_n) = C^T X = \sum_{j=1}^{n} c_j x_j \tag{2}$$

is minimal, subject to constraints indicated by the following linear equation:

$$Ax = b, \tag{3}$$

where $A = [a_{i,j}]_{n \times w}$ stores w constraints for n binary numbers.

Translation from initial formulation to BILP
In our problem, the BILP binary variables are defined as $x = [x_i]_n$ where $n = t \cdot m$ (see Section 4). Variable x_i is set to 1 if the task with number $(i \, mod \, t)$ runs on a machine with number $(i \, div \, t)$ where div is an integer division. Tab. 1 contains an example mapping of x_i parameters to task and machine combinations.

Table 1: The binary variables $X = [x_0, x_1, x_2 ... x_{31}]$ for BILP formulation of the sample workflow problem. Task count is $t = 8$ and machine count is $m = 4$. Binary variable x_i is set to 1 when task with number $(i \, mod \, t)$ runs on a machine with number $(i \, div \, t)$ and to 0 otherwise.

		Task							
		1	2	3	4	5	6	7	8
Machine	0	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
	1	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}
	2	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}
	3	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}	x_{30}	x_{31}

The BILP minimization function can subsequently be defined as the total cost of running tasks on selected machines

$$f(X) = \sum_{i=1}^{n} c_i x_i \ , \tag{4}$$

where $C = [c_i]_n$ is the cost vector and c_i indicates the cost of running task with number $(i \, mod \, t)$ on the machine with number $(i \, div \, t)$.

The cost vector needed for BILP formulation of the problem is obtained from the problem definition (Section 4) as follows: first, the cost matrix $[\gamma_{i,j}]_{t \times m}$ is created by multiplying $\gamma_{i,j} = \tau_{i,j} \cdot k_j$. Next, the cost vector C is obtained by row-by-row vectorization $[\tau_{i,j}]$ using $c_{i+t \cdot j} = \tau_{i,j}$. Finally, the time vector $T = [t_i]_n$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

is obtained in an analogous manner by calculating $t_{i+t \cdot j} = \gamma_{i,j}$.

In addition to the minimization function, there are also two types of constraints, the deadline and machine usage, where it is necessary to ensure that only a single machine will be selected for a given task.

Deadline constraints. We define a matrix $R = [r_{i,j}]_{n \times r}$, where r is the number of all possible paths in the workflow DAG. R stores the time for each machine-based task assignment $i \in [0..n-1]$, but only if the task belongs to the path $\theta \in [0, r-1]$ as follows:

$$r_{i,\theta} = \begin{cases} t_i & \text{if path } \theta \text{ contains task number } i \, mod \, t \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Next, we formulate a set of constraints corresponding to each path θ

$$\sum_{i=0}^{n-1} r_{i,\theta} x_i \leq d \, , \tag{6}$$

which we can rewrite as (for $D = [d]_n$)

$$Rx \leq D. \tag{7}$$

To describe constraints as part of BILP according to (3) we need to transform an inequality (7) into an equality. To achieve this, we rely on the fact that there always exists a slack variable s such that for natural numbers

$$a \in \mathbb{N}, Z \in \mathbb{N}, a \leq Z \implies \exists s \in N : a + s = Z.$$

Therefore it is necessary to extend R with additional columns that represent all necessary slack variables [14]. Generally a slack value s is a natural number, so we represent it using standard binary expansion. If we indicate

$$\sum_{i; task(i) \in \theta} \min_j \tau_{i,j} \tag{8}$$

as the minimum time of running tasks in path θ, then the number of binary variables b for storing slack variables for each path θ equals:

$$b(p) = log_2(|d - \sum_{i; task(i) \in \theta} \min_j \tau_{i,j}|). \tag{9}$$

Machine usage constraints. The next category of constraints comprises "one machine per task only" constraints. We define $U = [u_{i,j}]_{n \times t}$ such that for each task $s \in [0, t-1]$ and its machine-based position $i \in [0, n-1]$

$$u_{i,s} = \begin{cases} 1 & \text{if } i \, mod \, n = s \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

Therefore, each row of U indicates all machine-based positions of a task corresponding to that row. Then, to assure that only one machine is assigned to each task, the following constraint must be fulfilled:

$$\sum_{i=0}^{n-1} u_{i,t} x_i = 1. \tag{11}$$

If we indicate s as the total number of slack variables, the final BILP constraints matrix (3) is defined as $A = [a_{i,j}]_{(n+s)\times(r+t)}$ and combines all constraints together as follows:

$$a_{i,j} = \begin{cases} r_{i,j} & 0 \leq j < r \\ u_{i,j-r} & r \leq j < r+t. \end{cases} \tag{12}$$

To perform this operation, we need matrices of compatible sizes. Therefore, the matrix U must also be extended with slack variables, which are set to 0.

Vector b from (3) is defined in a similar manner:

$$b_i = \begin{cases} d & 0 \leq j < r \\ 1 & r \leq j < r+t. \end{cases} \tag{13}$$

Translation from BILP to QUBO problem.
Given matrices A, C and vector b a QUBO matrix can be calculated using the following formula (from [14]):

$$y = x^T C x + P \cdot (Ax - b)^T (Ax - b) = x^T C x + x^T D x + c = x^T Q x + c. \tag{14}$$

By dropping the additive constant c, the exact QUBO problem form, which is minimizing $x^T Q x$, can be formulated. However it is necessary to introduce two additional scalar parameters:

- P - relative strength of all constraints in relation to the objective function, see (14)
- S - weight required for balancing R and U values so that the constraints they represent are efficiently included in the QUBO problem. Namely, it replaces constraints defined by (11) with

$$\sum_{i=0}^{n-1} S u_{i,t} x_i = S. \tag{15}$$

Both mentioned parameters, along with the minor embedding chain strength, need to be balanced, to make sure that (1) the solution meets constraints and (2) the objective function is minimized. Finding proper values for these parameters is not a trivial task. It is important to mention that the resulting QUBO problem might have high resolution, which can result in errors due to the limited resolution of the annealer's Digital to Analog Converter (DAC), so the hardware conversion from QUBO to analog values may not be accurate enough.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

6 Results

In this section we describe the experimental results obtained using D-Wave 2000Q, compared with selected classical reference methods. In order to match the limitations of the existing quantum annealer, we considered four sample instances of the problem. Their DAG representation is shown in Fig. 2 and parameters in Tab. 2.

Table 2: The tested instances of the workflow problem (for definitions of parameters, see Section 4)

No.	Binary variable count	T	K	paths
1	8	$\begin{bmatrix} 6 & 3 & 12 & 9 \\ 2 & 1 & 4 & 3 \end{bmatrix}$	[1,4]	[[0,1,3],[0,2,3]]
2	10	$\begin{bmatrix} 6 & 3 & 12 & 9 & 6 \\ 2 & 1 & 4 & 3 & 2 \end{bmatrix}$	[1,4]	[[0,1,4],[0,2,4],[0,3,4]]
3	15	$\begin{bmatrix} 6 & 3 & 12 & 9 & 6 \\ 2 & 1 & 4 & 3 & 2 \\ 4 & 2 & 8 & 6 & 4 \end{bmatrix}$	[1,5,2]	[[0,1,4],[0,2,4],[0,3,4]]
4	18	$\begin{bmatrix} 12 & 6 & 42 & 18 & 30 & 24 \\ 4 & 2 & 14 & 6 & 10 & 8 \\ 8 & 4 & 28 & 12 & 20 & 16 \end{bmatrix}$	K=[8,18,6]	[[0,1,3,5],[0,1,4,5],[0,2,4,5]]

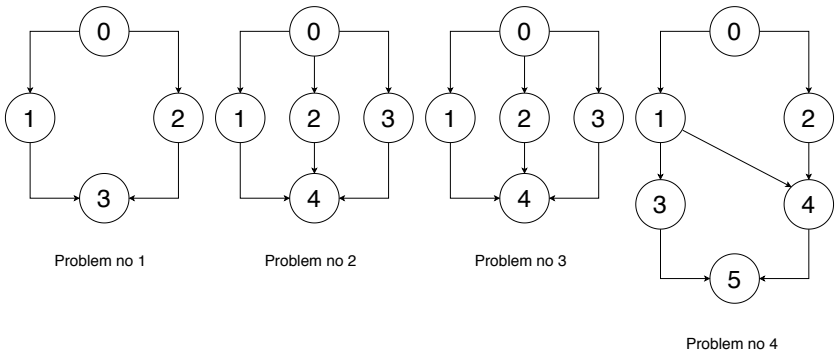

Fig. 2: Graph representations of the tested workflow problems.

Finding parameters: P, S and chain strength. Parameters needed for QUBO problem formulation are dependent on the specific problem instance. For the purpose of this research, parameters were obtained using a metaheuristic. First, (1), we find an initial value of P basing on [14]. The potential initial

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

value of parameter S should be similar to values in vector T in order to achieve proper balance between constraints. Then, (2), we search the discrete space of possible pairs (P, S) for values relatively close to the initial values, using a classical solver – Gurobi[4]. Next, (3), we set the chain strength between physical qubits for the minor-embedded problem basing on the D-Wave guidebook's [11] suggestion that it should be large enough to keep chains and small enough not to cover the actual problem. We have found that it should approximate the largest values of the Q matrix. Finally, (4), the selected chain strength is used as an input for the dwave.embedding library, which performs actual minor embedding. The final parameter values are presented in Tab. 3. Deadline values for all four problems were selected in such a way that they yield similar percentages of correct solutions.

Table 3: Values of parameters P, S and chain strength found for each size of the workflow problem for the given deadline (which is predefined, but has an impact on the parameters' values). The rightmost column represents the percentage of correct solutions in relation to all possible solutions.

No.	Binary variable number	Deadline	P	S	Chain strength	Percentage of correct solutions
1	8	19	8	10	1200	62,5%
2	10	19	14	25	6650	56,25%
3	15	17	11	10	2800	54,3%
4	18	70	6	40	18000	46,91%

Reference methods. The problems discussed in this paper are small enough to be solved using classical methods. The following four classical methods have been used to verify D-Wave machine results:

- A brute-force method using initial problem formulation. It was used to calculate exact results, along with minimal energy, through direct use of the objective function. (1),
- GNU Linear Programming Kit[5] library for solving BILP problem prior to its translation to QUBO. This method always finds the global optima.
- Gurobi sampler for the QUBO problem without minor-embedding. This method always finds the global optima, provided that parameters P and S are set up properly.
- Gurobi sampler for QUBO problem with minor-embedding, requiring three parameters (P, S, chain strength). The summary of the results is presented in Tab. 4.

Experiments with D-Wave. We perform experiments using the D-Wave 2000Q 5.0 machine sampled 2000 times with annealing time set to $8\mu s$. Fig. 3 shows

[4] http://www.gurobi.com
[5] https://www.gnu.org/software/glpk/

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

Table 4: Gurobi solutions table. The rightmost column shows whether the global optimum was found, with the absolute energy error between the lowest Gurobi solution and the brute force global optimum.

No.	Binary variables number	Global optimum found
1	8	YES
2	10	YES
3	15	YES
4	18	NO, wrong (22831)

the energy distribution of the actual results. For Problem 1 and Problem 2, the minimal energy corresponds to the global optimum, while for Problem 3 it indicates a correct solution, but not the best possible one. All energies found for the most complex Problem 4 correspond to wrong solutions. It can be noticed that the count of wrong solutions grows along with the size of the solution space. In general, the obtained characteristics of energies remain similar to [17].

Details of the results are described in Tab. 5, where the number of correct solutions found by D-Wave and their relation to the total number of correct solutions is presented. The results are compared to the global optima obtained by the brute force method. It can be noted that D-Wave results are comparable to Gurobi results. In the first two problems, workflow optimization was solved exactly and the global optimum was found. In problem 3, the Gurobi solution was still optimal, while the D-Wave solution was correct, but not the best ($\tilde{1}2\%$ worse in terms of cost; 33th out of 132 correct solutions). For problem 4 neither sampler found the global optimum.

Table 5: Summary of D-Wave 2000Q results. The fourth column presents the number of D-Wave unique correct solutions in relation to the total number of brute force correct solutions. The fifth column indicates whether the global optimum was found, listing the absolute energy error between the lowest D-Wave solution and the brute-force global optimum. The two rightmost columns refer to the execution time and cost of the lowest D-Wave solution respectively (*min* means global optimum cost).

No.	Binary variables number	Correct solutions samples (from 2000 samples)	Unique correct solutions	Global optimum found	Time	Cost
1	8	98	10 (100%)	YES	19(d=19)	34(min=34)
2	10	27	14 (77.8%)	YES	16(d=19)	40(min=40)
3	15	4	4 (3.03%)	NO (6)	16(d=17)	45(min=40)
4	18	0	0 (0%)	NO (65862)	N/A	N/A

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

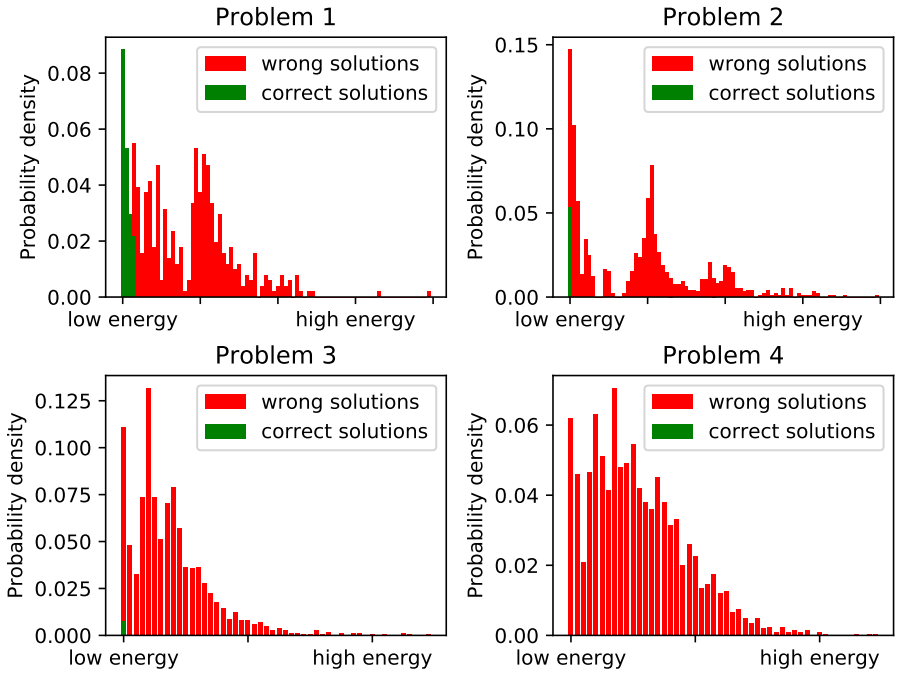

Fig. 3: Histograms of results for the four types of workflow problems tested. The x axis represent values of energies equal to the value of minimized objective function (1). The y axis represents the probability density. For Problems 1-3 the correct solutions found are shown on the left-hand side of the spectrum.

7 Conclusions and future work

In this paper we showed that it is possible to translate the workflow scheduling problem into a QUBO problem, execute it on a quantum annealer and achieve not only correct, but also globally optimal results for some of the analyzed problem instances. However, the presented method of adapting the scheduling challenge for D-Wave significantly increases the size of the problem. For example, the 18-binary variable problem required each of 39 QUBO problem variables to be represented by 11 physical qubits; thus the initial problem with a solution space size of $= 3^6 = 729$ ($m^t, m = 3, t = 6$, see Section 4) was converted into a problem with size $2^{429} \sim 10^{143}$ (QUBO problem with 39 variables, 11 qubits each, $39 \cdot 11 = 429$). For such a large QUBO it is difficult for the quantum annealer to find the lowest energy solution. Therefore, in the future we will focus on solutions such as domain wall encoding [7] to reduce the number of binary variables.

However, it is worth noting that for larger problems the brute force method would no longer be usable because of its exponential complexity. This leaves a

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

space for experimenting with a quantum annealer for larger instances, as the annealing process is very fast.

To sum up, this work proved that it is possible to solve workflow scheduling problems with the use of currently existing quantum annealers. Future work might involve finding a better translation of the problem to a QUBO problem, making the Q matrix more sparse and using minor-embedding heuristics. Additionally, the problem solution could be tested on the Pegasus machine [12] upon its release. It may also be possible to divide the problem into smaller pieces – this means dividing the original problem or dividing the QUBO problem (or even using both methods in parallel) with tools such as D-Wave QBSolv [10]. Another interesting direction of research would be to compare the performance of the presented problem on the D-Wave 2000Q machine with a non-quantum Fujitsu digital annealer [28] as both machines are designed to solve problems with a similar approach but different hardware.

Acknowledgements. The research presented in this paper has been partially supported by the National Science Centre, Poland, Grant no. 2016/21/B/ST6/01497. The authors also thank Piotr Gawron, Bartłomiej Gardas, Andy Mason and Piotr Nowakowski for helpful remarks.

References

1. Abbott, B.P., Abbott, R., Abbott, T., et al.: Observation of gravitational waves from a binary black hole merger. Physical review letters **116**(6), 061102 (2016)
2. Arabnejad, H., Barbosa, J.G., Prodan, R.: Low-time complexity budget–deadline constrained workflow scheduling on heterogeneous resources. Future Generation Computer Systems **55**, 29–40 (2016)
3. Baldini, I., Castro, P., Chang, K., Cheng, P., et al.: Serverless computing: Current trends and open problems. In: Research Advances in Cloud Computing, pp. 1–20. Springer (2017)
4. Berriman, G., Good, J., Laity, A., et al.: Montage: A grid enabled image mosaic service for the national virtual observatory. In: Astronomical Data Analysis Software and Systems (ADASS) XIII. vol. 314, p. 593 (2004)
5. Bian, Z., Chudak, F., Macready, W.G., Rose, G.: The Ising model: teaching an old problem new tricks. D-wave systems **2** (2010)
6. Cai, J., Macready, W.G., Roy, A.: A practical heuristic for finding graph minors. arXiv preprint arXiv:1406.2741 (2014)
7. Chancellor, N.: Domain wall encoding of discrete variables for quantum annealing and QAOA. Quantum Science and Technology **4**(4) (2019)
8. Chapuis, G., Djidjev, H., Hahn, G., Rizk, G.: Finding maximum cliques on the d-wave quantum annealer. Journal of Signal Processing Systems **91**(3), 363–377
9. Coffman, E.G., Bruno, J.L.: Computer and job-shop scheduling theory. John Wiley & Sons (1976)
10. D-Wave Systems: D-Wave Initiates Open Quantum Software Environment (2017), https://www.dwavesys.com/press-releases/d-wave-initiates-open-quantum-software-environment
11. D-Wave Systems Inc.: D'wave problem solving handbook. https://docs.dwavesys.com/docs/latest/_downloads/09-1171A-A_Developer_Guide_Problem_Solving_Handbook.pdf

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40

12. Dattani, N., Szalay, S., Chancellor, N.: Pegasus: The second connectivity graph for large-scale quantum annealing hardware. arXiv preprint arXiv:1901.07636 (2019)
13. Deelman, E., Gannon, D., Shields, M., Taylor, I.: Workflows and e-science: An overview of workflow system features and capabilities. Future Generation Computer Systems **25**(5), 528–540 (2009)
14. Glover, F., Kochenberger, G., Du, Y.: A Tutorial on Formulating and Using QUBO Models. arXiv preprint arXiv:1811.11538 (2018)
15. Graham, R.L.: Bounds for certain multiprocessing anomalies. The Bell System Technical Journal **45**(9), 1563–1581 (Nov 1966). https://doi.org/10.1002/j.1538-7305.1966.tb01709.x
16. Grover, L.K.: A Fast Quantum Mechanical Algorithm for Database Search. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. pp. 212–219. STOC '96 (1996)
17. Jałowiecki, K., Więckowski, A., Gawron, P., Gardas, B.: Parallel in time dynamics with quantum annealers. arXiv preprint arXiv:1909.0429
18. Jordan, S.: Quantum algorithms zoo web page. https://quantumalgorithmzoo.org/
19. Karp, R.M.: Reducibility among Combinatorial Problems, pp. 85–103. Springer US, Boston, MA (1972), https://doi.org/10.1007/978-1-4684-2001-2_9
20. Kijak, J., Martyna, P., Pawlik, M., Balis, B., Malawski, M.: Challenges for scheduling scientific workflows on cloud functions. In: 11th IEEE International Conference on Cloud Computing, CLOUD 2018, San Francisco, CA, USA, July 2-7, 2018. pp. 460–467. IEEE Computer Society (2018). https://doi.org/10.1109/CLOUD.2018.00065
21. Lewis, M., Glover, F.: Quadratic unconstrained binary optimization problem preprocessing: Theory and empirical analysis. Networks **70**(2), 79–97 (2017)
22. Lucas, A.: Ising formulations of many NP problems. Frontiers in Physics **2**, 5 (2014). https://doi.org/10.3389/fphy.2014.00005, https://www.frontiersin.org/article/10.3389/fphy.2014.00005
23. Maechling, P., Chalupsky, H., Dougherty, M., et al.: Simplifying construction of complex workflows for non-expert users of the southern california earthquake center community modeling environment. ACM SIGMOD Record **34**(3), 24–30 (2005)
24. Pawlik, M., Figiela, K., Malawski, M.: Performance considerations on execution of large scale workflow applications on cloud functions. arXiv preprint arXiv:1909.03555 (2019)
25. Pelofske, E., Hahn, G., Djidjev, H.: Solving Large Maximum Clique Problems on a Quantum Annealer. In: Feld, S., Linnhoff-Popien, C. (eds.) Quantum Technology and Optimization Problems. pp. 123–135. Springer International Publishing, Cham
26. 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)
27. Spillner, J., Mateos, C., Monge, D.A.: FaaSter, Better, Cheaper: The Prospect of Serverless Scientific Computing and HPC. In: Latin American High Performance Computing Conference. pp. 154–168. Springer (2017)
28. Tsukamoto, S., Takatsu, M., Matsubara, S., Tamura, H.: An accelerator architecture for combinatorial optimization problems. Fujitsu Sci. Tech. J **53**(5), 8–13 (2017)
29. Venturelli, D., M.D.J.J., Rojo, G.: Quantum Annealing Implementation of Job-Shop Scheduling. arXiv:1506.08479 (2015)
30. Zhou, A.C., He, B., Liu, C.: Monetary cost optimizations for hosting workflow-as-a-service in iaas clouds. IEEE Transactions on Cloud Computing **4**(1), 34–48 (2015)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_40