Distributed Quantum Annealing on D-Wave for the single machine total weighted tardiness scheduling problem

Wojciech Bożejko[1][0000−0002−1868−8603], Jarosław Pempera[1][0000−0002−0614−0085], Mariusz Uchroński[1,2][0000−0002−9185−1841], and Mieczysław Wodecki[3][0000−0001−8188−4503]

[1] Department of Control Systems and Mechatronics
Wrocław University of Science and Technology,
Janiszewskiego 11-17, 50-372 Wrocław, Poland
wojciech.bozejko@pwr.edu.pl, jaroslaw.pempera@pwr.edu.pl
[2] Wroclaw Centre for Networking and Supercomputing
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław
mariusz.uchronski@pwr.edu.pl
[3] Department of Telecommunications and Teleinformatics,
Wrocław University of Science and Technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
mieczyslaw.wodecki@pwr.edu.pl

Abstract. In the work, we are proposing a new distributed quantum annealing method of algorithm construction for solving an NP-hard scheduling problem. A method of diversification of calculations has been proposed by dividing the space of feasible solutions and using the fact that the quantum annealer of the D-Wave machine is able to optimally solve (for now) small-size subproblems only. The proposed methodology was tested on a difficult instance of a single machine total weighted tardiness scheduling problem proposed by Lawler.

Keywords: discrete optimization · scheduling · quantum algorithm · quantum annealing

1 Introduction

The idea of quantum computing and computers emerged in the 1980s as a result of work of Richard Feynman, Paul Benioff and Yuri Manin. An approach, known as the gate model, expresses the interactions between qubits as quantum gates. Quantum gates, because of quantum physics, operate differently than classical electrical gates such as AND or OR. In a quantum computer with a gate model, there is no AND gate. Instead, there are Hadamard gates and Toffoli gates. Unlike many classical logic gates, quantum logic gates are reversible.

Instead of expressing the problem in terms of quantum gates, the user presents it as an optimization problem and the quantum annealing computer tries to find

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_15

the best solution. D-Wave Systems makes quantum annealing computers available to the public today, proposing an approach to quantum computing that is admittedly limited to the use of quantum annealing, but that fits well with the needs of the operations research discipline. This paper presents a method for solving a one-machine problem with a sum-of-cost criterion based on quantum annealing.

2 The specificity of quantum annealers

Quantum annealing (QA) [8] is a hardware optimization method implemented through quantum fluctuations, instead of – as in simulated annealing – temperature fluctuations. It can be treated as metaheuristics, because the solutions achieved on a quantum computer designed to perform quantum annealing do not have a guarantee of optimality. This process is carried out by a special type of quantum computer searching for the minimum energy Ising Hamiltonian configuration, the ground states of which represent the optimal solution to the problem under consideration. The Ising model, traditionally used in mechanics, is formulated for variables denoting the directions of spins: 'up' and 'down' – which corresponds to the values of $+1$ and -1. The energy function is given by the formula: $E_{Ising}(s) = \sum_{i=1}^{N} h_i s_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} J_{i,j} s_i s_j$, where N is the number of qubits, $s_i \in \{+1, -1\}$, and the vector h and the matrix J are the Ising Hamiltonian coefficients. In practice, it is easier to adapt the problem under consideration to the binary quadratic model, BQM, traditionally considered in operations research, using binary variables: $E_{BQM}(v) = \sum_{i=1}^{N} a_i v_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} b_{i,j} v_i v_j + c$, where $v_i \in \{1, +1\}$ or $\{0, 1\}$ and $a_i, b_{i,j}, c$ are some real numbers depending on the instance of the problem being solved. Of course, the BQM model is a generalization of the Ising model.

In practice, on D-Wave machines, a constrained model is used to solve optimization problems, which can be formulated as follows: minimize an objective

$$\sum_i a_i x_i + \sum_{i<j} b_{ij} x_i x_j + c \tag{1}$$

subject to constrains

$$\sum_i a_i^{(c)} x_i + \sum_{i<j} b_{ij}^{(c)} x_i x_j + c^{(c)} \leq 0 \quad c = 1, \ldots, C_{ineq}, \tag{2}$$

$$\sum_i a_i^{(c)} x_i + \sum_{i<j} b_{ij}^{(c)} x_i x_j + c^{(c)} = 0 \quad c = 1, \ldots, C_{eq}, \tag{3}$$

where x_i, $i = 1, \ldots, n$ can be binary or integer variables, a_i, b_{ij} and c are real values and C_{ineq}, C_{eq} are the number of inequality and equality constraints respectively.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_15

3 Single machine total weighted tardiness problem

In the considered problem of tasks scheduling on a single machine with minimization of total delays costs (marked in the literature by $1||\sum w_i T_i$) we have a set of tasks that must be performed on one machine. Each task has an associated execution time, desired completion date, and the weight of the late penalty function. The order in which the tasks are performed should be determined, minimizing the sum of the costs of delays. It is one of the most studied, strongly NP-hard problems with total-cost objective functions. The first work on this topic, Rinnooy Kan et al. [10] has been published in the mid-1970s. Optimal algorithms, based on the dynamic programming or branch and bound methods, were published by: Congram et al. [5], and Wodecki [11]. These are mainly metaheuristics that have been widely used since the 1990s: tabu search of Bożejko et al. [3], ant colony optimization algorithm (Den Basten et al. [6]). Extensive reviews of the literature on scheduling problems with due dates was also presented by Adamu and Adewumi [1]. The literature also deals with problems with random execution times or completion dates (Rajba and Wodecki [9], Bożejko et al. [2]).

The single-machine problem of minimizing the sum of costs delays (*Total Weighted Tardiness Problem*, abbreviated AS TWT), marked in the literature by $1||\sum w_i T_i$, can be formulated as follows: the set of tasks is given $\mathcal{J} = \{1, 2, \ldots, n\}$. For the $i \in \mathcal{J}$ task, let us define: p_i – processing time, d_i – due date, and w_i – weight of the cost function for the task's tardiness. Each task must be performed on the machine, the following restrictions must be met: (a) the machine can perform at most one task at any given time, (b) task execution cannot be interrupted, (c) the task execution may begin at time zero.

Any solution to the TWT problem can be represented by the sequence $S_1, S_2, \ldots, S_n$ of times when tasks meet the constraints:

$$S_i + p_i \leq S_j \vee S_j + p_j \leq S_i,\ i \neq j,\ i, j = 1, 2, \ldots, n, \tag{4}$$

$$S_i \geq 0,\ i = 1, 2, \ldots, n \tag{5}$$

Solution $S_1, S_2, \ldots, S_n$ can be represented by the order of execution of tasks expressed by a permutation $\pi \in \Pi$ of elements of the set $\mathcal{J}$, where Π is the set of all such permutations.

Let $C_{\pi(i)} = S_{\pi(i)} + p_{\pi(i)}$ be completion time and $d_{\pi(i)}$ the *due date* of a task $\pi(i) \in \mathcal{J}$. Then $T_{\pi(i)} = \max\{0, C_{\pi(i)} - d_{\pi(i)}\}$ is a *tardiness*, and $f_{\pi(i)}(C_{\pi(i)}) = w_{\pi(i)} T_{\pi(i)}$ *tardiness weight*. For any permutation of $\pi \in \Pi$, penalty for tasks tardiness (solution cost) is

$$\mathcal{F}(\pi) = \sum_{i=1}^{n} f_{\pi(i)}(C_{\pi(i)}) = \sum_{i=1}^{n} w_{\pi(i)} T_{\pi(i)}. \tag{6}$$

In the considered problem, the optimal order $\pi^* \in \Pi$ in which tasks should be performed should be determined minimizing the total penalty. In the further part of the work, we present a new method for solving the TWT problem and

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_15

an algorithm in which a quantum computer performing quantum annealing was used for calculations.

4 Solution method

The idea of solving the problem is based on the divide and conquer method. It can be summarized as follows.

From n element set of tasks $\mathcal{J}$ find all k $(0 < k < n)$ element subsets. Next:

1. Choose one of the subsets. We denote it as $\mathcal{K}$. Let $C_{\mathcal{K}} = \sum_{z\in\mathcal{K}} p_z$ be the due date for $\mathcal{K}$.
2. Using the QA quantum annealing algorithm, determine the order of execution, starting from $C_{\mathcal{K}}$ for the remaining tasks, i.e. tasks from the set $\overline{\mathcal{K}} = \mathcal{J} \setminus \mathcal{K}$. Let $F(\overline{\mathcal{K}})$ be the cost of executing them.
3. Find a set of all element permutations from $\mathcal{K}$.
4. For the δ permutation, calculate the F_δ penalty for performing tasks in the order δ.
5. Calculate $F_{\min}(\mathcal{K})$, minimum after all permutations, from $F_\delta + F(\overline{\mathcal{K}})$.
6. Perform 1–5 consecutively for all k elementary subsets of the task set $\mathcal{J}$.

The minimum, after subsets, of the $F_{\min}(\mathcal{K})$ values determined in 5th point is a solution to the problem considered in the study.

In the further part of the work, we assume that the optimization algorithms run on a quantum computer determine optimal solutions. If not, then the solution to the problem discussed in this paper is suboptimal.

Let TWT (A, α) be a sub-problem of the TWT problem under consideration, in which $A \subseteq \mathcal{J}$ and α is the starting point for the tasks from the A set. With these markings, TWT $(\mathcal{J}, 0)$ is the problem under consideration in this paper.

For a natural number k $(0 < k < n)$, $L = \binom{n}{k}$ is the number of k elementary subsets (combinations) of the elements of the set $\mathcal{J}$. Let

$$\mathcal{A} = \{A_1, A_2, \ldots, A_L\}, \tag{7}$$

will be a set of such k elementary subsets. Come on, through

$$\overline{A}_i = \mathcal{J} \setminus A_i,\ A_i \in \mathcal{A}$$

we denote the completion of A_i in the task set $\mathcal{J}$.

We consider a subset of $A_i \in \mathcal{A}$. We assumed that $|A_i| = k,\ 0 < k < n$. From the set A_i we can generate a $\tau = k!$ permutation. Let

$$\Phi^i = \{\pi_1^i, \pi_2^i, \ldots, \pi_\tau^i\} \tag{8}$$

will be a set of these permutations. Cost of executing tasks in the order π_j^i $(i = 1, 2, \ldots, L,\ \ j = 1, 2, \ldots, \tau)$

$$\mathcal{F}(\pi_j^i) = \sum_{l=1}^{k} \max\{0, C_{\pi_j^i(l)} - d_{\pi_j^i(l)}\} \cdot w_{\pi_j^i(l)}, \tag{9}$$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_15

where the completion time of the task $\pi_j^i(l)$ is $C_{\pi_j^i(l)} = \sum_{s=1}^{l} p_{\pi_j^i(s)}$, for $l = 1, 2, \ldots, k$.

For the A_i set, $C(A_i) = \sum_{l \in A_i} p_l$ is the end of all tasks from A_i. Let $\overline{\pi}_i$ be the solution to the problem TWT $(\overline{A}_i, C(A_i))$ with the value of $\mathcal{F}(\overline{A}_i)$ determined by the algorithm QA $(\overline{A}_i, C(A_i))$ $(i = 1, 2, \ldots, L)$. Then the set

$$\Pi^i = \{(\pi_j^i, \overline{\pi}^i) : \ \pi_j^i \in \Phi^i, \ j = 1.2, \ldots, \tau\},$$

contains permutations of $\mathcal{J}$ tasks resulting from concatenation of π_j^i elements of A_i set with $\overline{\pi}^i$ elements of $\overline{A}_i$, which is a solution to the TWT problem determined by the Quantum Annealing QA$(\overline{A}_i, C(A_i))$. The cost of the permutation $(\pi_j^i, \overline{\pi}^i) \in \Pi^i$ is $\mathcal{F}(\pi_j^i, \overline{\pi}^i) = \mathcal{F}(\pi_j^i) + \mathcal{F}(\overline{A}_i)$. Then $\mathcal{F}_i^* = \min\{\mathcal{F}(\pi_j^i, \overline{\pi}^i) : \ j = 1, 2, \ldots, \tau\}$ is the value of the optimal[1] solutions for permutations from the set Π^i, i.e. permutations in which the first k positions include tasks from the set A_i and on the following $k+1, k+2 \ldots, n$ tasks from the set $\overline{A}_i$.

Remark 1. For the set of $\mathcal{J}$ tasks and a natural number $(k, \ 0 < k < n)$, if the QA algorithm $(\overline{A}_i)$ determines optimal solutions, then $\mathcal{F}^* = \min\{\mathcal{F}_i^* : \ i = 1, 2, \ldots, L\}$, is the optimal value of the solution of the TWT problem.

Algorithm 1 shows the Distributed Quantum Annealing (DQA) scheme for determining, on a quantum computer, a solution to the single machine total weighted tardiness problem.

Algorithm 1: Distributed Quantum Annealing DQA

Input : $\mathcal{J}$ – a set of tasks, p_i, d_i, w_i – task parameters,
k – number of elements of generated subsets;
Output: $\mathcal{F}^*$ – penalty function value (solution of TWT problem);

```
Generate 𝒜 containing all k - elementary subsets of n - element set;
for all A_i ∈ 𝒜 do
    Ā_i = 𝒥 \ A_i;   C(A_i) = Σ_{l∈A_i} p_l;
    Find the solution value F(Ā_i) of TWT problem (Ā_i, C(A_i)) using the QA;
    According to (8) determine the set of permutations Φ^i elements from A_i;
    for all π_j^i ∈ Φ^i do
        calculate F(π_j^i) basis on (9);
    F_i^* = min{F(π_j^i) + F(Ā_i) : π_j^i ∈ Φ^i};
F^* = min{F_i^* : i = 1, 2, ..., |𝒜|};
```

Example 1. for $n = 8$ tasks $(\mathcal{J} = \{1, 2, 3, 4, 5, 6, 7, 8\})$ and $k = 3$, a set of 3-element subsets of $\mathcal{A}$ has $\binom{n}{k} = \binom{8}{5} = 56$ items. Using the quantum annealing algorithm, 56 sub-problems have to be solved. Each 3-element subset has $3! = 6$ strings. The total of such strings is $56 \cdot 6 = 336$.

[1] Assuming that quantum annealing will generate an optimal solution.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_15

5 Implementation on DWave quantum annealer

The implementation of the algorithm uses a constrained quadratic model with a constraints available under Ocean Developer Tools for D-Wave Systems. Ocean software is a suite of tools D-Wave Systems provides on the D-Wave GitHub repository for solving hard problems with quantum computers.

For solving TWT problem we introduce integer variables S_i, T_i and binary variables $x_{i,j}$, $i \neq j$, $i, j \in \{1, \ldots, n\}$ which equals to 1 if job i precedes job j.

Our aim is to minimize:

$$\sum_i w_i T_i \tag{10}$$

Subject to constrains:

$$S_j - T_j + p_j - d_j \leq 0 \quad j = 1, \ldots, n, \tag{11}$$

$$-T_j \leq 0 \quad j = 1, \ldots, n, \tag{12}$$

$$S_k - S_j + (p_j - p_k)x_{jk} + 2(S_j - S_k)x_{jk} + p_k \leq 0 \quad j < k,\ j, k = 1, \ldots, n, \tag{13}$$

$$-S_j \leq -S_0 \quad j = 1, 2, \ldots, n. \tag{14}$$

6 Results

Computer experiments have been conducted in D-Wave Leap environment on `hybrid_constrained_quadratic_model_version1p` solver executed on a North America quantum annealer. Exact QPU (Quantum Processing Unit) processing time has been measured.

Case Study. An instance taken from the work of Lawler [7] has been used for an experiment for checking usefulness of the proposed methodology, due to the limited quantum machine time available per month. Data of the used instance is shown in the Table 1.

i	1	2	3	4	5	6	7	8
p_i	121	79	147	83	130	102	96	88
d_i	260	266	269	336	337	400	683	719
w_i	1	1	1	1	1	1	1	1

Table 1. Lawler test instance

The weight for all tasks have been set as $w_i = 1$, $i = 1, 2, \ldots, n$ (original instance is dedicated for the single machine problem with total tardiness cost function (without weights), however its difficulty's is significant (it is hard to solve). The optimal value is 755 (see Lawler [7]).

It is important to note, that it was impossible to achieve optimal solution by execution QA algorithm on the whole Lawler instance on D-Wave machine – percentage relative errors of the obtained solutions were very big, not

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_15

less than 20.17% (regardless of the number of repetitions of the QA algorithm calls). However, as we can see above, it was possible to obtain optimal solution $(1, 2, 4, 6, 5, 7, 8, 3)$ in $i = 2$ iteration by the DQA method proposed in this work.

7 Conclusions

The study considers the single machine tasks scheduling problem with the criterion of minimizing the weighted sum of tardiness. A distributed quantum annealing DQA algorithm has been proposed. Currently, the possibilities of quantum annealers (they are only produced by D-Wave company) allow us for the optimal solution of very small instances, specifically for the problem under consideration, up to $n = 5$. The application of the proposed DQA approach made it possible to determine the optimal solution for a very difficult Lawler example for the number of tasks $n = 8$. The proposed methodology allows for optimal solving of similar problems (eg. TSP) and larger sizes.

Acknowledgments. Calculations have been partially carried out using resources provided by Wroclaw Centre for Networking and Supercomputing (http://wcss.pl), grant No. 96.

References

1. Adamu, M.O., Adewumi, A.O.: A survey of single machine scheduling to minimize weighted number of tardy jobs, J. Ind. Manag. Optim. **10**, 219–241 (2014)
2. Bożejko, W., Rajba, P., Wodecki, M.: Stable scheduling of single machine with probabilistic parameters, Bull. Pol. Acad. Sci. Tech. Sci. **65**, 219–231 (2017)
3. Bożejki W., Grabowski J., Wodecki M.: Block approach-tabu search algorithm for single machine total weighted tardiness problem, Computers& Industrial Engineering **50**, 1–14 (2006)
4. Chenga, T.C.E., Nga, C.T., Yuanab, J.J., Liua, Z.H.: Single machine scheduling to minimize total weighted tardiness, Eur. J. Oper. Res. **165**, 423–443 (2005)
5. Congram, R.K., Potts, C.N., van de Velde, S.L.: An Iterated Dynasearch Algorithm for the Single-Machine Total Weighted Tardiness Scheduling Problem, Informs J. Comput. **14**, 52–67 (2002)
6. Den Basten, M.; Stützle, T.: Ant colony optimization for the total weighted tardiness problem, Precedings of PPSN–VI, Eur. J. Oper. Res. **1917**, 611–620 (2000)
7. Lawler E.L.: A "Pseudopolynomial" Algorithm for Sequencing Jobs to Minimize Total Tardiness, Annals of Discrete Mathematics **1**, 331–342 (1977).
8. McGeoch, C. C.: Theory versus practice in annealing-based quantum computing. Theoretical Computer Science **816**, 169–183 (2020)
9. Rajba, P.; Wodecki, M. Stability of scheduling with random processing times on one machine, Applicationes Mathematicae **39**, 169–183 (2012)
10. Rinnooy Kan, A.H.G., Lageweg, B.J., Lenstra, J.K.: Minimizing total costs in one-machine scheduling, Oper. Res. **25**, 908–927 (1975)
11. Wodecki W.: A Branch-and-Bound Parallel Algorithm for Single-Machine Total Weighted Tardiness Problem, International Journal of Advanced Manufacturing Technology **37**, 996–1004 (2008)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08760-8_15