Exploring quantum optimization for solving the PCI planning problem in 5G networks

Full Paper

Adriano Borges
CENTRE FOR ADVANCED STUDIES AND SYSTEMS - CESAR School
Recife, Pernambuco, Brazil
aab3@cesar.school

Pamela Bezerra
CENTRE FOR ADVANCED STUDIES AND SYSTEMS - CESAR School
Recife, Pernambuco, Brazil
ptlb@cesar.school

Erico Teixeira
CENTRE FOR ADVANCED STUDIES AND SYSTEMS - CESAR School
Recife, Pernambuco, Brazil
est@cesar.school

ABSTRACT

Physical Cell IDs (PCIs) are numerical identifiers crucial for distinguishing various antennas, or cells, within telecommunication networks like 5G. They play a vital role in facilitating the efficient connection of mobile devices to different cells, preventing issues such as interference. However, the growing scale of 5G networks, coupled with a limited pool of unique PCIs, allocating different PCI to adjacent cells is a challenge known as the PCI planning problem.

In this scenario, this article explores the use of Quantum Computing (QC) to solve the PCI planning problem. With remarkable advancements in recent years, QC has shown great potential for solving complex optimization problems. To discern the advantages QC could bring to PCI planning, we analyzed the performance of classical and quantum methods across diverse network configurations. Our results show that quantum methods yield solutions equivalent to exhaustive search but with substantially reduced execution time, opening new research opportunities in QC and telecommunications.

1 INTRODUCTION

Mobile phones can transmit and receive data by connecting to antennas, also known as cells, at specific frequencies. These cells are often distributed among different telecommunications towers and are identified through a number known as Physical Cell ID (PCI). It is essential to assign distinct PCI values to nearby cells to mitigate Inter-Cell Interference (ICI), i.e., using the same frequency band by adjacent cells. An efficient PCI allocation provides a high-quality communication service to many users, avoiding, for example, long cell allocation time.

However, the number of available PCIs is limited to only 1008 in current 5G networks, making the efficient allocation of these identifiers challenging, especially with these networks' growing density. This problem, known as the **PCI planning problem**, is an NP-complete combinatorial optimization problem, with recent works in the area proposing the use of heuristics, such as reuse distance [9] and Glowworm swarm optimization (GSO)[11]. For example, commercial tools such as Atoll by Forsk use Monte Carlo simulation [6] to solve the PCI planning problem.

The innovative work of Gui et al. [7], for example, proposes a new combinatorial optimization model to describe collision, confusion, and *mod q* interference comprehensively and quantitatively. The PCI planning problem was mapped as a Binary Quadratic Programming (BQP) model, and a Greedy algorithm was developed to configure PCIs automatically for each cell in the whole network. To evaluate the optimization performance of the proposed algorithm, numerical simulations were performed compared with the scheme implemented in the current network and the classical graph coloring algorithm. The experimental results demonstrated that the Greedy algorithm had a significant advantage in reducing the collision, confusion, and *mod* 3 interference in scenarios using 1131 cells and 30 PCI. The Greedy algorithm not only eliminates conflict and confusion completely but also reduces the mod 3 interference by 26.213% more than the baseline scheme and far more than the improvement ratio of 4.436% given by the classical graph coloring algorithm.

On the other hand, Quantum Computing (QC) has seen rapid advancement in recent years, with companies like IBM and D-Wave launching new computers almost every year and an estimated investment of US$ 38.6 billion worldwide in just 2023[1]. Among the various application possibilities, QC has excellent potential for solving combinatorial optimization problems, such as PCI planning, due to using quantum mechanical phenomena, such as superposition. This phenomenon implies the main difference between classical and quantum computers: while the former use the bit as the basic unit of information, which can be 0 or 1, quantum computers use the quantum bit (qubit), a linear combination of the base states 0 and 1, allowing more information processing with fewer units.

The first work to propose using QC in PCI planning is Boella et al. Using a simpler Quadratic Unconstrained Binary Optimization (QUBO) formulation, [4] executed experiments using the quantum computer from D-Wave to solve a PCI planning. QUBO is a mathematical representation that provides a powerful tool for formulating and solving certain types of problems in computer science, particularly those that are NP-hard. The formulation was applied to the PCI planning of 5G and 4G and compared to the legacy procedure, Fast Greedy Algorithm. To analyze the algorithm's potential, a series of tests using a sample set of 450 cells of the TIM network were performed, decreasing the number of Secondary Synchronization Signal (SSS) compared to the maximum. In fact, as the number of SSS decreases, the probability of violating the constraints increases.

[1] https://qureca.com/overview-of-quantum-initiatives-worldwide-2023/

In light of the aforementioned considerations, this paper further explores the potential of QC in solving the PCI planning problem. Building upon the formulation introduced by [7], we have devised classical and quantum algorithms to scrutinize their performance across diverse network configurations characterized by varying numbers of cells and PCI. The experiments were also performed using a D-Wave system quantum annealing-based computer.

To the best of our knowledge, this study is one of the pioneering endeavors to investigate the PCI planning problem within quantum machines comprehensively. Our investigation meticulously examines the influences of different parameters on the execution time and the quality of the obtained results (i.e., how close to the optimal value), thus contributing novel insights to the evolving landscape of quantum computing applications in telecommunications planning.

We performed experiments using three methods: exact brute force, Steepest Descent heuristics, and the hybrid CQM method. We systematically varied the number of cells and distributed PCIs across 66, exploring diverse cell arrangements and yielding 64 unique scenarios for each method. Our experiments show clear empirical evidence in favor of quantum computing solutions when comparing their performance with the exact methods and heuristics.

This paper is organized as follows: **Section 2** details the PCI planning problem; **Section 3** explains the QUBO model required to solve the problem; **Section 4** describes the experimental environment and algorithms used, while **Section 5** discuss the obtained results; finally, **Section 6** summarise the main conclusions and futures works of this research.

2 THE PCI PLANNING PROBLEM

The PCI is crucial in assisting User Equipment (UEs - the technical term for mobile devices) in identifying which cell to connect to among various signals within the same frequency. It comprises two parts: Primary Synchronization Signal (PSS), which can have values of 0, 1, or 2, and Secondary Synchronization Signal (SSS), with values ranging from 0 to 355. These elements combine to form the PCI using the formula $PCI = 3 * SSS + PSS$. In the 5G context, this results in a total of 1008 possible PCI values, allowing for reuse when conducting PCI planning in scenarios with a higher number of cells.

Due to the finite number of available PCIs, reuse becomes inevitable in networks with a cell count exceeding 1008. However, the wrong allocation of PCI will significantly increase the occurrence of ICI. To mitigate these ICIs effectively, it is imperative to thoroughly examine scenarios involving collisions, confusion, and *mod q* interference within neighboring cells (i.e., adjacent cells less than 1 km apart) operating on the same frequency. This comprehensive analysis is pivotal for optimizing PCI allocation strategies and enhancing overall network performance.

In regular use, as described by Figure 1 (a), UE navigates from one cell to another with different PCI. Collision may occur if some neighboring cells have the same frequency and PCI, Figure 1 (b). In this case, it is difficult for the UE to select which cell to address, as there are two different cells with the same PCI.

Confusion may occur with two or more neighbor cells sharing the same frequency and PCI, shown in Figure 1 (c), when the UE leaves a cell with a PCI and goes to a region with two cells with the same PCI.

Similar to the collision scenario, *mod q* interference may occur in some neighboring cells with the same frequency, as shown in Figure 1 (d); the PCI *mod q* value of one cell is equal to the PCI *mod q* value of other cells.

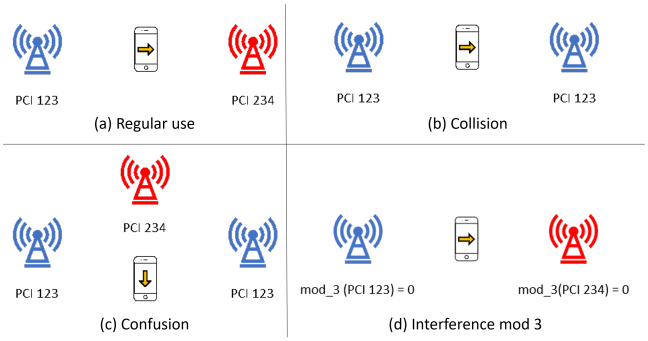

Figure 1: Types of Interferences

3 OPTIMIZATION MODEL

The Quadratic Unconstrained Binary Optimization (QUBO) model is applied to solve combinatorial optimization problems where the goal is to find the optimal binary values (0 or 1) for a set of variables to minimize a quadratic objective function. The optimize function can be formally defined by Equation 1:

$$\min F(x) = \sum_{i<j}^{N} Q_{i,j} x_i x_j + \sum_{i}^{N} Q_{i,i} x_i \tag{1}$$

, where Q is a NxN triangular matrix with real values, with the diagonal representing the linear weight terms, the off-diagonal the quadratic weight terms, and x is a vector of the binary variables.

This is an important formulation because real-world problems can be naturally transformed into QUBO form by providing a way to represent classical optimization problems as quantum problems. In addition, quantum annealer computers, like those offered by D-Wave[2], are designed to solve QUBO problems efficiently.

The QUBO formulation used in this work to address the PCI assignment problem was the same as that presented by Gui et al. [7], where the objective is to minimize collisions, confusion, and *mod q* interference.

This work assigns one binary variable for each pair composed by a cell i to PCI k:

$$x_{i,k} = \begin{cases} 1 & \text{if cell } i \text{ is associated to PCI } k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

For this formulation, a matrix A of dimensions $n \times n$, where n is the number of cells, is constructed, and each element $a_{i,j}$ represents the relationship between the cell in row i and the cell in column j. The value 1 will be assigned when they are neighbors and 0

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

otherwise. Furthermore, the value zero will be associated with the main diagonal.

In the same way, another matrix B is created, also of dimensions $n \times n$, which represents the state of confusion between the cells. Each element $b_{i,j}$ of B represents the relationship between the cell in row i and the cell in column j. The value 1 will be assigned when they are neighbors due to confusion, and 0 otherwise.

Finally, the matrix L of dimension $m \times q$ calculate *mod q* interferences of m PCIs. Each element $l_{i,j}$ indicates if the i-th PCI has *mod q* equal to j. The general expression of L is:

$$L = [\underbrace{I_q, I_q, \ldots, I_q}_{m/q\ items}]^T \tag{3}$$

, where I_q is the $q \times q$ identity matrix.

Thus, the QUBO model for PCI planning is described by the objective function

$$\begin{aligned} \min_{i,j} F = {} & \omega_1 \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} \sum_{k=1}^{m} x_{i,k}x_{j,k} \\ & + \omega_2 \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij} \sum_{k=1}^{m} x_{i,k}x_{j,k} \\ & + \omega_3 \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} \sum_{h=1}^{q} (\sum_{k=1}^{m} x_{i,k}l_{k,h}).(\sum_{k=1}^{m} x_{j,k}l_{k,h}) \\ \text{s.t.} \sum_{k=1}^{m} x_{i,k} = 1, \forall i \rightarrow & \sum_{i}^{n} \left(\sum_{k=1}^{m} x_{i,k} - 1 \right)^2 \\ x_{i,k} \in \{0,1\}, \forall i,k & \end{aligned} \tag{4}$$

with the constraint

$$\sum_{k=1}^{m} x_{i,k} = 1, \forall i \rightarrow \sum_{i}^{n} \left(\sum_{k=1}^{m} x_{i,k} - 1 \right)^2 \tag{5}$$

where ω_1, ω_2 and ω_3 are weighting factors, respectively, for the condition of collision, confusion, and *mod q* interference, n is the total of cells, m the number of PCI to be distributed, and the constraint that associates only one PCI to each cell leads to a penalty function to be added to the QUBO formulation.

4 ENVIRONMENT AND ALGORITHMS

As previously mentioned, PCI planning is an NP-complete combinatorial optimization problem, and several studies have proposed heuristics methods and other classical (i.e., non-quantum) strategies to solve it efficiently [9][11].

In our analysis, we used two classical optimization methods:

(1) **Exhaustive Search**, which is the easiest method for finding optimization solutions by testing all possible solutions. Although this method can always find the best solution, its use is not practical for large problems due to the exponential number of cases to be tested (combinatorial explosion). This study used the exhaustive search in small scenarios as a benchmark for the solution's value for the heuristic and quantum methods.
(2) **Gradient Descent**, a simple iterative heuristic that gradually moves toward a minimal local function based on its gradient. The gradient of any differentiable function represents how quickly it moves towards a local minimum. Therefore, the gradient descent method moves in the opposite direction, always selecting the largest step. For our analysis, this method was used as a benchmark for the algorithm execution time for the exhaustive and quantum methods.

4.1 Adiabatic Quantum Computing

Adiabatic Quantum Computing (AQC) [2] stands as a fundamental paradigm for quantum computation, drawing upon the adiabatic theorem [5] in quantum mechanics. According to this theorem, a quantum system will remain in its ground state if the associated Hamiltonian changes sufficiently slowly. In broad strokes, AQC commences with a simple Hamiltonian ($\hat{H}_0$) whose ground state is readily preparable. Over time, the Hamiltonian is smoothly changed to represent the problem Hamiltonian ($\hat{H}_f$) wherein the ground state encapsulates the solution to a computational problem. A parameterized time-dependent Hamiltonian describes the algorithms associated with AQC (Equation 6)

$$\hat{H} = A(t)\hat{H}_0 + B(t)\hat{H}_f \tag{6}$$

where $t \in [0,1]$ and $A(t)$ and $B(t)$ are the functions that describe the interpolation between the Hamiltonians and that obey the generic boundary conditions given by

$$A(0) \neq 0, \quad B(1) \neq 0, \quad A(1) = B(0) = 0 \tag{7}$$

As the system undergoes evolution, $A(t)$ gradually decreases while $B(t)$ increases until, ultimately, the total Hamiltonian is solely defined by the term associated with $B(t)$. If the process is slow enough -achieving adiabatic conditions- the resultant state will correspond to the ground state of the final Hamiltonian of the system, which encodes a solution to the problem.

4.2 Quantum Annealing

Quantum annealing (QA) [10] represents a specific implementation of AQC, with the time evolution of the quantum system drawing inspiration from the classical annealing process in metallurgy, where a material is heated and slowly cooled to remove defects and optimize its structure. QA is a heuristic quantum approximation because the switch from $\hat{H}_0$ to $\hat{H}_f$ is determined heuristically, and the adiabatic conditions are not guaranteed. As a result, QA is particularly well-suited for encoding binary combinatorial optimization problems, expressed in Ising or QUBO form. These two representations are equivalent and can be readily transformed into each other through a simple change of basis. This flexibility makes QA a valuable approach for tackling a broad class of optimization challenges in quantum computing.

One of the most popular, widely used QA devices is the D-Wave System, featuring a quantum processor with a set of superconducting qubits arranged in a configuration analogous to a chain of Ising-type magnetic spins. In this platform, $\hat{H}_0$ is constructed by applying transverse magnetic fields, aligning the spins (qubits) in the direction of the field. The adiabatic interpolation (process $\hat{H}_0 \rightarrow \hat{H}_f$) unfolds by slowly reducing the intensity of the transverse field to zero. Simultaneously, the intensity of the couplings

between the qubits increases, facilitating the transition from the initial state to the final Hamiltonian.

5 COMPUTATIONAL EXPERIMENTS

As previously mentioned (Section 4.2), D-Wave Systems is the market leader in QA devices. Founded in 1999 in Canada, the company developed "the world's first commercially available quantum computer" with 128 qubits in 2011 [8]. Currently, its more powerful device, the Advantage[3], has 5000 qubits and an optimized topology, being capable of solving complex commercial problems with more than 1 million variables. Over the years, the company has accumulated more than 200 patents and 100 research publications in various areas, such as logistics and financial services.

D-Wave is not only a pioneer in quantum hardware but also in software and services. It developed its own Python-based open-source software development kit (SDK), the D-Wave Ocean [4]. Additionally, the company provides a cloud service, the D-Wave Leap[5], for real-time remote access to its devices. The experiments reported in this article were performed directly through the Leap interface.

The essential elements of a D-Wave solution are the *samplers* and *solvers*. A sampler is a process that samples low-energy states from objective functions to find the best solution. It takes a problem formulated as QUBO and returns a set of potential solutions represented as binary assignments (0s and 1s) for the variables in the model. These samplers run on a device known as a solver, which can be mainly classified into three types: (1) Classical (i.e., a classic computer), (2) Quantum (i.e., a QA computer), and (3) Hybrid, an architecture that explores the advantages of both types of resources (classical and quantum). In our experimental analyses, we've used two classical solvers, **Exact Solver** and **Steepest Decent**, which implement, respectively, the exhaustive search and a discrete version of gradient descent, and a Hybrid Solver Service (HSS) known as **Constrained Quadratic Model** (CQM).

5.1 Constrained Quadratic Models

CQM involves linear constraints, i.e., the constraint does not need to be translated into a penalty function. Thus, the restriction that associates only one PCI for each cell presented in Equation 5 can be used without the need to rewrite it as a penalty. In this framework, besides the quantum stage, a classical pre-processing involves generating an initial solution with a classical heuristic to the CQM that can be used as a starting point for the quantum annealing algorithm, potentially reducing the time required to find a good solution. The starting point will depend on the classic algorithm used in the heuristic. In our experiment, we used the default D-Wave configuration, which corresponds to Simulated Annealing.

Figure 2 illustrates how HSS CQM works [1]. The solver (blue) invokes some heuristics (threads) that run on classic CPUs and GPUs (green) and searches for good-quality starting point solutions. Each heuristic solver contains a quantum module (QM) that formulates and sends quantum queries to a D-Wave QPU (orange). The responses to these queries can guide heuristic search or improve the quality of a current set of solutions. In the end, each heuristic sends its best solutions to the solver.

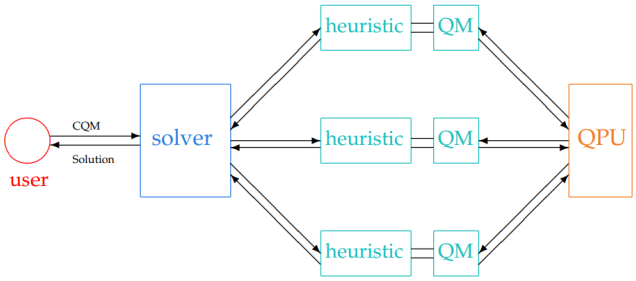

Figure 2: How CQM works at D-Wave.

5.2 The problem instances

Were created two groups of instances:

(1) **Synthetic Instances** - cell arrangements generated synthetically to simulate collision, confusion and mod q interference.
(2) **Real Instances** - cell arrangements based on real world datasets.

Each algorithm was expected to be tested through 64 instances (Qtty Exp), with varying cell neighborhood arrangements (Ins) and the quantity of PCI available for allocation (Qtty PCI), according to Table 1. Synthetic instances were arranged in four ways: a 5-cell instance, named X5, Figure 3, with matrix A configured in chess pattern, i.e., cells 1, 3, and 5 neighbors of cells 2 and 4; and 5, 6, and 9-cell instances, with all cells being neighbors between to each other, respectively named V5, Figure 4, V6, and V9. Real instances were created using public information of 5G network cells available from a telecommunications governmental agency in Brazil[6]. The circles in Figure 5 define the groups with 15, 27, 48, and 66 cells named R15 (yellow circle with a 0.5 km radius), R27 (white circle with a 0.7 km radius), R48 (red circle with a 0.98 km radius), and R66 (blue circle with a 1.25 km radius), respectively. Cells that are less than 1 km apart are considered neighbors. In Figures 3 and 4, who represent neighborhood matrix A, value 0, zero, indicates that the cell from that line is not neighbor to the cell in that column, the same way that the value 1, one, means that they are neighbors.

0	1	0	1	0
1	0	1	0	1
0	1	0	1	0
1	0	1	0	1
0	1	0	1	0

Figure 3: Instance X5, five cells as a chess pattern.

Initially, it was decided to assign the same weight to all penalties, that is, the factors ω_1, ω_2, and ω_3 will have values equal to 1. This promotes a better understanding of each algorithm, allowing them to find solutions that satisfy all constraints in a balanced way. In

[3]https://www.dwavesys.com/solutions-and-products/systems/
[4]https://www.dwavesys.com/solutions-and-products/ocean/
[5]https://www.dwavesys.com/solutions-and-products/cloud-platform/
[6]https://sistemas.anatel.gov.br/se/public/view/b/licenciamento.php

0	1	1	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Figure 4: Instance V5, five cells all neighbors.

Table 1: Instances Distribution. Qtty PCI and Exp describe, respectively, the number of PCI and experiments per instance

Ins	Type	Qtty PCI	Qtty Exp
X5	Synth	1 to 5	5
V5	Synth	1 to 5	5
V6	Synth	1 to 6	6
V9	Synth	1 to 9	9
R15	Real	1 to 15	15
R27	Real	1,2,3,6,9,15,27	7
R48	Real	1,2,3,6,9,15,27,48	8
R66	Real	1,2,3,6,9,15,27,48,66	9
		TOTAL	64

Figure 5: Real instances, showing cell arrangements with 15 cells (yellow circle), 27 cell (white circle), 48 cell (red circle) and 66 cell (blue circle).

this scenario, of the three algorithms tested, the exact solver was the slowest to complete each experiment. Furthermore, the fact of having to traverse the entire search space made the execution of some scenarios unfeasible. Thus, the algorithm failed to generate results in 33 of the 64 predicted scenarios.

Table 2 shows the energy (i.e., the final Hamiltonian) of exact (EX), steepest descent (SD), and CQM methods to some of the synthetic instances simulated. Table 3 shows the energy of steepest descent (SD) and CQM methods for some real instances simulated. It can be seen that the exhaustive and CQM methods always find the lowest energy when compared to SD.

Figures 6, 7 and 8 show the execution time for V9 instances applying the exact algorithm, SD and CQM, respectively. It can be seen that SD has the shortest execution time, but, as highlighted in Table 2, this method does not always present the optimal solution (e.g, instances V5 with 4 and 5 PCI). This behavior is repeated in all other instances. These graphs also demonstrate the more rapid growth of the exhaustive search in comparison to other methods.

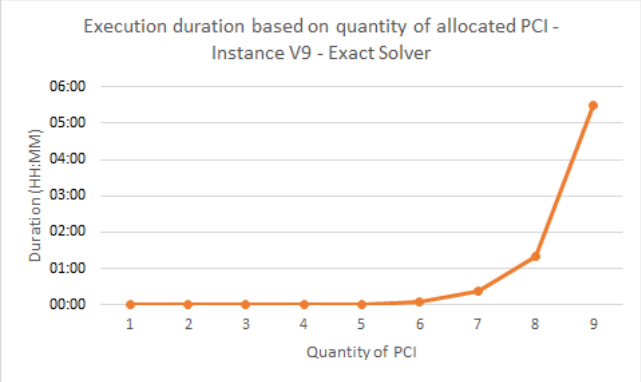

Figure 6: Execution duration based on the quantity of allocated PCI - Instance V9 - Exact Solver.

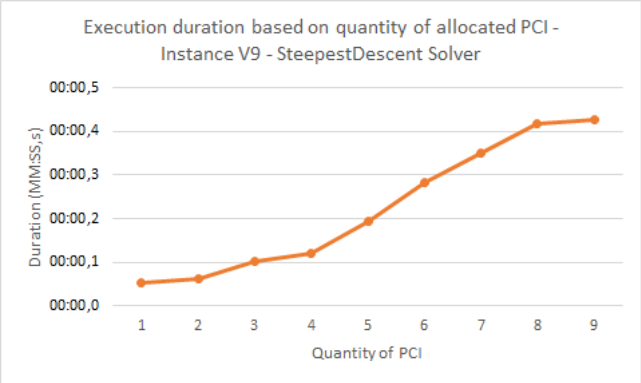

Figure 7: Execution duration based on the quantity of allocated PCI - Instance V9 - SteepestDescent Solver.

6 CONCLUSION

This paper explored the use of quantum computing to better solve the Physical Cell ID (PCI) planning problem. PCI planning is essential for minimizing the impacts of Inter-Cell Interference (ICI) in 5G Networks, such as collision, confusion, and *mod q* interference. Due to its complexity, this problem is usually solved with heuristics.

Our results show a clear advantage of quantum computing. CQM, the hybrid quantum-classical method, found solutions equivalent to the exact solver, but faster. In addition, besides the steepest descent has the shortest run, it does not produce optimal results.

As a future work, we plan to enhance our comparative analysis by exploring D-Wave Quantum Solvers, such as the Binary Quadratic Model (BQM)[7], other classic heuristics, such as Simulated

[7]https://docs.ocean.dwavesys.com/en/stable/concepts/bqm.html

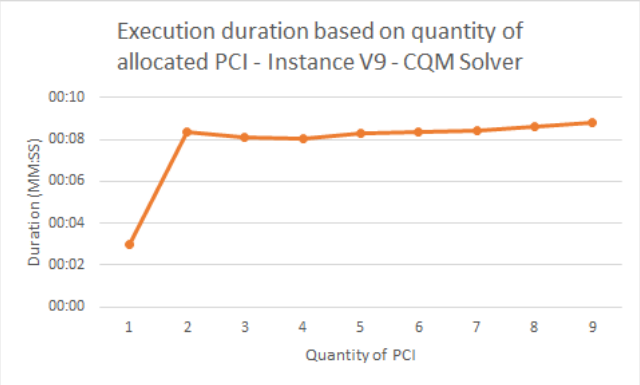

Figure 8: Execution duration based on the quantity of allocated PCI - Instance V9 - CQM Solver.

Table 2: Energy of Synthetic Instances

Ins	PCI	EX	SD	CQM
X5	1	32	32	32
X5	2	8	8	8
X5	3	4	4	4
X5	4	2	2	2
X5	5	0	0	0
V5	1	60	60	60
V5	2	24	24	24
V5	3	12	12	12
V5	4	8	**12**	8
V5	5	4	**8**	4
V6	1	90	90	90
V6	2	36	36	36
V6	3	18	18	18
V6	4	14	14	14
V6	5	10	10	10
V6	6	6	**12**	6
V9	1	216	216	216
V9	2	96	96	96
V9	3	54	54	54
V9	4	44	**48**	44
V9	5	36	36	36
V9	6	30	30	30
V9	7	26	**28**	26
V9	8	22	**30**	22
V9	9	18	**22**	18

Annealing, Tabu Search, and better exact algorithms for solving QUBO problems [3]. Additionally, we want to extend our current analysis beyond Quantum Annealing systems, testing, for example, gate-based systems such as the ones provided by IBM[8].

[8]https://www.ibm.com/quantum

Table 3: Energy of Real Instances

Ins	PCI	SD	CQM	Ins	PCI	SD	CQM
R15	2	266	242	R27	3	566	468
R15	3	148	134	R27	6	336	258
R15	4	124	98	R27	9	222	180
R15	5	104	66	R27	15	188	154
R15	6	84	62	R27	27	152	126
R15	7	70	58	R48	3	1392	1266
R15	8	56	54	R48	6	852	682
R15	9	54	50	R48	9	616	488
R15	10	50	46	R48	15	420	378
R15	11	62	44	R48	27	354	312
R15	12	62	42	R66	3	1896	1824
R15	13	44	40	R66	6	1220	920
R15	14	44	38	R66	9	910	650
R15	15	42	38	R66	15	614	468

ACKNOWLEDGMENTS

The authors would like to thank the Center of Advances Studies and Systems (CESAR) for investing in the Quantum Applications in Technology and Software (|QATS)), a research group that the authors are members, enabling access to the D-Wave computers as well as other relevant materials for developing this research paper.

REFERENCES

[1] 2022. Hybrid Solvers for Quadratic Optimization. https://api.semanticscholar.org/CorpusID:251834718
[2] Tameem Albash and Daniel A. Lidar. 2018. Adiabatic quantum computation. *Reviews of Modern Physics* 90, 1 (jan 2018), 015002. https://doi.org/10.1103/RevModPhys.90.015002 arXiv:arXiv:1611.04471v2
[3] Alain Billionnet and Sourour Elloumi. 2007. Using a Mixed Integer Quadratic Programming Solver for the Unconstrained Quadratic 0-1 Problem. *Mathematical Programming* 109, 1 (2007), 55–68. https://doi.org/10.1007/s10107-005-0637-9
[4] Andrea Boella, Michele Ludovico, Giuseppe Minerva, and Mauro Alberto Rossotto. 2021. *Application of Quantum Computing to Cell-ID Planning of 4.5G and 5G Networks (TIM).* Technical Report. 42–45 pages. https://www.gsma.com/newsroom/wp-content/uploads/IG-11-Quantum-Computing-Networking-and-Security.pdf
[5] M Born and V Fock. 1928. Beweis des Adiabatensatzes. *Zeitschrift für Physik* 51 (1928), 165–180. https://doi.org/10.1007/BF01343193
[6] Forsk. 2018. *Atoll - Wireless Network Engineering Software - version 3.4.* Retrieved November 27, 2023 from https://www.forsk.com/sites/default/files/atoll_34_sept2018-light_0.pdf
[7] Jihong Gui, Zhipeng Jiang, and Suixiang Gao. 2018. PCI Planning Based on Binary Quadratic Programming in LTE/LTE-A Networks. *IEEE Access* 7 (2018), 203–214. https://doi.org/10.1109/ACCESS.2018.2885313
[8] Mark W Johnson, Mohammad HS Amin, Suzanne Gildert, Trevor Lanting, Firas Hamze, Neil Dickson, Richard Harris, Andrew J Berkley, Jan Johansson, Paul Bunyk, et al. 2011. Quantum annealing with manufactured spins. *Nature* 473, 7346 (2011), 194–198.
[9] Anggita Putri Lestari, A Nachwan Mufti, and Uke Kurniawan Usman. 2018. Optimization of PCI conflict detection in LTE-advanced using collision and confusion methods considering reuse distance algorithm. In *2018 International Conference on Signals and Systems (ICSigSys).* IEEE, 20–24.
[10] Satoshi Morita and Hidetoshi Nishimori. 2008. Mathematical foundation of quantum annealing. *J. Math. Phys.* 49, 12 (dec 2008), 125210. https://doi.org/10.1063/1.2995837 arXiv:0806.1859
[11] Cheng-fu Zhang, Navrati Saxena, Fan-guang Kong, and Jian-hua He. 2013. Self-Configuration of Physical Cell Identity in LTE Based on Improved GSO Algorithm. In *Proceedings of 20th International Conference on Industrial Engineering and Engineering Management: Theory and Apply of Industrial Engineering.* Springer, 377–387.