Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

Solving Complex Combinatorial Optimization Problems Using Quantum Annealing Approaches

Vasileios K. Mappas[a]*, Bogdan Dorneanu[a], and Harvey Arellano-Garcia[a]

[a] FG Prozess, und Anlagentechnik, Brandenburgische Technische Universität, Cottbus, Germany
* Corresponding Author: vasileios.mappas@b-tu.de.

ABSTRACT

Currently, state-of-the-art approaches to solving complex optimization problems have focused solely on methods requiring high computational time and unable to find the global optimal solution. In this work, a methodology based on quantum computing is presented to overcome these drawbacks. The novelty of this framework stems from the quantum computer's architecture and taking into consideration the quantum phenomena that take place to solve optimization problems with specific structure. The proposed methodology includes steps for the transformation of the initial optimization problem into an unconstrainted optimization problem with binary variables and its embedding onto a quantum device. Moreover, different resolution levels for the transformation step and different architectures for the embedding process are utilized. To illustrate the procedure, a case study based on Haverly's pooling and blending problem is examined while demonstrating the potential of the proposed approach. The results indicate that the succinct formulation exhibited higher success rate during the embedding procedure for the different examined architectures, and the quantum annealing solver exhibited the best performance among the various solvers investigated. This highlights the potential of the approach for solving this type of problems with the rapid development and improvement of quantum hardware and expanding it to more complex chemical engineering optimization systems.

Keywords: Optimization, Scheduling, Algorithms, Quantum Computing, Quantum Annealing

1. INTRODUCTION

Combinatorial optimization (CO) is a branch of mathematical optimization that is applied for a broad set of applications in the fields of industrial engineering, transportation or manufacturing. Finding the global solutions for this type of problems is challenging, due to its NP-hard nature [1].

In real-world applications, processes are described by highly nonlinear and nonconvex expressions leading to an exponential growth of the number of local solutions as the number of variables increases [2].

Several strategies, based on classical optimization techniques, heuristics and neural network methods, have been proposed to solve optimization problems [3]. Nonetheless, these state-of-the-art methods exhibit the following limitations:

1. High computational timeframes are required in deterministic optimization algorithms.
2. Solution quality is not guaranteed in heuristic approaches, leading to suboptimal results.
3. The training and the tuning of the neural networks, as well as the use of small instances of problems affects the solution.

Quantum computing (QC) is a rapidly growing research field and promising technology that differentiates itself from classical computers and utilize quantum phenomena such as entanglement and superposition to perform calculations and simulations. QC is capable of providing polynomial-time solutions for problems in the BQP (bounded error, quantum, polynomial time) complexity class [4].

To address these drawbacks, this contribution proposes a solution framework, based on QC, that involves formulating the original optimization problem as a Quadratic Unconstrained Binary Optimization (QUBO) problem and embedding it onto a quantum device.

The remainder of the contribution is structured as follows: in Section 2 the methodology for reformulating the CO problem into QUBO is presented. The proposed solution framework is applied for a case study in Section 3. The key results and future work required are summarized in Sections 4 and 5, respectively.

2. METHODOLOGY

In the first step, the original CO problem is formulated as a QUBO problem, where continuous and integer variables are transformed into binaries and its formulation is given by:

$$\min_{x} xQx^T, x \in \{0,1\} \tag{1}$$

where x represents the vector of binary decision variables and Q is a square matrix of constants.

To construct the Q matrix, additional steps are required such as: the transformation of inequality constraints into equalities, the removal of bilinear terms and the introduction of quadratic penalty terms, as shown in Figure 1.

The QUBO problem, defined by Eq. (1), is subsequently solved using Quantum Annealing (QA) in a quantum adiabatic environment. QA is a model of quantum computation that utilizes quantum fluctuations to search for ground state solutions of the CO problem that is encoded as a Hamiltonian by applying a transverse field [5]. Furthermore, QA in D-Wave systems utilizes the QUBO formulation for optimization problems, supports over 5,000 physical qubits, and can efficiently explore the solution space by overcoming local minima.

Apart from QA, the D-Wave suite offers different solution approaches that can be utilized to solve QUBO such as the multistart Tabu search (TS) and qboslv (TN) solvers. TS is based on memory-based strategies, and explores regions of the solution space and prevents the recurrence of recent moves. TN partitions the original problem into smaller sub-problems defined by the algorithm which are solved by simulated annealing (SA). Furthermore, the user can define the size of the sub-problem (TSN-x solver), where x represents the maximum number of variables allowed in each subproblem.

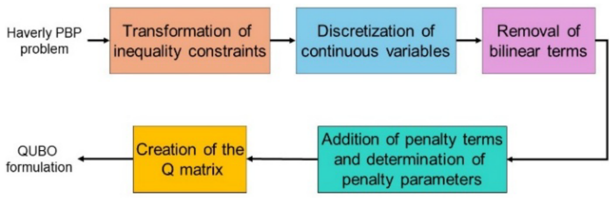
Haverly PBP problem
Transformation of inequality constraints
Discretization of continuous variables
Removal of bilinear terms
Addition of penalty terms and determination of penalty parameters
Creation of the Q matrix
QUBO formulation

Figure 1. Tranformation procedure of the PBP into QUBO

After the QUBO transformation, an embedding procedure onto the QA hardware is required to find the optimal solution, involving mapping the graph described by the QUBO problem as a minor of the graph representing the qubit lattice of the QA system.

To achieve this, Chimera and Pegasus topology unit cells are used to attain connectivity among the qubits which are available in D-Wave systems. These embedding architectures depend on QA hardware design, and parameters such as the largest complete subgraph of an arbitrary problem system (maximal clique size) and the number of nodes in one half of the bipartite cell (shore size).

The benefits of the proposed solution scheme are summarized below:

- The QA framework has the ability to explore the solution space more broadly and not get stuck in a local solution through quantum tunneling.
- Multiple states can be simultaneously explored during the annealing process, reducing the required computational time due to quantum superposition.

3. COMPUTATIONAL RESULTS

In this section, a case study based on Haverly's pooling and blending problem (PBP) [6] is examined using the proposed methodology. This problem is selected due to its non-convex nature and the existence of bilinear terms, which can result in multiple local minima. Furthermore, PBPs are easily scalable in terms of complexity, as the problem definition can be readily expanded to include additional streams, products, and pools.

All the simulations are performed in Python, where the NetworkX package is used for creating and analyzing the Chimera and Pegasus graph objects and the D-Wave's Ocean software for embedding and solving the QUBO problem.

The p-formulation of the PBP including three reagent streams, two product streams, and one pool in which two reagents are blended is given by:

$$\min F_A C_A + F_B C_B + F_{Cx} C_C + F_{Cy} C_C + F_X C_X + F_Y C_Y \tag{2a}$$

$$\text{s.t. } F_{px} + F_{py} - F_A - F_B = 0 \tag{2b}$$

$$F_x - F_{Cx} - F_{px} = 0 \tag{2c}$$

$$F_y - F_{Cy} - F_{py} = 0 \tag{2d}$$

$$S_p F_{px} + \mathrm{S_p} F_{py} - \mathrm{S_A F_A} - \mathrm{S_B F_B} = 0 \tag{2e}$$

$$S_p F_{px} + \mathrm{S_C} F_{Cx} - \mathrm{S_x F_x} \le 0 \tag{2f}$$

$$S_p F_{py} + \mathrm{S_C} F_{Cy} - \mathrm{S_y F_y} \le 0 \tag{2g}$$

$$F_x - 100 \le 0 \tag{2h}$$

$$F_y - 200 \le 0 \tag{2i}$$

where F_A, F_B, F_{Cx}, and F_{Cy} are the feed streams, F_{px} and F_{py} are the pool streams, F_x and F_y denote the product streams, and S_x and S_y represent the product composition. The values of parameters for the examined Haverly

PBP are given in Figure 2.

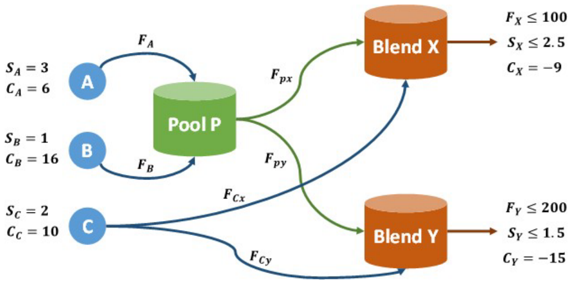

Figure 2. Schematic representation of Harvely PBP

Following the steps illustrated in Figure 1, the reformulated PBP can be written as:

$$\begin{aligned}\min F_A C_A + F_B C_B + F_{Cx} C_C + F_{Cy} C_C + F_x C_x + F_y C_y + \\ P\Big\{\left(F_{px} + F_{py} - F_A - F_B\right)^2 + \left(F_x + F_{Cx} - F_{px}\right)^2 + \left(F_y + F_{Cy} - F_{py}\right)^2 + (T_x + \mathrm{S_C F_{Cx}} - \mathrm{S_x F_x} + \Gamma_3)^2 + \left(T_y + \mathrm{S_C F_{Cy}} - \mathrm{S_y F_y} + \Gamma_4\right)^2 + \left(T_x + T_y - S_A F_A - S_B F_B\right)^2 + (F_x - 100 + \Gamma_1)^2 + \\ \left(F_y - 200 + \Gamma_2\right)^2 + \sum_{i=1}^{n_{fpx}} \sum_{j=1}^{n_{sp}} \left(F_{px,i} S_{p,j} - 2T_{x,ij} F_{px,i} - 2T_{x,ij} S_{p,j} + 3T_{x,ij}\right) + \sum_{i=1}^{n_{fpy}} \sum_{j=1}^{n_{sp}} \left(F_{py,i} S_{p,j} - 2T_{y,ij} F_{py,i} - 2T_{y,ij} S_{p,j} + 3T_{y,ij}\right)\Big\}\end{aligned} \quad (3)$$

where $\Gamma_1 - \Gamma_4$ are the slack variables, P is the penalty parameter, and T_x and T_y are the binary transformation variables for the removal of bilinear terms $S_p F_{px}$ and $S_p F_{py}$, respectively.

In the following, two transformations have been utilized: a verbose transformation, in which the discretized variables are expanded to allow for greater precision, and a succinct transformation, where each variable is constrained to integer values.

In both formulations, the number of binary variables is determined based on the required precision for expressing the continuous variables of the Haverly's PBP. Furthermore, the slack variables introduced to transform the inequality constraints to equalities utilize more qubits compared to the continuous to prevent the imposition of suboptimal values. Based on this protocol, the succinct formulation needed approximately two-thirds fewer binary variables to represent the problem compared to the verbose formulation. This reduction can be attributed to the decrease in the number of transformation variables required to eliminate the bilinear terms present in Eqs. 2e – 2g. Table 1 presents some of the parameters of the resulting succinct and verbose formulations.

Once the QUBO transformation is obtained, the embedding process is employed to insert the reformulated optimization problem into the lattice of qubits that form the quantum processing unit. In this study, the Chimera and Pegasus graphs are used for the embedding. To successfully insert the QUBO to the QA, the value of the maximal clique is required. The maximal cliques for both qubit architectures are obtained using the Bron and Kerboch algorithm [7] and found to be 75 and 30 for the verbose and succinct formulations, respectively, as shown in Table 1.

Table 1: Succinct and verbose QUBO characteristics for the Haverly's PBP

Parameter	Verbose	Succinct
Binaries for continuous variables	66	63
Binaries for slack variables	36	30
Binaries for transformation variables	75	30
Maximum clique size	75	30
Q matrix size	177	123
Non-zero matrix elements	7,765	3,319
Total binary combinations	1.9×10^{53}	1.0×10^{37}

4. RESULTS AND DISCUSSION

All the simulations for the reformulation of the PBP are performed in Python using the D-Wave Ocean toolbox [8], where SA, TS, TN and TSN-x solvers as well as the Pegasus and Chimera topologies are used to solve the succinct and the verbose formulations. As a reference case, the QUBO problem is solved using the Multistart and GlobalSearch solvers available in MATLAB's Global Optimization (GO) toolbox. After the execution of each solver, a sample set is returned and the set with the most occurrences among the collected data is returned as the best solution.

4.1 PBP embedding procedure

In this subsection, the performance of the different embedding architectures is examined and compared. Apart form the calculation of the maximum clique value, there are other features that affect the performance of the embedding procedure, such as the number of units, shore size, and the unit's height and width.

In the Chimera graph architecture, two different cases are examined. In the first case, the shore size is fixed to the value of 4 and the height and the width of the graph varied in integer increments from 16 to 26 with the structure remaining symmetric. In the second case, the graph's height and width kept constant to 16 and the shore size varied in integer increments from 4 to 10. In both cases, the probability and the time required to embed the problem system into the Chimera graph are obtained.

It can be observed that the verbose formulation requires a high number of qubits compared to the succinct formulation for a successful embedding, as shown in Figure 3. Furthermore, the verbose architecture failed to embed on any of the Chimera graphs with shore size equal to 4 and varying graph's height and width.

Figure 4 presents the required wall clock for successful embedding. The succinct architecture exhibits rapid convergence in the range of 20-30 seconds with a standard deviation of 15-30 seconds. On the other hand, the verbose embedding requires higher average embedding times and standard deviations.

It can be concluded that when the embedding is successful, the embedding time will converge, with the rate of convergence being influenced by the system's connectivity.

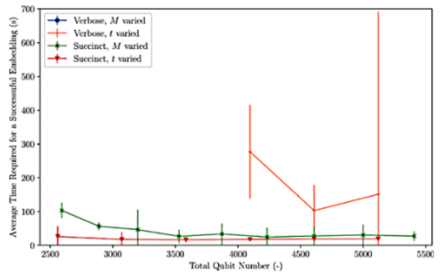

Figure 3. Required time for successful embedding in a Chimera graph as a function of qubit number

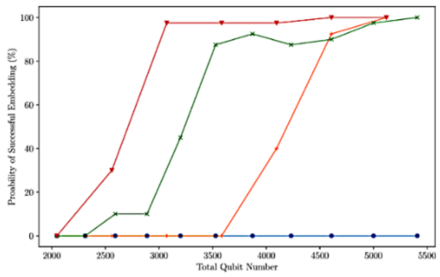

Figure 4. Propability for the problem system in a Chimera graph as a funtion of qubit number

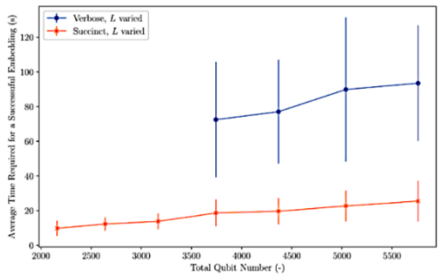

Figure 5. Required time for successful embedding in a Pegasus graph as a function of qubit number

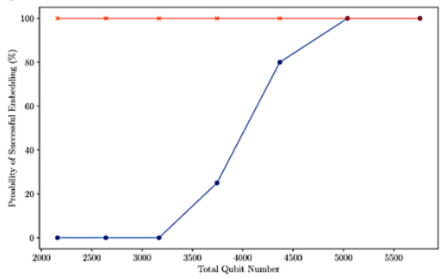

Figure 6. Propability for the problem system in a Pegasus graph as a funtion of qubit number

The same procedure is used for the Pegasus configuration where the graph's size parameter is varied in integer increments from 10 to 16. It can be seen that the succinct formulation is successfully embedded for all the sizes, whereas the verbose formulation only when the number of qubits is 3,700 and above, as shown in Figure 5.

Contrary to the behaviour observed for the Chimera embedding, Figure 6 illustrates that the required wall clock time as well as the standard deviation for the embedding step increases with the increase in the number of qubits.

4.2 Penalty multiplier value selection

As mentioned in Section 2, a penalty term is introduced during the QUBO transformation procedure to enforce constrained behaviour without the use of constraints. The selection of the appropriate P value is a non-trivial task because there is a trade-off between enforcing constrained behaviour that usually requires large values and retaining the objective function information [9].

For this task, the optimal value of the penalty multiplier was found by solving the verbose and succinct formulations of the problem at different values ranging from 75% to 150% of the optimal objective function value, $\hat{f}^*$ (which is equal to -400), in 5% increments. Each system was solved using 40 calls and the obtained values were processed to yield the unweighted objective function value, given by:

$$\bar{f} = \frac{(\hat{f}-c)}{k} \tag{4}$$

where c is a constant representing the total sum of the constant cross terms from the quadratic penalties and k is a constant ranging from 0.75 to 1.5.

In Figure 7, it can be observed that the succinct formulation of the QUBO problem is solved more easily than the verbose formulation. Furthermore, this effect is most clear in the TN method, while for the TSN-x method is the least evident.

As far as the constraint violation is concerned, the verbose formulation demonstrates lower values compared to the succinct, as shown in Figure 8. Furthermore, SA and TS solvers exhibit the highest and the lowest value for the squared constraint violation, respectively.

It should be noted that none of the solvers returned negative solutions and consequently corresponding all to a setup that would be unprofitable or physically impossible. For each solver, the optimal value of the penalty parameter was obtained by calculating a composite rank that is equally weighted in terms of the objective function value and constraint violations. The obtained results of the ranking method are shown in Table 2.

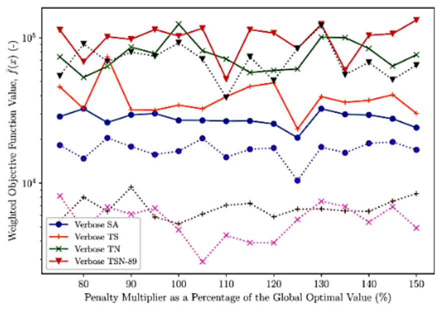

Figure 7. Avarage unweighted objective function value

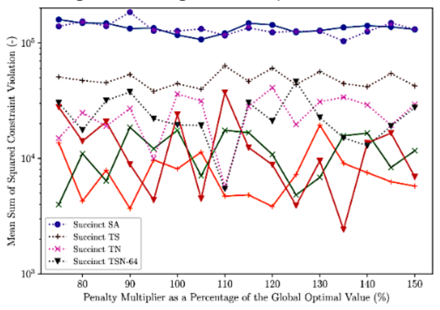

Figure 8. Mean sum of squared constraint violation

Table 2: Penalty multiplier values P as a % of the objective function values that yielded the lowest composite score

Formulation	SA	TS	TN	TSN-x
Verbose	125%	90%	125%	135%
Succinct	110%	95%	110%	110%

4.3 QUBO solution

Based on the penalty multiplier values determined in Section 4.2, both the succinct and verbose formulations were created and each problem system is solved using all the aforementioned solvers, using a total of 500 calls and 10,000 initial points. For the GO solvers, the findings indicate that the inclusion of penalty terms enhances the complexity of the resulting system due to the bilinear transformation, thereby making it more challenging to solve.

Tables 3 and 4 present the statistics for the verbose and the succinct of the PBP formulation. It can be seen that the succinct formulation delivers better solutions in terms of the unweighted objective function across all solvers. In particular, this effect is more visible for the TN solver. Furthermore, SA exhibits the best performance for the verbose and the second-best for the succinct formulation among the examined solvers, producing low standard deviations for the unweighted objective function value.

In contrast, the verbose formulation shows better performance in the mean sum of squared violations, except for the case of the TN solver. It can be observed that the succinct formulation exhibits lower performance compared to the verbose at the recovered values of the original objective function.

Based on these findings, the succinct formulation runs into fewer local minima. Therefore, this behaviour can be explained as a result of minimizing toward better underlying solutions but being unable to reduce the error in those solutions.

The TN solver demonstrates better performance in constraint violation and unweighted objective function value compared to the SA solver, as shown in Tables 3

Table 3: Summary of results for the verbose formulation of the PBP

Parameter	SA	TS	TN	TSN-89	TSN-45	TSN-23
Average solving time (ms)	214	596	2,087	3,039	1,875	782
Mean of $\bar{f}$	6,290	14,600	24,800	30,300	23,800	36,000
Standard deviation of $\bar{f}$	972	17,100	27,000	31,700	27,100	31,300
Mean of c	76,240	44,990	40,961	50,174	41,436	51,999
Standard deviation of c	81,999	74,329	73,619	79,924	73,528	73,324
Mean of f	-9.62	-77.85	-72.27	-76.98	-79.59	-76.35
Standard deviation of f	238	205	254	263	243	271

Table 4: Summary of results for the succinct formulation of the PBP

Parameter	SA	TS	TN	TSN-62	TSN-31	TSN-16
Average solving time (ms)	93	587	1,514	1,480	960	481
Mean of $\bar{f}$	3,340	3,900	1,620	11,400	5,560	30,700
Standard deviation of $\bar{f}$	732	1,360	625	16,700	8,690	32,900
Mean of c	81,282	76,608	28,499	64,408	52,443	59,919
Standard deviation of c	88,689	89,682	54,898	90,968	82,191	87,127
Mean of f	-25.02	54.75	-46.43	-34.15	-38.06	-60.16
Standard deviation of f	241	252	168	158	217	294

and 4. However, the TN solver requires approximately 10 – 15 times more computational time. In the case of the TSN solver, its performance is notably sensitive to the size of the sub-problems, highlighting the pivotal role of subproblem size in achieving optimal solutions.

Overall, the SA solver shows strong performance in terms of the unweighted objective function value and computational time. Therefore, the SA solver is the most promising candidate which is capable of producing good quality results and is suitable for solving QUBO formulation problems successfully.

5. CONCLUSIONS

This contribution introduces a quantum approach to formulating and solving CO problems using a QA solution framework. In this work, the procedure for the CO reformulation is showcased as well as embedding and solution of the QUBO reformulation of the Haverly PBP on a quantum machine.

Two QUBO problem formulations, verbose and succinct, are selected and compared. The results revealed that the succinct version of the optimization problem requires smaller value for penalty multipliers and produced better results compared to the verbose formulation.

The numerical experiments have shown that the verbose formulation can be embedded even on very large Chimera graphs. Contrary, the Pegasus architecture is able to successfully handle both succinct and verbose formulations, requiring an average of 25 and 90 seconds, respectively.

The SA solver showed the best performance among the examined solvers in the D-Wave platform and the deterministic GO solvers, although it required high computational time. Furthermore, the results of the TSN-x solver highly depend on the size of subproblem.

This methodology holds the potential for solving efficiently CO problems with the improvement capabilities of the quantum computers in terms of computational power and the availability of the maximum qubits. This approach can be extended to solve chemical engineering applications such as process synthesis and design and production planning and scheduling problems.

As future work, alternative ways for handling bilinear and higher order terms and the transformation of the continuous and integer variables into binaries should be investigated to obtain good quality results in lower computational time.

Additionally, the structure of the optimization problem plays an important role in the solution procedure. Specifically, the embedding algorithm, the sub-problem size and the penalty multiplier values can be optimized, providing the ability of QA systems to offer scaling advantages over traditional heuristics when applied to larger PBPs.

REFERENCES

1. Zhang C, Wu Y, Ma Y, Song W, Le Z, Cao Z, Zhang J. A review on learning to solve combinatorial optimisation problems in manufacturing. IET Collaborative Intelligent Manufacturing, 5:e12072 (2023) https://doi.org/10.1049/cim2.12072
2. Peres F, Castelli M. Combinatorial optimization problems and metaheuristics: Review, challenges, design, and development. Applied Sciences, 11:6449 (2021) https://doi.org/10.3390/app11146449
3. Blekos K, Brand D, Ceschini A, Chou C-H, Li R-H, Pandya K, Summer A. A review on quantum approximate optimization algorithm and its variants. Physics Reports, 1068:1-66 (2024) https://doi.org/10.1016/j.physrep.2024.03.002
4. Leichtle D, Music L, Kashefi E, Olivier H. Verifying BQP computations on noisy devices with minimal overhead. PRX Quantum, 2:040302 (2021) https://doi.org/10.1103/PRXQuantum.2.040302
5. Yarkoni S, Raponi E, Bäck T, Schmitt S. Quantum annealing for industry applications: Introduction and review. Reports on Progress in Physics, 85:104001 (2022) https://doi.org/10.1088/1361-6633/ac8c54
6. Lotero I., Trespalacios F., Grossmann I. E., Papageorgiou D. J., Cheon M-S. An MILP-MINLP decomposition method for the global optimization of a source based model of the multiperiod blending problem. Computers & Chemical Engineering, 87:13-35 (2016) https://doi.org/10.1016/j.compchemeng.2015.12.017
7. Gupta S. K., Singh D. H., Choudhary J. A review of clique-based overlapping community detection algorithms. Knowledge and Information Systems, 64:2023-2058 (2022) https://doi.org/10.1007/s10115-022-01704-6
8. Ocean Developer Tools, https://docs.ocean.dwavesys.com
9. Ju J., Liu Q. Convergence properties of a class of exact penalty methods for semi-infinite optimization problems. Mathematical Methods of Operations Research, 2020, 91:383-403 https://doi.org/10.1007/s00186-019-00693-7