Hybrid Quantum-Classical Heuristic for the Bin Packing Problem

Mikel Garcia de Andoin[1,2,3], Eneko Osaba[2], Izaskun Oregi[2], Esther Villar-Rodriguez[2], Mikel Sanz[1,3,4,5]

[1]Department of Physical Chemistry, University of the Basque Country UPV/EHU, Apartado 644, 48940 Leioa, Spain
[2]TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Spain
[3]EHU Quantum Center, University of the Basque Country UPV/EHU, 48940 Leioa, Spain
[4]IKERBASQUE, Basque Foundation for Science, Plaza Euskadi 5, 48009 Bilbao, Spain
[5]BCAM - Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Spain
mikel.garciadeandoin@ehu.eus,eneko.osaba@tecnalia.com

ABSTRACT

Optimization problems is one of the most challenging applications of quantum computers, as well as one of the most relevants. As a consequence, it has attracted huge efforts to obtain a speedup over classical algorithms using quantum resources. Up to now, many problems of different nature have been addressed through the perspective of this revolutionary computation paradigm, but there are still many open questions. In this work, a hybrid classical-quantum approach is presented for dealing with the one-dimensional Bin Packing Problem (1dBPP). The algorithm comprises two modules, each one designed for being executed in different computational ecosystems. First, a quantum subroutine seeks a set of feasible bin configurations of the problem at hand. Secondly, a classical computation subroutine builds complete solutions to the problem from the subsets given by the quantum subroutine. Being a hybrid solver, we have called our method H-BPP. To test our algorithm, we have built 18 different 1dBPP instances as a benchmarking set, in which we analyse the fitness, the number of solutions and the performance of the QC subroutine. Based on these figures of merit we verify that H-BPP is a valid technique to address the 1dBPP.

CCS CONCEPTS

• **Mathematics of computing** → **Combinatorial optimization**;
• **Theory of computation** → *Quantum computation theory*.

KEYWORDS

Quantum Computing; Bin Packing Problem; Combinatorial Optimization; Hybrid Quantum Algorithm

1 INTRODUCTION

Computational optimization is a highly studied research topic within the wider Artificial Intelligence field. Its contrasted applicability is one of the main reasons of the success of this knowledge area, which is resorted to address a myriad of real-world oriented tasks.

Usually, the efficient tackling of optimization problems involves the need of significant computational resources, making impractical the adoption of a brute-force approach. For this reason, the formulation of time-efficient solving strategies has emerged as a hot topic, leading to the design of a plethora of heterogeneous techniques during the last decades.

At the time this paper is written, the vast majority of optimization algorithms have been conceived for being executed on classical computers. In this context, Quantum Computing (QC, [66]) has recently emerged as a promising paradigm for facing optimization problems. In a nutshell, QC provides a revolutionary approach for dealing with complex problems with a possible significant advantage[3]. Today, QC is the pivotal point of a growing amount of experimental and theoretical scientific studies [35, 53, 62].

Despite the potential demonstrated by QC, this research field is still at its dawn, and the available quantum processors present some limitations in terms of performance and capability [4, 27]. Issues such as the reduced size of the quantum processors, the inaccurate control of the quantum resources, or the limitations in material science should be faced by researchers and engineers on their effort towards fault-tolerant quantum-computing. As a result of these handicaps, two kind of solvers prevail in the current literature: i) *purely quantum* approaches, whose objective is to face a problem employing only QC resources, and *quantum-classical hybrid* techniques, which are conceived to enhance applicability within the short and mid-term QC limitations.

In this context, our objective with this paper is to take a step forward in QC by proposing a *quantum-classical hybrid* algorithm for addressing one of the best-known combinatorial optimization problems: the Bin Packing Problem (BPP, [32]). Although it is a canonical use case, the BPP often emerges in a wide variety of industrial problems. To our best knowledge, this problem has not yet been faced from a QC perspective. Thus, the present work represents a step further over the current QC literature, presenting the following contributions:

- We introduce a *quantum-classical hybrid* algorithm for solving the well-known one-dimensional BPP (1dBPP,[52]). The algorithm is composed of two modules, each one designed for being executed in different computational ecosystems.

First, a QC-based module is employed for finding the set of feasible bin configurations of the problem at hand. In other words, this module seeks as many feasible bins as possible, considering the constraints of the problem (i.e. the weight of the items and the capacity of the bins available). The second module, which is implemented to be run in classical computers, takes as input the list of feasible subsets provided by the QC-based module, and generates complete solutions to the problem finding the optimal combination of subsets. As it is a hybrid solver, we have called our method H-BPP. To the best of our knowledge, it is the first time that a BPP is solved using such strategy.

- As mentioned, the proposed method has been developed for facing the 1dBPP. Despite related problems, such as the knapsack problem [61] or the subset-sum problem [16] have been considered in the literature, no variant of the BPP has been studied from a QC perspective up to now. Therefore, taking into account the scientific and business interest that BPP-related problems still arise in the community and in the industry [10, 11], it is relevant to further research in this direction. For testing the performance and accuracy of our method, we have conducted a preliminary but extensive experimentation over 18 synthetically-generated heterogeneous instances of the 1dBPP, with sizes ranging from 10 to 12 items. Results obtained by our algorithm are compared with the ones obtained by a brute-force algorithm.

The rest of this manuscript is structured as follows: we present in Section 2 a brief introduction on QC and BPP. Afterwards, the problem addressed in this paper, the 1dBPP, is described in Section 3. In Section 4, we delve into the main characteristics of our hybrid algorithm. Experimental setup and results are discussed in Section 5. Finally, in Section 6 we summarize our conclusions and outline future research lines.

2 BACKGROUND

This section briefly introduces the two main pillars of this paper: Quantum Computing, in Section 2.1; and the Bin Packing Problem, in Section 2.2.

2.1 Introduction to Quantum Computing

Quantum computing takes advantage of quantum resources to solve computational problems. Indeed, exploiting entanglement and superposition properties, some quantum algorithms have shown potential to speed up the convergence towards a solution for certain problems. Two representative algorithms that show quantum advantage are Grover's algorithm for the unstructured searching problem [33] and Shor's algorithm for the integer factorization [64]. Even though in these two cases a speedup is achieved, not every problem can benefit from employing quantum resources [1], especially in the current Near-Intermediate Scale Quantum era [43, 67] (NISQ, noisy quantum hardware with a size in the order of 10-100 qubits [59]). However, quantum algorithms might have a speedup compared with classical approximate algorithms for particularly hard problems (i.e. problems in the approxNP-hard set), such as the Max-E3-SAT problem [36], [73].

Within the framework of quantum computing, there are three main different paradigms. Firstly, digital quantum computing (DQC) [23], which works with a sequence of ideal operations (or gates) taken from an universal set. For implementing algorithms in this paradigms, we have a pletora of gate-based quantum computers (IBM[1], Google[2], Rigetti[3], ColdQuanta[4]), which allows users to run programs written as the application of seccessive gates. Then, analog quantum computing (AQC) is a less flexible but much more robust approach [37]. In AQC, a controllable quantum system is manipulated to mimic the system of interest. This approach also includes quantum annealing and simulators, which can be implemenmted in any system which provides a way to adiabatically modify the system Hamiltonian (D-Wave[5], Xanadu[6]). finally, an universal quantum computing paradigm has been proposed, namely digital-analog (DAQC [57]). This approach combines the robustness of AQC with the flexibility of DQC. The algorithms are implemented by alternating the application of (digital) single-qubit gates with analog entangling blocks. Running a circuit using DAQC requires a hardware in which one has access to fast single-qubit gates and an interaction Hamiltonian, as it has been implemented in different platforms.

Encouraged by early results, the quantum-computing and computer science-communities have conducted a huge effort for achieving quantum advantage. As a prominent example of this research, an information processing task performed by a QC was proven unaffordable for any classical computer. Even though this has been attained for certain artificial problems [9, 72], achieving this for a useful application is still an open question. Approximate optimization problems are considered a suitable target for this goal. On this regard, quantum annealing takes advantage of a purely quantum mechanism, the adiabatic theorem [18, 38], to find the ground state of a Hamiltonian which encodes the solution to the problem [25, 34]. A family of near-term quantum algorithms that have shown a reasonable success are variational quantum algorithms (VQAs) [20], with variational quantum eigensolvers (VQE) and quantum approximate optimization algorithm (QAOA) [26] as the most prominent examples. These algorithms relay on the classical optimization of control parameters with the objective of obtaining the best possible solution given a certain amount of resources. On another vein, hybrid quantum algorithms employs quantum algorithms to speedup certain operations, while subrogating some tasks to classical routines [24, 49]. When evaluating quantum algorithms, it should be noted that the probability to obtain a solution is unlikely to reach exactly 1. On top of this, the problems of efficiently encoding information into a quantum system and retrieving it by means of measurements are generally open problems.

2.2 BPP state of the art

A paradigmatic example of a NP-Hard problem is the BPP, which is a classical combinatorial optimization problem whose objective is to pack a finite set of items into a group of available bins. More

---

[1]https://www.ibm.com/quantum-computing/

[2]https://www.quantumai.google/

[3]https://www.rigetti.com/

[4]https://www.coldquanta.com/

[5]https://www.dwavesys.com/

[6]https://www.xanadu.ai/

precisely, the goal is to minimize the total number of bins used not exceeding the fixed maximum capacity of each bin considered in the solution. The packaging of items or packages in different containers or bins is a daily and crucial task in the field of production and logistics. For this reason, multiple packaging problems can be formulated depending on the size of the items to be packed, as well as the size and capacity of the containers. These types of issues have been widely discussed in the literature for several decades, giving rise to remarkable survey and review papers [44, 46, 48].

The 1dBPP is considered to be the simplest packaging problem. Anyway, despite its simplicity, it is applied to a wide variety of real-world problems [8, 70]. Furthermore, it is frequently used as a benchmarking problem [28, 55]. Recent works focused on 1dBPP led to the proposition of novel methods for its resolutions, such as the cooperative parallel grouping genetic algorithm introduced in [40], the branch-and-price-and-cut technique proposed in [71], or the adaptive fitness-dependent optimizer developed in [2].

Beyond its basic formulation, several variants of the 1dBPP have been proposed in order to give an answer to some specific constraints. Some examples of these variants can be found in [39], in which a maximum number of items per bin is fixed; or in [21], in which items should be placed in bins according to some concrete time intervals. Further variants of the basic 1dBPP can be found in [22]. In this present research, the canonical version of the 1dBPP is considered. This variant still gathers lot of attention from the community, as can be seen in recent surveys such as [51] and [52].

Additionally, based on the original formulation of the 1dBPP, many variants have been recently proposed in recent literature to address real-world situations. On this regard, two are the most frequently referred variants of the BPP. The first one is the two dimensional BPP (2dBPP, [45]), in which each item is described by two different parameters: height and width. There are many works in the recent literature that revolve around this problem and its resolution. In [41], for example, the authors address this problem with a genetic algorithm, adding a search mechanism called Crow Search to increase the exploration capacity of the algorithm. In the work [47] a hybrid approach applied to the same problem is presented. Additional examples can be found in articles like [15] or [63]. A second variant is the three-dimensional BPP (3dBPP, [50]), in which each packet has three dimensions: height, width, and depth. Consequently, the containers in which the packages are stored have three dimensions as well. This variant has also been widely investigated by the community [54, 60].

Furthermore, variations with a variable number of additional restrictions or constraints arising from real-world use cases have been studied, usually leading to an increment of the problem complexity. Some examples are *i*) the minimum clearance of the container [6], *ii*) load balancing inside the container [69], *iii*) expiration dates of rapidly perishable products [58], *iv*) transport restrictions [56], or *v*) possible packet fragmentation [14].

In addition to these restrictions, the problem has been treated from different computational approaches, highlighting the single-objective and multi-objective variants [65].

To sum up, the research around BPP-related problems is still vibrant today. Even the most basic variant of the BPP, the 1dBPP, still gathers a significant attention from the community. However, as far as we know, there is no work in the literature that deals

with this problem from the QC perspective. On the contrary, there are few publications mentioning the BPP and quantum algorithms, but most of them discuss it from quantum-inspired classical algorithms perspective [29, 42, 68]. Furthermore, slightly related with the 1dBPP, there are quantum algorithms solving the subset-sum problem [5, 13, 17]. However, these algorithms take advantage of some-problem specific properties which can not straightforwardly be translated to our scheme.

For this reason, this paper implies a remarkable step forward in the literature, tackling a BPP problem through a hybrid classical-quantum algorithm.

3 THE ONE DIMENSIONAL BIN PACKING PROBLEM

As mentioned beforehand, this paper is focused on solving a BPP variant known as 1dBPP. The problem can be formally defined as follows [51]: considering a maximum amount of n bins $\{B_1, B_2, \ldots, B_n\}$ of equal non-negative capacity C, and a set of n items $\mathcal{W} = \{w_1, w_2, \ldots, w_n\}$, where w_i is a positive value $0 < w_i \leq C$ representing the weight of item i, the objective is to find the smallest amount of bins $b \leq n$ such that partition $L = B_1 \cup B_2 \ldots \cup B_b$ fits the whole $\mathcal{W}$. Furthermore, the sum of the item weights in each bin must not exceed C.

Regarding the codification used for representing a solution to the problem, the well-known binary encoding has been considered. In this way, the 1bBPP can be formulated in the following way:

$$\min b \tag{1}$$

Subject to:

$$\sum_{i=1}^{n} w_i x_i^{(j)} \leq C, \ \forall j \in \{1, \ldots, b\}, \tag{2}$$

$$\sum_{j=1}^{b} x_i^{(j)} = 1, \ \forall i \in \{1, \ldots, n\}, \tag{3}$$

$$x_i^{(j)} \in \{0, 1\}, \ \forall i \in \{1, \ldots, n\}, \ \forall j \in \{1, \ldots, b\}, \tag{4}$$

where b is the number of employed bins, such that the set of bins is $\{B_1, \ldots, B_b\}$, and $x_i^{(j)} = 1$ if item i is in bin j.

This strategy has the advantage of being fully compliant with digital QC devices, designed for working with binary variables. On this basis, one solution to the 1dBPP is provided by means of b binary arrays of size n ($L = \{x_1^{(1)}, \ldots, x_n^{(1)}\}, \{x_1^{(2)}, \ldots, x_n^{(2)}\}, \ldots, \{x_1^{(b)}, \ldots, x_n^{(b)}\}$). Let us assume a possible instance composed by 10 items ($n = 10$). By using a binary representation, one feasible solution comprising two bins ($b = 2$) might be:

$$B_1 = [0, 1, 1, 0, 0, 0, 1, 1, 0, 0]$$

$$B_2 = [1, 0, 0, 1, 1, 1, 0, 0, 1, 1]$$

meaning that four items (w_2, w_3, w_7 and w_8) are placed in the first bin, while the remaining six assigned to the second one, hence being the intersection $\emptyset$. This solution can also be encoded as ($L = \{x_1^{(1)} = 0, x_2^{(1)} = 1, x_3^{(1)} = 1, x_4^{(1)} = 0, x_5^{(1)} = 0, x_6^{(1)} = 0, x_7^{(1)} = 1, x_8^{(1)} = 1, x_9^{(1)} = 0, x_{10}^{(1)} = 0\}, \{x_1^{(2)} = 1, x_2^{(2)} = 0, x_3^{(2)} = 0, x_4^{(2)} = $

$1, x_5^{(2)} = 1, x_6^{(2)} = 1, x_7^{(2)} = 0, x_8^{(2)} = 0, x_9^{(2)} = 1, x_{10}^{(2)} = 1\})$. An additional, yet worse, solution for this instance could be:

$$B_1 = [0, 1, 1, 0, 0, 0, 1, 1, 0, 0]$$

$$B_2 = [1, 0, 0, 0, 0, 1, 0, 0, 0, 1]$$

$$B_3 = [0, 0, 0, 1, 1, 0, 0, 0, 1, 0]$$

where three bins are used for storing the complete set of items. As can be deduced, for these solutions to be feasible, each item $w_i \in \mathcal{W}$ should be placed in one, and only one, bin B_j, and the capacity C should not be exceeded by any B_j.

As the BPP is a combinatorial optimization problem, one can find a quadratic unconstrained binary optimization (QUBO) formulation for solving it. With this QUBO formulation, it is immediate to map it to an Ising Hamiltonian whose ground state corresponds to the solution of the original problem. However, the QUBO formulation comes with a cost in terms of employing extra bits or qubits to express express the problem. In particular, for the 1dBPP we would need n^2 to encode the assignment of the packages to the container. To encode the first constraint (Eq. 2) we can introduce slack variables [19, sec 4.1.3]. Employing a binary codification of the slack variables, we would need to introduce $\log_2(C)$ ancillary qubits per container. Since the second constraint (Eq. 3) is given as an equality, so it can be implemented without additional qubits. By measuring the quantum state in the computational basis, the third constraint would be fulfilled by construction. With all these ingredients, we could run a quantum annealing algorithm to solve the full 1dBPP problem, but at the expense of employing $n^2 + n \log_2(C)$ qubits, which is impractical within the limitations in size of NISQ systems. Thus, we have avoided to address the full problem with a quantum annealing algorithm.

4 PROPOSED QUANTUM BASED HEURISTIC FOR SOLVING THE BPP

In this section, we propose a hybrid quantum-classical algorithm for solving 1dBPP, that is, the H-BPP. The algorithm is divided into two main subroutines. In the first one, a quantum annealing algorithm is executed multiple times to sample feasible configurations of items in a single bin. When this first subroutine ends, a classical subroutine is run, whose objective is to optimally combine the previous list of feasible subsets, considering that each item is assigned to a feasible bin. As a result of this second subroutine, the algorithm returns the best solutions found, this is, the solutions requiring the minimum amount of containers to accommodate all items. A diagram with the overview of H-BPP is shown in Fig. 1. In the rest of the section, we explain these two subroutines in detail.

4.1 Quantum annealing algorithm for the subset sampling subroutine

The objective of the subset sampling subroutine is to obtain a set of feasible subsets, which effectively reduces the search space for the classical solution construction algorithm. On this regard, we define a feasible subset as a set of items $B_j \in \{1, \ldots, n\}$ such that they fulfill the constraint from Equation 2 for a single container. The process of sampling feasible subsets will be performed with a quantum annealing algorithm.
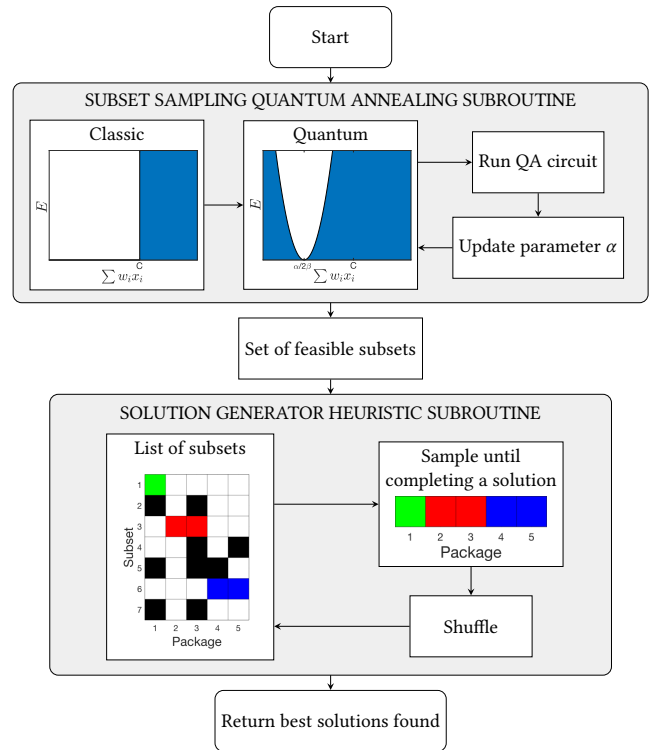
Figure 1: Diagram of the full H-BPP algorithm. The algorithm is divided into two subroutines. The first one is a quantum annealing algorithm to obtain a set of feasible configuration for single containers. The second one takes the set provided by the quantum subroutine to generate possible solutions. The solution yielded by the algorithm is a list with the best results found.

In a nutshell, quantum annealing is a heuristic quantum algorithm for finding the ground state of a Hamiltonian. By means of an adiabatic evolution, the quantum state evolves from an easily preparable ground state of an initial Hamiltonian H_0 to the ground state of the problem Hamiltonian H_P, which encodes the information of the optimization problem. In general, the time-dependent Hamiltonian is given by

$$H(t) = (1 - \Lambda(t))H_0 + \Lambda(t)H_P, \tag{5}$$

where $\Lambda(t)$ is the mixing function with the boundary conditions $\Lambda(0) = 0$, $\Lambda(\tau) = 1$, and τ is the annealing time. For simplicity, and without loss of generality, we choose this function to be a linear function $\Lambda(t) = t$, although it can be designed to verify that the adiabaticity condition is fulfilled [7].

Furthermore, to solve the subset sampling problem, ideally we seek the ground states of a stepwise Hamiltonian

$$H_{\text{ideal}} = \begin{cases} \infty & \text{if } \sum_i w_i x_i > C, \\ 0 & \text{otherwise}, \end{cases} \tag{6}$$

where we have slightly modified the notation used in the previous section. Now $x_i = 1$ if the i-th item belongs to the subset, and $x_i = 0$ otherwise. To encode the classical information into the quantum

system, we do a one-to-one assignment of each binary variable to a qubit, where we represent the state in the computational basis $x_i = 1 \equiv |1\rangle_i$ and $x_i = 0 \equiv |0\rangle_i$. This way, a classical bit x_i has a one-to-one map to a local Hamiltonian term of the form $(\mathbb{1} - \sigma_i^z)/2$, where σ_i^μ is the Pauli μ operator acting on qubit i.

Additionally, we cannot generate stepwise Hamiltonians in a real system, so we have to find an implementable model for Equation 6. Instead of employing a single Hamiltonian, we divide the search of feasible subsets. To achieve this, we divide the problem into a number of quadratic Hamiltonians in which the (possibly degenerated) ground state corresponds to configurations for which $\sum_i w_i x_i = C - \alpha/2\beta$, i.e. the ground states corresponds to subsets with a total weight equal or lower than the maximum capacity C. For this, we employ the following quadratic Hamiltonian

$$
\begin{aligned}
H_P &= \alpha \left(\sum_i w_i x_i - C \right) + \beta \left(\sum_i w_i x_i - C \right)^2 \\
&\equiv -\sum_i w_i \left(\frac{\alpha}{2} + \beta \epsilon_w \right) \sigma_i^z + \sum_{i<j} \frac{\beta w_i w_j}{2} \sigma_i^z \sigma_j^z,
\end{aligned}
\tag{7}
$$

where $\epsilon_w \equiv \sum_i w_i/2 - C$. H_P has two-body interactions between all pairs of qubits, i.e. it is an all-to-all (ATA) Hamiltonian. The two parameters α, β allow us to tweak the sum of the weights of the packages, and the energy penalty added to the states not fulfilling the condition. This way, we can sample the allowed configurations by setting $\alpha = 2\beta(C - k\Delta_w)$, where Δ_w is the minimum weight difference between two different configurations and $k = 1, ..., C/\Delta_w$. Since we cannot know how many runs for the different values of α are required for sampling all possible configurations, we set a total number of runs. Then, we will repeat the annealing algorithm for each value of α and save all the results obtained as shown in Algorithm 1. A visual representation of the subroutine sampling strategy is displayed in Figure 2.

Algorithm 1: Quantum annealing subroutine for the subset sampling

Data: w_i, C, β, #runs
1   **for** $k = 1,...,C/\Delta_w$ **do**
2   Update Hamiltonian parameter, $\alpha = 2\beta(C - k\Delta_w)$;
3   **for** $i = 1,...,\#runs\ C/\Delta_w$ **do**
4   Run the quantum annealing algorithm, B_j;
5   Add the result to the list of subsets, $\mathcal{F} \leftarrow B_j$;

Result: The list of all sampled subsets. $\mathcal{F}$

Once implemented, we have to run the quantum annealing algorithm. For this, we have two distinct alternatives. On the one hand, we can use a quantum annealer in which we have the resources to implement the evolution under our problem Hamiltonian. On the other hand, we can simulate the digitized evolution, i.e. approximating the time dependant evolution with n_T time-independent evolution steps [12], with a DQC or DAQC implementation. With the currently available quantum hardware capabilities and qubit connectivity topology, we can not directly simulate the evolution under H_P. Thus, we have to resort to methods that simulate an

ATA Hamiltonian with limited resources, for example, using the decomposition shown in [30].

4.2 Solution Generator Heuristic

This section briefly describes the heuristic procedure used to solve the 1dBPP by using as input the set of feasible subsets $\mathcal{F}$ computed in the previous subroutine. Specifically, the proposed scheme aims at seeking, among the elements in $\mathcal{F}$, an optimal combination that packages all items in $\mathcal{W}$. In this case, the search strategy has the objective of finding the configuration with the smallest number of bins. It is important to highlight that this algorithm has been designed to be run in classical devices.

Algorithm 2: Classical heuristic for finding the best combination of bins

Data: $\mathcal{F}, \mathcal{W}, \texttt{MaxIter}$
1   Initialize solution set, $\mathcal{S} = \emptyset$;
2   **for** $c = 1, \dots, \texttt{MaxIter}$ **do**
3   Initialize number of bins, $b = 0$;
4   Initialize the set of assigned packages, $\mathcal{A} = \emptyset$;
5   Randomly shuffle elements in $\mathcal{F}$;
6   **for** $j = 1, \dots, |\mathcal{F}|$ **do**
7   Select subset j from $\mathcal{F}$, $B_j = \mathcal{F}[j]$;
8   **if** *all elements in B_j are unassigned,* $(B_j \cap \mathcal{A}) = \emptyset$
9   Include items in B_j in $\mathcal{A}$, $\mathcal{A} = \mathcal{A} + B_j$;
10   Increment the number of bins, $b = b + 1$;
11   **if** *all packages are assigned,* $\mathcal{W} = \mathcal{A}$
12   Add solution to $\mathcal{S} \leftarrow \mathcal{S} \cup \{b\}$;
13   Break for loop;

Result: $\min \mathcal{S}$

The pseudocode of the proposed heuristic is given in Algorithm 2. As aforementioned the procedure gets the set of feasible subsets $\mathcal{F}$ and creates a solution (lines 6-11) by looking for bins (lines 7-9) that might package all the items in $\mathcal{W}$ (line 10). Note that this search is made sequentially in $\mathcal{F}$, so this strategy cannot guarantee the optimal solution if it is computed just once. To improve the search, the algorithm repeats this procedure MaxItems times (line 2), saving the computed solution in a temporary variable $\mathcal{S}$. Finally, the procedure outputs the solution with the smallest number of bins stored in $\mathcal{S}$.

5 EXPERIMENTAL SETUP, RESULTS AND DISCUSSION

This section is devoted to showcasing the whole experimentation conducted in this study. To this end, this section have been divided into two different subsections, the first aiming at describing the experimental setup (Section 5.1), and the second focused on showing and discussing the main results obtained by our H-BPP (Section 5.2).

5.1 Experimental Setup

As pointed in the introduction, the benchmark comprises the resolution of 18 different synthetic instances. With the main intention
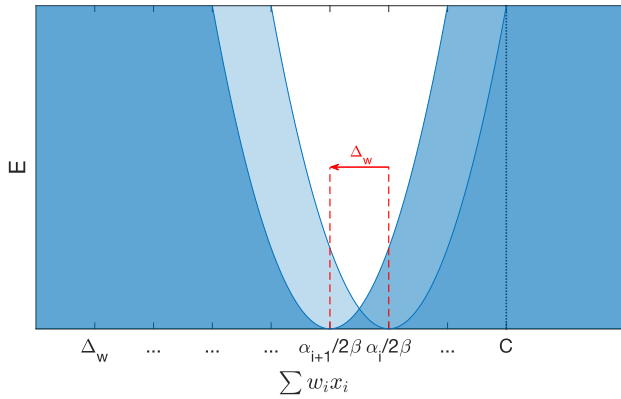
Figure 2: Representation of the sampling strategy used in the 1dBPP problem. The x axis ticks shows the ideal weights of the subsets we are sampling in each step, while the y axis shows the energy of the eigenvalues. If one implements the strategy for not measuring the same eigenvector twice, we would see Dirac delta functions that represent the penalty factors added to the measured states.

of building an unbiased and heterogeneous benchmark, we have developed an instance generator script. This script takes three different values as configuration parameters: *i*) the number of items (n), *ii*) the bin capacity (C), and *iii*) the random distribution of item weights that compose the instance. In reference to the number of items, two different values have been considered: 10 and 12. These sizes are sufficiently large for finding meaningful outcomes, regarding the limited resources of the current QC devices. For the maximum capacity, 100, 120 and 150 values have been deemed. As distribution functions, three configuration have been employed: a single Gaussian centered in $C/2$, a 2-component Gaussian mixture model with centers established in $C/3$ and $2C/3$, and a uniform distribution between 0 and C. We summarize the characteristics of the whole benchmark in Table 1. Finally, in order to assure the replicability of this study, all the algorithms, instances, and data used in this work are publicly available[7].

As mentioned in the previous paragraph, the size of the instances has been chosen considering the QC hardware limitations, but also the computational restrictions of the brute-force algorithm used for measuring the quality of our developed H-BPP. On this regard, as a preliminary study, our main objective is to compare the outcomes obtained by our implemented method with the optimal results of each instance. For carrying out this task, we have developed a brute-force algorithm which exhaustively seeks for the minimum number of bins needed for storing all the items of each instance.

The procedure followed by the brute-force algorithm is the following one. First, the technique checks if all items of the instance fit into a single bin (note that none of these cases have been contemplated in our benchmark). Otherwise, it starts an iterative process using a starting exploratory value of $b = 2$, and it evaluates all possible feasible combinations for item allocation in b number of bins. In case all items cannot be inserted into b bins due to capacity

---

[7]https://bitbucket.org/mikel_gda/bpp

restrictions, b value is increased by 1. This iterative procedure is executed until the algorithm finds a solution in which all items fit into b bins. After finding the optimal value of b, and in the interest of exhaustive experimentation, the algorithm continues the execution in order to find each and every possible item assignment combination considering those b bins. In other words, the brute-force algorithm finds all the optimal solutions, if more than one exists. It is noteworthy that we define two L and L' solutions as different if the arrangement of items $\{x_1^{(1)}, \ldots, x_n^{(1)}\}, \ldots, \{x_1^{(b)}, \ldots, x_n^{(b)}\}$ differs on its composition, and not in the artificial order promoted by the bin code j.

Lastly, the circuit implementing the digitized quantum annealing algorithm has been simulated with a classical computer. As a first approximation to the problem, we have employed an ideal setup, in which we have assumed we have an ATA connectivity and noiseless gates. For the initial Hamiltonian, we used $H_0 = \sum_j h_0 \sigma_j^x$, where $h_0 > 0$ is a tweakable parameter. A more rigorous analysis would require to introduce the effect of noise and the extra steps required to simulate the evolution in a hardware with limited resources. For this work, we run the algorithm with the following set of parameters: #runs=1000, $\beta = min_j(w_j)/5$, $\tau = 10^{-14}s$, $n_T = 500$, and $h_0 = 10$. We have selected these parameters based on previous experience solving similar problems and adjusting them to obtain the maximum fidelity within a reasonable range of values [31].

5.2 Results and Discussion

Table 2 summarizes the outcomes reached by H-BPP for all instances. Each entry on the table depicts the following metrics:

- *Fitness average ($\bar{b}$)*: this metric shows the average value of b found by H-BPP, considering that the lower the value, the better the result.
- *Average number of optima ($\overline{\#optima}$)*: considering that each instance counts with more than one optimal solution, the value $\overline{\#optima}$ represents the average number of optima that the H-BPP has found per each run.
- *Average size of the ideal subset in $\mathcal{F}$*: considering one specific instance of the 1dBPP, the ideal subset is composed of all the B bin configurations needed to build the whole set of optima solutions. On this regard, the *average size of the ideal subset $\mathcal{F}$* represents the average amount of bins that the H-BPP has found per run which are part of this complete ideal subset. This metric helps to independently assess the H-BPP performance.

These results have been computed over 10 independent runs conducted for each instance. At this point, it should be pointed that $\overline{\#optima}$ and the *average size of the ideal subset in $\mathcal{F}$* are calculated using these runs in which the algorithm has obtained the optimal value. Furthermore, we also show in Table 2 the optimal values for each test case on each of the considered metrics. Thus, we provide the optimal solution (*Opt.*) and the total number of optimum configurations (*#optima*) for each instance, as well as the size of the optimal ideal subset (*size of opt. ideal sub.*).

Several remarks can be drawn from these outcomes. The first one is to highlight the performance of the proposed method on instances composed by 10 items, in which the H-BPP reaches the optimal results in the 100% of the cases. It is important to observe

Table 1: Description of the 18 1dBPP instances generated for measuring the quality of the developed H-BPP algorithm

Instance	# of items	min-max weight	capacity	Distribution	Instance	# of items	min-max weight	capacity	Distribution
10_100_2G	10	30-80	100	Double Gaussian	12_100_1G	12	10-70	100	Single Gaussian
10_120_2G	10	30-80	120	Double Gaussian	12_120_1G	12	20-100	120	Single Gaussian
10_150_2G	10	40-100	150	Double Gaussian	12_150_1G	12	40-120	150	Single Gaussian
12_100_2G	12	10-90	100	Double Gaussian	10_100_U	10	10-70	100	Uniform
12_120_2G	12	30-110	120	Double Gaussian	10_120_U	10	10-100	120	Uniform
12_150_2G	12	30-120	150	Double Gaussian	10_150_U	10	20-140	150	Uniform
10_100_1G	10	20-80	100	Single Gaussian	12_100_U	12	10-60	100	Uniform
10_120_1G	10	20-70	120	Single Gaussian	12_120_U	12	10-110	120	Uniform
10_150_1G	10	30-120	150	Single Gaussian	12_150_U	12	20-130	150	Uniform

Table 2: Results obtained by the H-BPP and their comparison with the optima values.

Instance	H-BPP $\bar{b}$	$\overline{\#optima}$	Avg. size of ideal sub.	Opt.	Optimum values $\#optima$	size of opt. ideal sub.	Instance	H-BPP $\bar{b}$	$\overline{\#optima}$	Avg. size of ideal sub.	Opt.	Optimum values $\#optima$	size of opt. ideal sub.
10_100_2G	6.0	20.1	22.7	6	36	26	12_100_1G	6.0	103.4	61.6	6	5376	110
10_120_2G	6.0	185.3	49.8	5	284	59	12_120_1G	7.6	10.2	23.2	7	260	38
10_150_2G	5.0	8.0	15.5	5	24	20	12_150_1G	7.3	5.8	19.8	7	162	34
12_100_2G	5.4	2.1	12.1	5	36	31	10_100_U	4.0	688.7	113.4	4	1574	111
12_120_2G	8.4	14.1	18.8	8	180	29	10_120_U	6.0	140.9	48.9	6	285	59
12_150_2G	6.9	1.0	13.0	6	30	21	10_150_U	5.0	5.1	14.7	5	8	17
10_100_1G	5.0	188.1	55.4	5	500	72	12_100_U	6.0	370.7	58.3	6	3132	95
10_120_1G	4.0	126.2	60.7	4	337	79	12_120_U	7.0	59.9	46.1	7	1230	92
10_150_1G	6.0	15.9	20.3	6	42	25	12_150_U	7.3	10.4	29.2	7	224	54

that this trend is maintained regardless the total number of optima or the size of the optimal ideal subset, assuming that an instance with less number of optima or a reduced optimal ideal subset is harder to be optimized. The performance demonstrated by H-BPP in the instance 10_150_U is a specially noteworthy example of this situation. This instance counts with 8 optimum values and a size of optimal ideal subset of 17. These values makes 10_150_U the most complex 10-item case to solve. In any case, H-BPP obtains not only the optimum in all the 10 independent runs (which is the main objective of the method), but a remarkable amount of optima (more than the 60% of them) and a significant percentage of the ideal subset (higher than the 80%).

Regarding the instances composed by 12 items, despite some results are slightly degraded, the overall performance of the method is still meritorious considering the current capacities of QC resources and that the input parameters of the quantum annealing subroutine have not been optimized in this preliminary study. In this way, there are three specific 12-item instances in which the H-BPP gets the optimal results on each of the 10 independent executions(12_100_1G, 12_100_U and 12_120_U). If we further analyze the composition of these cases, it can be seen how they count with a high number of optima. This situation makes easier the optimal resolution of these instances. In any case, it can be observed how the H-BPP obtains a significant fraction of the optima in more complex examples, such as 12_100_2G or 12_150_1G.

Moreover, we have conducted an additional analysis in Table 3, in which we compute the *average size of the ideal subset in* $\mathcal{F}$ obtained by the H-BPP for the runs in which the optimal solution has not been reached. Our main objective with this is to better

understand the behaviour of the method in these specific cases. In this sense, and conducting a deeper analysis of these results, it can be seen in instances such as 12_150_2G, 12_120_1G or 12_150_U, that attaining more than the 50% of the ideal subsets does not guarantee the obtention of any optimal solution. This situation unveils the complexity of some instances, in which the weight of finding some concrete subsets is crucial for reaching the optimum of the problem. In other words, some specific examples are more restrictive than others, since few limited subsets are compulsory needed for obtaining any optima. Not finding these subsets in the quantum annealing algorithm for the subset sampling subroutine (Section 4.1) leads to the impossibility of the method to reach the optimal solution.

Table 3: Average size of ideal subset found by H-BPP in those runs that the optimal solution is not found.

Instance	Avg. size of ideal sub.	Instance	Avg. size of ideal sub.
12_100_2G	10.2 (out of 31)	12_120_1G	19.1 (out of 38)
12_120_2G	15.2 (out of 29)	12_150_1G	17.0 (out of 34)
12_150_2G	11.5 (out of 21)	12_150_U	28.i3 (out of 54)

6 CONCLUSIONS AND FUTURE WORK

In this work, we present a new hybrid quantum-classical algorithm for solving 1dBPP. The idea that we have introduced is to use a quantum annealing algorithm to reduce the space of possible solutions. This is performed by searching for feasible configurations of items fitting in a single bin. Within this reduced search space,

the algorithm implements a classical heuristic for generating the configuration minimizing the number of containers. As a proof of concept, we have compared the accuracy of our results against a classical brute-force algorithm. Employing a benchmark composed by 18 randomly generated instances with different number of packages and weight distributions, we have found that our algorithm is capable of reliably finding the optimal solution for the smaller instances. For the instances with a higher number of packages, the performance of the algorithm slightly degrades. However, the reduction of the search space and the expected resource consumption of the quantum annealing algorithm in terms of the qubit number, allows us to find approximate solutions for larger instances, while the brute-force algorithm can not find the solution in a reasonable time.

Despite of the simplicity of the problem solved in this work, this paves the way for addressing more complex BPP problem variants (such as the two-dimensional or three-dimensional BPPs) using hybrid quantum-classical algorithms. There is room for improvement regarding the quantum annealing subroutine, and the selection of its hyper-parameters. As a future work, we have planned to improve the classical module of the H-BPP, in order to increase the probability of finding the subset of packages found in the optimal configurations. In this regard, we need to perform further analysis of the algorithm to prove the possible advantage of using a complete quantum algorithm for the whole process. We also leave the possibility of expanding the algorithm developed in this exploratory work to other problems whose solution can also be divided into different partial solutions.

ACKNOWLEDGMENTS

REFERENCES

[1] Scott Aaronson. 2010. BQP and the Polynomial Hierarchy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing* (Cambridge, Massachusetts, USA) *(STOC '10)*. Association for Computing Machinery, New York, NY, USA, 141–150.

[2] Diaa Salama Abdul-Minaam, Wadha Mohammed Edkheel Saqar Al-Mutairi, Mohamed A Awad, and Walaa H El-Ashmawi. 2020. An adaptive fitness-dependent optimizer for the one-dimensional bin packing problem. *IEEE Access* 8 (2020), 97959–97974.

[3] Akshay Ajagekar, Travis Humble, and Fengqi You. 2020. Quantum computing based hybrid solution strategies for large-scale discrete-continuous optimization problems. *Computers & Chemical Engineering* 132 (2020), 106610.

[4] FF Al Adeh. 2017. Natural Limitations of Quantum Computing. *Int J Swarm Intel Evol Comput* 6, 152 (2017), 2.

[5] Jonathan Allcock, Yassine Hamoudi, Antoine Joux, Felix Klingelhöfer, and Miklos Santha. 2021. Classical and quantum dynamic programming for Subset-Sum and variants. arXiv:2111.07059 [quant-ph]

[6] Ricardo De Almeida and Maria Teresinha Arns Steiner. 2016. Resolution of one-dimensional bin packing problems using augmented neural networks and minimum bin slack. *International Journal of Innovative Computing and Applications* 7, 4 (2016), 214–224.

[7] M. H. S. Amin. 2009. Consistency of the Adiabatic Theorem. *Physical Review Letters* 102 (2009), 220401. Issue 22.

[8] Enrico Angelelli, Nicola Bianchessi, and Carlo Filippi. 2014. Optimal interval scheduling with a resource constraint. *Computers & operations research* 51 (2014), 268–281.

[9] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.

[10] Nurşen Aydın, İbrahim Muter, and Ş İlker Birbil. 2020. Multi-objective temporal bin packing problem: An application in cloud computing. *Computers & Operations Research* 121 (2020), 104959.

[11] N Aydin, I Muter, and S Ilker Birbil. 2019. Bin packing problem with time dimension: An application in cloud computing.

[12] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. Las Heras, R. Babbush, A. G. Fowler, B. Campbell, Yu Chen, Z. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, E. Solano, H. Neven, and John M. Martinis. 2016. Digitized adiabatic quantum computing with a superconducting circuit. *Nature* 534, 7606 (2016), 222–226.

[13] Daniel J. Bernstein, Stacey Jeffery, Tanja Lange, and Alexander Meurer. 2013. Quantum Algorithms for the Subset-Sum Problem. In *Post-Quantum Cryptography*, Philippe Gaborit (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 16–33.

[14] Luca Bertazzi, Bruce Golden, and Xingyin Wang. 2019. The bin packing problem with item fragmentation: a worst-case analysis. *Discrete Applied Mathematics* 261 (2019), 63–77.

[15] Vanessa MR Bezerra, Aline AS Leao, José Fernando Oliveira, and Maristela O Santos. 2020. Models for the two-dimensional level strip packing problem–a review and a computational evaluation. *Journal of the Operational Research Society* 71, 4 (2020), 606–627.

[16] Xavier Bonnetain, Rémi Bricout, André Schrottenloher, and Yixin Shen. 2020. Improved classical and quantum algorithms for subset-sum. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 633–666.

[17] Xavier Bonnetain, Rémi Bricout, André Schrottenloher, and Yixin Shen. 2020. Improved Classical and Quantum Algorithms for Subset-Sum. arXiv:2002.05276 [quant-ph]

[18] M. Born and V. Fock. 1928. Beweis des Adiabatensatzes. *Zeitschrift für Physik* 51, 3 (1928), 165–180.

[19] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press.

[20] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. 2021. Variational quantum algorithms. *Nature Reviews Physics* 3, 9 (2021), 625–644.

[21] Edward G Coffman, Jr, Michael R Garey, and David S Johnson. 1983. Dynamic bin packing. *SIAM J. Comput.* 12, 2 (1983), 227–258.

[22] János Csirik and Csanád Imreh. 2018. Variants of Classical One Dimensional Bin Packing. In *Handbook of Approximation Algorithms and Metaheuristics, Second Edition*. Chapman and Hall/CRC, 519–538.

[23] David Elieser Deutsch. 1989. Quantum computational networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences* 425, 1868 (1989), 73–90.

[24] Yongcheng Ding, Xi Chen, Lucas Lamata, Enrique Solano, and Mikel Sanz. 2021. Implementation of a Hybrid Classical-Quantum Annealing Algorithm for Logistic Network Design. *SN Computer Science* 2, 2 (2021), 68.

[25] Krzysztof Domino, Mátyás Koniorczyk, Krzysztof Krawiec, Konrad Jałowiecki, Sebastian Deffner, and Bartłomiej Gardas. 2021. Quantum annealing in the NISQ era: railway conflict management. arXiv:2112.03674 [quant-ph]

[26] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A Quantum Approximate Optimization Algorithm. arXiv:1411.4028 [quant-ph]

[27] Marco Fellous-Asiani, Jing Hao Chai, Robert S Whitney, Alexia Auffèves, and Hui Khoon Ng. 2020. Limitations in quantum computing from resource constraints. arXiv:2007.01966 [quant-ph]

[28] Krzysztof Fleszar and Khalil S Hindi. 2002. New heuristics for one-dimensional bin-packing. *Computers & operations research* 29, 7 (2002), 821–839.

[29] Toru Fujimura. 2021. Quantum Algorithm for Bin-Packing. *Global Journal of Pure and Applied Mathematics* 17, 1 (2021), 9–15.

[30] Asier Galicia, Borja Ramon, Enrique Solano, and Mikel Sanz. 2020. Enhanced connectivity of quantum hardware with digital-analog control. *Phys. Rev. Research* 2 (2020), 033103. Issue 3.

[31] Mikel Garcia de Andoin, Ana Martin, and Mikel Sanz. 2022. Digital Analog Quantum Algorithm for Logistic Network Design Problem. *to be published, available under request* (2022).

[32] Michael R Garey and David S Johnson. 1981. Approximation algorithms for bin packing problems: A survey. In *Analysis and design of algorithms in combinatorial optimization*. Springer, 147–172.

[33] Lov K. Grover. 1996. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (Philadelphia, Pennsylvania, USA) *(STOC '96)*. Association for Computing Machinery, New York, NY, USA, 212–219.

[34] N. N. Hegade, P. Chandarana, K. Paul, X. Chen, F. Albarrán-Arriagada, and E. Solano. 2021. Portfolio Optimization with Digitized-Counterdiabatic Quantum Algorithms. arXiv:2112.08347 [quant-ph]

[35] Dylan Herman, Cody Googin, Xiaoyuan Liu, Alexey Galda, Ilya Safro, Yue Sun, Marco Pistoia, and Yuri Alexeev. 2022. A Survey of Quantum Computing for Finance. arXiv:2201.02773 [quant-ph]

[36] Johan Håstad. 2001. Some Optimal Inapproximability Results. *J. ACM* 48, 4 (2001), 798–859.

[37] Tomi H. Johnson, Stephen R. Clark, and Dieter Jaksch. 2014. What is a quantum simulator? *EPJ Quantum Technology* 1, 1 (2014), 10.

[38] Tosio Kato. 1950. On the Adiabatic Theorem of Quantum Mechanics. *Journal of the Physical Society of Japan* 5, 6 (1950), 435–439.

[39] Kenneth L Krause, Vincent Y Shen, and Herbert D Schwetman. 1975. Analysis of several task-scheduling algorithms for a model of multiprogramming computer systems. *Journal of the ACM (JACM)* 22, 4 (1975), 522–550.

[40] Tayfun Kucukyilmaz and Hakan Ezgi Kiziloz. 2018. Cooperative parallel grouping genetic algorithm for the one-dimensional bin packing problem. *Computers & Industrial Engineering* 125 (2018), 157–170.

[41] Soukaina Laabadi, Mohamed Naimi, Hassan El Amri, and Boujemâa Achchab. 2019. A crow search-based genetic algorithm for solving two-dimensional bin packing problem. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*. Springer, 203–215.

[42] Abdesslem Layeb and Seriel Rayene Boussalia. 2012. A Novel Quantum Inspired Cuckoo Search Algorithm for Bin Packing Problem. *International Journal of Information Technology and Computer Science(IJITCS)* 4, 5 (2012), 58–67.

[43] Frank Leymann and Johanna Barzen. 2020. The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Science and Technology* 5, 4 (2020), 044007.

[44] Andrea Lodi, Silvano Martello, and Michele Monaci. 2002. Two-dimensional packing problems: A survey. *European journal of operational research* 141, 2 (2002), 241–252.

[45] Andrea Lodi, Silvano Martello, Michele Monaci, and Daniele Vigo. 2014. Two-Dimensional Bin Packing Problems. *Paradigms of combinatorial optimization: Problems and new approaches* (2014), 107–129.

[46] Andrea Lodi, Silvano Martello, and Daniele Vigo. 2002. Heuristic algorithms for the three-dimensional bin packing problem. *European Journal of Operational Research* 141, 2 (2002), 410–420.

[47] P López. 2019. Introducing an approach based on an advanced hybrid model for the two-dimensional bin packing problem. *Annals of Electrical and Electronic Engineering* 2, 11 (2019), 7–13.

[48] EG Co man Jr, MR Garey, and DS Johnson. 1996. Approximation algorithms for bin packing: A survey. *Approximation algorithms for NP-hard problems* (1996), 46–93.

[49] S. Marsh and J. B. Wang. 2019. A quantum walk-assisted approximate algorithm for bounded NP optimisation problems. *Quantum Information Processing* 18, 3 (2019), 61.

[50] Silvano Martello, David Pisinger, and Daniele Vigo. 2000. The three-dimensional bin packing problem. *Operations research* 48, 2 (2000), 256–267.

[51] Chanaleä Munien and Absalom E Ezugwu. 2021. Metaheuristic algorithms for one-dimensional bin-packing problems: A survey of recent advances and applications. *Journal of Intelligent Systems* 30, 1 (2021), 636–663.

[52] Chanaleä Munien, Shiv Mahabeer, Esther Dzitiro, Sharad Singh, Siluleko Zungu, and Absalom El-Shamir Ezugwu. 2020. Metaheuristic Approaches for One-Dimensional Bin Packing Problem: A Comparative Performance Study. *IEEE Access* 8 (2020), 227438–227465.

[53] AV Navaneeth and MR Dileep. 2021. A study and analysis of applications of classical computing and quantum computing: A survey. In *ICT Analysis and Applications*. Springer, 235–246.

[54] Óscar Oliveira, Telmo Matos, and Dorabela Gamboa. 2019. Adaptive Sequence-Based Heuristic for the Three-Dimensional Bin Packing Problem. In *International Conference on Learning and Intelligent Optimization*. Springer, 69–76.

[55] Sukru Ozer Ozcan, Tansel Dokeroglu, Ahmet Cosar, and Adnan Yazici. 2016. A novel grouping genetic algorithm for the one-dimensional bin packing problem on gpu. In *International Symposium on Computer and Information Sciences*. Springer, 52–60.

[56] Célia Paquay, Sabine Limbourg, and Michaël Schyns. 2018. A tailored two-phase constructive heuristic for the three-dimensional Multiple Bin Size Bin Packing Problem with transportation constraints. *European Journal of Operational Research* 267, 1 (2018), 52–64.

[57] Adrian Parra-Rodriguez, Pavel Lougovski, Lucas Lamata, Enrique Solano, and Mikel Sanz. 2020. Digital-analog quantum computation. *Phys. Rev. A* 101 (2020), 022305. Issue 2.

[58] Sergey Polyakovskiy and Rym M'Hallah. 2018. A hybrid feasibility constraints-guided search to the two-dimensional bin packing problem with due dates. *European Journal of Operational Research* 266, 3 (2018), 819–839.

[59] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (2018), 79.

[60] Luigi Di Puglia Pugliese, Francesca Guerriero, and Roberto Calbi. 2019. Solving a Three-Dimensional Bin-Packing Problem Arising in the Groupage Process: Application to the Port of Gioia Tauro. In *A View of Operations Research Applications in Italy, 2018*. Springer, 29–40.

[61] Lauren Pusey-Nazzaro and Presanna Date. 2020. Adiabatic quantum optimization fails to solve the knapsack problem. arXiv:2008.07456 [qunat-ph]

[62] Abdullah Ash Saki, Mahabubul Alam, Koustubh Phalak, Aakarshitha Suresh, Rasit Onur Topaloglu, and Swaroop Ghosh. 2021. A survey and tutorial on security and resilience of quantum computing. In *2021 IEEE European Test Symposium (ETS)*. IEEE, 1–10.

[63] Ines Sbai and Saoussen Krichen. 2019. A Hybrid PSO-LS approach for solving the Two-Dimensional Bin Packing Problem with weight capacities constraint: A case study. In *Proceedings of the 9th International Conference on Information Systems and Technologies*. 1–8.

[64] P.W. Shor. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*. 124–134.

[65] Kristina Yancey Spencer, Pavel V Tsvetkov, and Joshua J Jarrell. 2019. A greedy memetic algorithm for a multiobjective dynamic bin packing problem for storing cooling objects. *Journal of Heuristics* 25, 1 (2019), 1–45.

[66] Andrew Steane. 1998. Quantum computing. *Reports on Progress in Physics* 61, 2 (1998), 117.

[67] Daniel Stilck França and Raul García-Patrón. 2021. Limitations of optimization algorithms on noisy quantum devices. *Nature Physics* 17, 11 (2021), 1221–1227.

[68] Bhagaban Swain, Rajdeep Ghosh, and Pranjal Borah. 2014. Quantum Evolutionary Algorithm for Solving Bin Packing Problem. *International Journal of Engineering Science Invention* 3 (2014), 01–07. Issue 8.

[69] Alessio Trivella and David Pisinger. 2016. The load-balanced multi-dimensional bin-packing problem. *Computers & Operations Research* 74 (2016), 152–164.

[70] Mark Van Houdenhoven, Jeroen M Van Oostrum, Erwin W Hans, Gerhard Wullink, and Geert Kazemier. 2007. Improving operating room efficiency by applying bin-packing and portfolio techniques to surgical case scheduling. *Anesthesia & Analgesia* 105, 3 (2007), 707–714.

[71] Lijun Wei, Zhixing Luo, Roberto Baldacci, and Andrew Lim. 2020. A new branch-and-price-and-cut algorithm for one-dimensional bin-packing problems. *INFORMS Journal on Computing* 32, 2 (2020), 428–443.

[72] Yulin Wu, Wan-Su Bao, Sirui Cao, Fusheng Chen, Ming-Cheng Chen, Xiawei Chen, Tung-Hsun Chung, Hui Deng, Yajie Du, Daojin Fan, Ming Gong, Cheng Guo, Chu Guo, Shaojun Guo, Lianchen Han, Linyin Hong, He-Liang Huang, Yong-Heng Huo, Liping Li, Na Li, Shaowei Li, Yuan Li, Futian Liang, Chun Lin, Jin Lin, Haoran Qian, Dan Qiao, Hao Rong, Hong Su, Lihua Sun, Liangyuan Wang, Shiyu Wang, Dachao Wu, Yu Xu, Kai Yan, Weifeng Yang, Yang Yang, Yangsen Ye, Jianghan Yin, Chong Ying, Jiale Yu, Chen Zha, Cha Zhang, Haibin Zhang, Kaili Zhang, Yiming Zhang, Han Zhao, Youwei Zhao, Liang Zhou, Qingling Zhu, Chao-Yang Lu, Cheng-Zhi Peng, Xiaobo Zhu, and Jian-Wei Pan. 2021. Strong Quantum Computational Advantage Using a Superconducting Quantum Processor. *Phys. Rev. Lett.* 127 (2021), 180501. Issue 18.

[73] Ahmed Younes and Jonathan E. Rowe. 2015. A Polynomial Time Bounded-error Quantum Algorithm for Boolean Satisfiability. arXiv:1507.05061 [cs.CC]