# Hardware Efficient Framework for QAOA

Thiago Assis*, Laila Lopes*, Cristiano Arbex, Henrique Assumpção, Pedro Baptista, Rodrigo Chaves, Diego Ferreira, Luan Henrique, Mathias Oliveira e Gabriel Coutinho*

***Abstract*— The Quantum Approximate Optimization Algorithm (QAOA) is a leading candidate for solving Quadratic Unconstrained Binary Optimization (QUBO) problems. However, its performance on Noisy Intermediate-Scale Quantum (NISQ) devices is hindered by limited qubit connectivity, which increases circuit depth. We introduce a hardware-aware QAOA framework, using a Mixed Integer Semidefinite Programming (MISDP) formulation to find an approximate cost Hamiltonian compatible with hardware topology. To solve the MISDP, we propose heuristics based on spectral properties of the hardware graph. We apply this framework to the index tracking problem, and our numerical results demonstrate that this approach yields high-quality solutions.**

***Keywords*— QAOA, NISQ, MISDP, Index Tracking**

## I. Introduction

In the near term, one of the most anticipated promises of Quantum Computing lies in solving Quadratic Unconstrained Binary Optimization (QUBO) problems, a NP-hard class of problems with applications in many areas, including finance, logistics and chemistry [1], [2], [3]. A prominent quantum approach for the solution of such problems is the Quantum Approximate Optimization Algorithm (QAOA), which promises to find high-quality approximate solutions efficiently [4], [5].

However, one of the principal barriers on the use of QAOA lies in the restrictive nature of today's Noise Intermediate Scale Quantum (NISQ) hardware [6]. When the interaction between qubits in the QAOA circuit does not align with the hardware's topology, the quantum transpiler must insert a large number of SWAP operations to enable such interactions, significantly increasing the depth of the circuit.

To overcome this hardware-imposed limitation, we introduce a framework for designing hardware-aware QAOA circuits. We tailor this approach specifically for the index tracking problem, although the framework can be used in any other problem framed as a QUBO with or without cardinality constraints.

We begin Section II by defining the index tracking problem and presenting its formulation as a QUBO with cardinality constraints.

In Section III, we review the standard application of QAOA to this QUBO. Subsequently, we identify a critical flaw within this framework in the NISQ era: circuits generated by this standard approach often have a depth that is prohibitive for current quantum devices, leading to significant performance degradation from decoherence and accumulated gate errors.

Our primary contribution, presented in Section IV is a method to mitigate this issue. We introduce a Mixed Integer Semidefinite Programming (MISDP) formulation whose solution is used for the construction of a hardware-friendly QAOA framework. We further propose heuristics to solve the MISDP efficiently for large-scale instances.

Finally, in Section V, we provide numerical results validating the performance of our proposed heuristics and demonstrating the efficacy of the overall framework in generating high-quality solutions with hardware-efficient circuits.

The experiments are based on the architecture of OQC's Toshiko Gen 1 quantum computer. We define the hardware graph of a quantum computer as the graph with vertices as the qubits and edges whenever the two qubits allow 2-qubit operations.

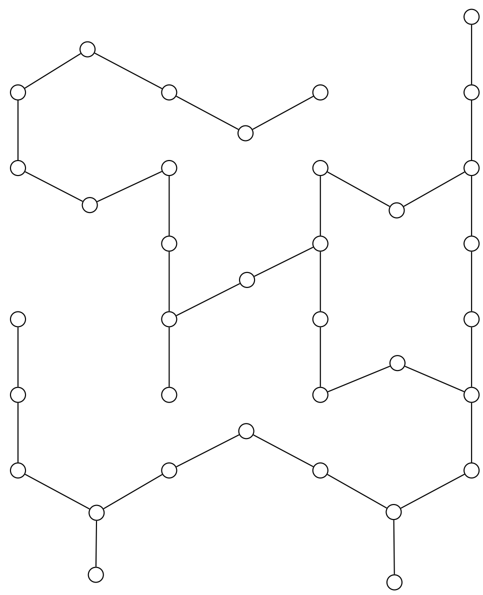

Fig. 1. The hardware graph for OQC's Toshiko Gen 1 quantum computer

*These authors are the main authors, and contributed equally to this work. Thiago Assis, Dept. of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: thiago.assis@dcc.ufmg.br; Laila Lopes, Dept. of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: lailalopes@dcc.ufmg.br; Cristiano Arbex, Dept. of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: arbex@dcc.ufmg Henrique Assumpção, Dept. of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: henriquesoares@dcc.ufmg.br; Pedro Baptista, Dept. of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: pedro.baptista@dcc.ufmg.br; Rodrigo Chaves, OQC, Oxford Quantum Circuits, London-UK, e-mail: rchaves@oqc.tech; Diego Ferreira, Banco Inter, Belo Horizonte-MG, e-mail: e-mail: diego.ferreira@inter.co; Luan Henrique, Dept. of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: luan.costa@dcc.ufmg.br; Mathias Oliveira, Dept. of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: oliveira.mathias@dcc.ufmg.br; Gabriel Coutinho, Dept. of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte-MG, e-mail: gabriel@dcc.ufmg.br;

## II. The Index Tracking Problem

In finance, creating a portfolio that consistently outperforms the market is extremely difficult. Extensive historical data from the SPIVA U.S. Scorecard [7] show that the vast majority of fund managers tend to underperform the S&P 500 index. This can be explained by the Efficient Market Hypothesis [8], which suggests that asset prices fully reflect all available information, making it extremely hard to leverage potential price inefficiencies into gains.

Given this scenario, index tracking has emerged as a compelling alternative [9]. Rather than attempting the statistically improbable task of outperforming the market, this strategy seeks to simply replicate the performance of a market benchmark, such as the S&P 500. By capitalizing on the market's efficiency instead of fighting it, investors gain broad diversification while avoiding the high management fees and trading commissions inherent to active strategies [10].

While a simple approach for index tracking would be to purchase all assets from the index in the same exact weights, this would lead to frequent rebalancing and the possession of numerous small positions, making such portfolio undesirable at the least to manage. Therefore, a better strategy is to construct an index fund where the number of managed assets is substantially smaller than the number of assets in the benchmark index, yet still behaves similarly to the index.

### A. QUBO formulation for Index Tracking

The authors of [11] propose a solution for index tracking well-suited for quantum computers. Their approach is to select a small, representative subset of assets by using the k-medoid clustering technique.

Consider a set of $m$ points $S = \{x_1, \ldots, x_m\} \subseteq \mathbb{R}^n$. The sample mean is

$$\operatorname{argmin} \sum_{i=1}^{m} ||y - x_i||^2 : y \in \mathbb{R}^n. \tag{1}$$

A less widely measure is the sample medoid, defined as

$$\operatorname{argmin} \sum_{i=1}^{m} ||y - x_i||^2 : y \in S. \tag{2}$$

Contrary to the sample mean, the sample medoid is always guaranteed to be a data point. The $k$-medoid clustering algorithm is analouguos to $k$-means, but now each cluster center is required to be a data point. This is ideal for index tracking, as it guarantees that the data centers are actual investible assets from the index.

We can frame the index tracking problem as $k$-medoid clustering, and use a QUBO formulation to solve the clustering problem. The goal is to find an optimal solution of

$$\min x^T C x - \alpha \mathbf{1}^T C x : x^T \mathbf{1} = k, x \in \{0,1\}^n \tag{3}$$

where $C$ is some matrix that indicates similarity between the assets, $\mathbf{1}$ is the all ones vector, and $\alpha$ is an empirically determined constant. Following the example of [12], we set $\alpha = 1/k$ and take $C$ to be

$$C_{ij} = 1 - \exp\left( -\frac{\sqrt{2(1 - \text{Correlation}_{i,j})}}{2} \right). \tag{4}$$

This QUBO formulation searches for a subset $x$ consisting of $k$ assets from the index that are not similar from each other, hence minimizing $x^T C x$, and that are at the same time central to the clusters, by maximizing $\mathbf{1}^T C x$.

Note that as $x \in \{0,1\}^n$, we can incorporate the $\mathbf{1}^T C x$ term inside the $x^T C x$ one. By defining

$$\hat{C}_{ij} = \begin{cases} C_{ii} - \alpha \sum_j C_{ij} & \text{if } i = j \\ C_{ij} & \text{otherwise,} \end{cases} \tag{5}$$

we can rewrite the QUBO formulation as:

$$\min x^T \hat{C} x : x^T \mathbf{1} = k, x \in \{0,1\}^n. \tag{6}$$

Differently from [11], here we use a simpler objective function by omitting penalty terms associated with the restriction $\mathbf{1}^T x = k$. We use an alternate version of the QAOA algorithm, adapted for cardinality constrained problems. This simplifies the optimization problem, without compromising the quality of the solution.

## III. The QAOA framework

The Quantum Approximate Optimization Algorithm (QAOA) was originally introduced in [4], and has attracted great interest since. It is a hybrid variational algorithm designed for combinatorial problems. In our case, we will treat QUBO problems with cardinality constraints:

$$\min \left\{ x^T \hat{C} x : x^T \mathbf{1} = k, x \in \{0,1\}^n \right\} \tag{7}$$

To apply the QAOA framework, we must map the objective function to a cost Hamiltonian, given by

$$H_{\hat{C}} = \sum_{i \neq j} J_{ij} Z_i Z_j + \sum_i h_i Z_i \tag{8}$$

where $Z_i$ is the $Z$ gate applied to the $i$-th qubit, and the coefficients are defined as

$$J_{ij} = \frac{\hat{C}_{ij}}{4} \quad \text{and} \quad h_i = -\frac{\hat{C}_{ii}}{2} - \sum_{j \neq i} \frac{\hat{C}_{ij}}{4} \tag{9}$$

By construction, the cost Hamiltonian satisfies the following property: for any binary vector $x \in \{0,1\}^n$,

$$x^T \hat{C} x = \langle x | H_{\hat{C}} | x \rangle + c_0 \tag{10}$$

for some constant term $c_0$.

Furthermore, we need to define some mixer Hamiltonian $H_M$. In [14], it was noted that the use of $XY$ mixers is advantageous in cardinality constrained problems, because they preserve the Hamming weight of a state. Thus, we adopt the mixer:

$$H_M = \frac{1}{2} \sum_{ij \in E(G)} (X_i X_j + Y_i Y_j) \tag{11}$$

where $G$ is the hardware graph. Thus, the above sum is taken over all pairs of qubits physically connect on the quantum computer.

Then, we create a parametrized quantum circuit by applying $p$ mixing and cost layers:

$$U(\gamma, \beta) = \prod_{j=1}^{p} e^{-i\beta_j H_M} e^{-i\gamma_j H_{\hat{C}}} \tag{12}$$

The base state is chosen as the Dicke State $|D_k^n\rangle$, the uniform superposition over all feasible solutions, i.e., binary strings $x$ such that $\mathbf{1}^T x = k$. The Dicke State can be written as:

$$|D_k^n\rangle = \frac{1}{\sqrt{\binom{n}{k}}} \sum_{\mathbf{1}^T x = k} |x\rangle \tag{13}$$

An efficient algorithm for the construction of the Dicke State was proposed in [13], and we adopt this construction in our framework.

### A. QAOA Problems in NISQ Hardware

So far, this approach for QAOA is usual. However, one problem with it is that the cost Hamiltonian is not hardware friendly. When we map the matrix description of the circuit to a hardware one, in a process called transpilation, we significantly increase the depth of the circuit. The general approach is to decompose each cost layer $e^{-i\gamma_j H_C}$ into the product of individual terms $\prod_{v \neq u} e^{-i\gamma_j \hat{C}_{vu} Z_v Z_u}$, and implement each of these terms as depicted by Figure 2.

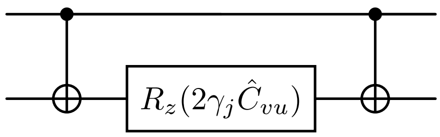


Fig. 2. Circuit implementing $e^{-i\gamma_j \hat{C}_{vu} Z_v Z_u}$. The first wire corresponds to qubit $v$, and the second one corresponds to qubit $u$.

However, if qubits $v, u$ are not physically connected in the quantum hardware, then we must use SWAP operations to move them into adjacent qubits, so that we are able to perform the CNOT operations. It can be showed that if $d$ is the distance between vertices $v, u$ in the hardware graph $G$, then at least $2(d-1)$ SWAPS are required.

In the next section, we propose a solution to this problem, by modifying the usual QAOA framework.

## IV. The MISDP formulation

Let $\tilde{C}$ be the optimal solution for $X$ in the MISDP with integer restrictions defined bellow:

$$\begin{aligned} \min \lambda : \lambda I &\succcurlyeq X - \hat{C} \succcurlyeq -\lambda I, \\ X &\circ P A_{\overline{G}} P^T = 0, \\ P &\text{ is a permutation matrix} \end{aligned} \tag{14}$$

where $A_{\overline{G}}$ is the adjacency matrix of the complement $\overline{G}$ of the hardware graph, $\hat{C}$ is the matrix from the QUBO problem objective function, and $\circ$ is the Haddamard product (element-wise multiplication). Note that the restriction $X \circ A_{\overline{G}} = 0$ forces $X_{vu} = 0$ for any $v \not\sim u$ in $G$.

This formulation accomplishes two tasks simultaneously. First, the constraint $X \circ P^T A_{\overline{G}} P$ enforces the hardware topology. Secondly, the formulation seeks for the optimal permutation $P$ that maps the problem variables to the physical qubits.

Note that a optimal solution has $\lambda = 0$ if and only if $\tilde{C} = \hat{C}$. Also, the value found for $\lambda$ provides a bound for the deviation from the true QUBO optimal value. Indeed, for any binary vector with $\mathbf{1}^T x = k$, we can bound:

$$x^T \hat{C} x - \lambda k \leq x^T \tilde{C} x \leq x^T \hat{C} x + \lambda k \tag{15}$$

### A. A Heuristic for solving the MISDP

Since the optimization over permutation matrices makes solving the MISDP NP-hard, we propose 2 heuristics for finding high-quality solutions in polynomial time.

Let $v$ be the eigenvector associated with the largest eigenvalue of $A_G$, and $u$ be the eigenvector associated with the largest eigenvalue of $\hat{C}$. Define $\pi$ to be a permutation that orders $v$ in non-increasing order, that is

$$v_{\pi(1)} \geq v_{\pi(2)} \geq \cdots \geq v_{\pi(n)} \tag{16}$$

Also, let $\sigma$ be such an permutation for $u$. The order given by $\pi$ defines the most influential nodes in the graph. Similarly, the order given by $\sigma$ defines the most influential assets in the index.

We can define $P$ to be the permutation matrix for the permutation $\pi^{-1}\sigma$, and solve the MISDP with this permutation fixed. This choice for $P$ maps the most central assets from $C$ to the most central qubits in $G$. With $P$ fixed, the MISDP loses the integer restrictions, and can be solved efficiently. This is our first proposed heuristic.

A potential drawback of this approach is that the most central qubits in the hardware graph may not form a connected subgraph. If this occurs, important off-diagonal entries $\hat{C}_{ij}$ (representing high covariance between assets) might be mapped to disconnected qubits, forcing their corresponding values in $\tilde{C}$ to be zero and potentially leading to a suboptimal approximation.

To solve this problem, we also adopt a second heuristic, illustrated in Algorithm 1. First, map the most central asset to the most central vertex graph, i.e., $p[\pi(1)] \leftarrow \sigma(1)$. Also, initialize $S \leftarrow \{\pi(1)\}$. This is the set of vertices (forming a connected component) processed so far. Now, iterate through the assets, in the order defined by $\sigma$, and assign asset $\sigma(i)$ to the graph node $v \in N(S)$ with smallest $\pi(v)$, i.e., the most influential one in $N(S)$.

This method ensures that the most influential assets are mapped to a connected subgraph of the most influential qubits, thereby avoiding suboptimal approximations. In the next section, we show experimental results for the heuristics.

## V. Tests

In this section we present our computational results for both of our heuristics. The financial data for our experiments was sourced from a historical dataset of constituent stocks of the S&P 500 index. For each experimental run, a universe of $n_a$ stocks was created by randomly sampling from this database. The empirical correlation matrix was then computed based on daily returns from a trailing 120-day period.

**Algorithm 1:** Heuristic for the MISDP formulation

**Data:** Permutations $\pi$, $\sigma$
**Result:** Permutation $p$
// $S$ is the set of used vertices
$p[\pi(1)] \leftarrow \sigma(1)$
$S \leftarrow \{\pi(1)\}$
**for** $i = 2, \ldots, n$ **do**
  // Select the most important vertex in $N(S)$
  $v = \text{argmin}_u \{\pi(u) : u \in N(S)\}$
  // Set asset $\sigma(i)$ to vertex $\pi(v)$
  $p[\pi(v)] \leftarrow \sigma(i)$.
  $S \leftarrow S \cup \pi(v)$
**end**

### A. *Tests on random graphs*

These tests were structured to analyze the impact of key problem parameters, such as the size of the asset universe in comparison to the number of graph nodes, the graph density, and the desired portfolio size.

The graphs were algorithmically generated using the Erdős–Rényi model $G(N, p)$. This model creates a graph with $N$ vertices by independently forming an edge between each vertex pair with a specified probability, $p$. The value of $p$ determines the graph's density, enabling the simulation of topologies from sparse to dense. To ensure the validity of our experimental design, we exclusively utilized graphs that satisfied a connectivity constraint.

The primary objective of the experiments is to quantify the performance of two distinct methods, the Simple Heuristic and the Connected Heuristic, by comparing their final portfolio solutions against the true optimal solution derived from a brute-force search. The experiments were conducted on random graphs with a fixed size of $N = 15$, and with $n_a = 15$ possible assets. The key parameters were varied as follows: the graph density $p$ was tested across the set $\{0.2, 0.3, 0.5, 0.7, 0.8\}$, and the portfolio proportion $k$ was set to $\{0.3, 0.5, 0.7\}$ of the available assets.

The experimental results reveal a remarkable convergence in performance between the Connected Heuristic and the Simple Heuristic across all tested scenarios, as illustrated in Figure 3. A quantitative view, presented in Table I, confirms this observation. In sparse graphs ($p < 0.6$), the mean optimality gap for the Connected Heuristic was 17.72%, while the Simple Heuristic achieved a nearly identical gap of 17.78%. A similar outcome was observed in dense graphs ($p \geq 0.6$), where both methods produced a mean optimality gap of approximately 12.04%, indicating no clear performance advantage for either method under these conditions.

TABLE I
MEAN OPTIMALITY GAP (%) FOR EACH HEURISTIC, GROUPED BY GRAPH DENSITY.

| Graph Density | Connected Heuristic | Simple Heuristic |
|---|---|---|
| $p \geq 0.6$ | 10.05 | 10.43 |
| $p < 0.6$ | 17.78 | 16.85 |

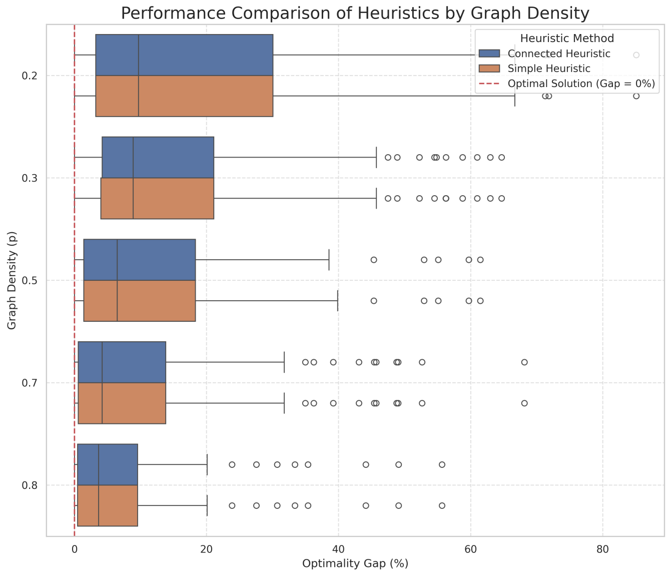


Fig. 3. Performance Comparison of the Connected and Simple Heuristics by Graph Density.

### B. *Tests on OQC's Toshiko Gen 1 Graph*

To evaluate the practical applicability and performance of our heuristics in a real-world scenario, we conducted a case study using the hardware graph of OQC's Toshiko Gen 1 quantum computer. This graph is composed of 34 vertices and is inherently sparse, reflecting the physical constraints of qubit connectivity in near-term quantum hardware.

We performed diverse experiments, varying the number the percentage of the desired portfolio in relation to the assets number, $k = \{0.3, 0.5, 0.7\}$. The primary performance metric for this analysis was the normalized lambda, which quantifies the matrix approximation error. This quantity is defined as the ratio between the found optimal $\lambda$ and the spectral norm of the matrix $\hat{C}$. The results are summarized in Table II. Note that for these specific instances, both heuristics yield the same results.

TABLE II
MEAN AND STANDARD DEVIATION FOR EACH HEURISTIC BY K.

| k | Metric Value (Mean ± Std. Dev.) | |
|---|---|---|
| | Connected Heuristic | Simple Heuristic |
| 0.15 | 0.655 ± 0.0045 | 0.655 ± 0.0045 |
| 0.30 | 0.582 ± 0.0035 | 0.582 ± 0.0035 |
| 0.50 | 0.557 ± 0.0035 | 0.557 ± 0.0035 |
| 0.70 | 0.548 ± 0.0035 | 0.548 ± 0.0035 |

## VI. CONCLUSION

In this work, we addressed a critical challenge in the practical application of the QAOA algorithm: the degradation in circuit depth when transpiling the matrix circuit description to a hardware one, due to the restrictive number of two-qubit iterations allowed in current NISQ technology. We introduced a novel, hardware-friendly framework that reformulates the cost Hamiltonian, aligning the matrix description of the circuit with the device architecture.

Our main contribution is a MISDP formulation that finds the closest hardware-friendly matrix to the original QUBO problem, while simultaneously optimizing the mapping of problem variables to circuit qubits. To solve this MISDP, we propose heuristics based on the spectral properties of the problem and the hardware graph. As demonstrated by our numerical results, this approach generates a sufficiently good approximation, that allows for the problem solution, while also reducing the final depth of the circuit. This reduction in depth is crucial for minimizing the impact of decoherence and gate errors, enabling higher-quality solutions on today's noisy quantum computers.

While we focused on the index tracking problem as a motivating example, the proposed framework is general and can be extended to other QUBO problems. Future work could involve exploring more sophisticated heuristics, analyzing the Hamiltonian approximation error more rigorously, and experimenting the framework on a wider range of optimization problems and quantum hardware.

## VII. Acknowledgements

This work was funded by a cooperation agreement between UFMG and Banco Inter. We also acknowledge support from FAPEMIG and CNPq.

## References

[1] Lucas A. (2014). Ising formulations of many NP problems. Frontiers in Physics, 2, 5.

[2] G. Kochenberger, J. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang & Y. Wang (2014). The unconstrained binary quadratic optimization problem: a survey. Journal of Combinatorial Optimization, 28(1), 58-81.

[3] R. Orús, S. Mugel & E. Lizaso Quantum computing for finance: Overview and prospects. Reviews in Physics, 4, 100028

[4] E. Farhi, J. Goldstone, S. Gutmann A Quantum Approximate Optimization Algorithm, *arXiv preprint arXiv:1411.4028*, 2014.

[5] L. Zhou, S. T. Wang, S. Choi, H. Pichler, & M. D. Lukin, (2020). Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices

[6] J. Preskill (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.

[7] S&P Dow Jones Indices, SPIVA U.S. Scorecard Year-End 2023. S&P Global, March 2024.

[8] E. F. Fama, (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. The Journal of Finance, 25(2), 383-417

[9] Beasley, J. E., Meade, N., & Chang, T. J. (2003). An evolutionary heuristic for the index tracking problem. European Journal of Operational Research, 148(3), 621-643

[10] Malkiel, B. G. (1973). A Random Walk Down Wall Street. W. W. Norton & Company.

[11] S. W. Hong, P. Miasnikof, R. Kwon e Y. Lawryshyn, Market Graph Clustering via QUBO and Digital Annealing, *Journal of Risk and Financial Management*, v. 14, n. 1, p. 34, Janeiro 2021.

[12] C. Bauckhage, N. Piatkowski, R. Sifa, D. Hecker e S. Wrobel, A QUBO Formulation of the k-Medoids Problem, in *Proceedings of the LWA 2019 Workshops: KDML, IR, FGWM*, Berlin, Germany, pp. 1-4, Outubro 2019.

[13] C. S. Mukherjee, S. Maitra, V. Gaurav, D. Roy, On Actual Preparation of Dicke State on a Quantum Computer, in *arXiv preprint arXiv:2007.01681*, 2020.

[14] Z. He, R. Shaydulin, S. Chakrabarti, D. Herman, C. Li, Y. Sun & M Pistoia Alignment between Initial State and Mixer Improves QAOA Performance for Constrained Optimization, in *npj quantum information*, 9, 121 (2023).