UNIVERSIDAD EAFIT

Vigilada Mineducación

USING QUANTUM COMPUTING TO SOLVE THE MAXIMAL COVERING LOCATION PROBLEM

(Computación cuántica para la solución del problema de la máxima cobertura)

ALEJANDRO GIRALDO QUINTERO

Tesis de Maestría

Asesor

Juan Guillermo Lalinde Pulido

Co-asesores:

Daniel Sierra Sosa

Juan Carlos Duque Cardona

UNIVERSIDAD EAFIT

ESCUELA DE CIENCIAS

MAESTRÍA EN MATEMÁTICAS APLICADAS

MEDELLÍN

2022

Using quantum computing to solve the Maximal Covering Location Problem.

Alejandro Giraldo-Quintero[1,2], Juan G. Lalinde-Pulido[1], Juan C. Duque[3*], and Daniel Sierra-Sosa[4]

[1]EAFIT University, Colombia.
[2]Corporacion Estudiantes Universitarios y Profesionales de Marinilla (CORUM), Colombia
[3]Research in Spatial Economics (RiSE) Group, Department of Mathematical Sciences, EAFIT University, Colombia.
[4]Department of Computer Science & Information Technology, Hood College, USA.
[*]corresponding author: jduquec1@eafit.edu.co, +57(4)261-9500

January 2022

Acknowledgments

We want to extend our gratitude to the Apolo Scientific Computing Center at EAFIT University, which allowed us to significantly reduce the computational processes involved in this research. Juan C. Duque thanks the PEAK Urban Programme for their support. This work was supported by the Global Challenge Research Fund from UKRI, UK, Grant Ref: ES/P011055/1.

Using quantum computing to solve the Maximal Covering Location Problem.

Abstract

In this article, we present the process and results of using quantum computing (QC) to solve the maximal covering location problem proposed by Church and ReVelle. With this contribution, we seek to lay the foundations for other regional scientists to begin to consider quantum technologies. We obtained promising results, but it is clear that there is a need for more capable devices with more qubits and less susceptible to electronic noise to solve instances that currently cannot be optimally solved by traditional solvers. We foresee that QC will be of common use in regional science and its applications in the years to come.

Keywords

Maximal Covering Location Problem · Quantum Computing

JEL Classification

C61 · C63 · R3 · R53

1 Introduction

Theoretical quantum computing was first proposed in the early 1980s by Yuri Manin, Richard Feynman, and Paul Benioff as a new way to leverage quantum phenomena to process information (Hidary, 2019). Later, in 1985, David Deutsch formalized the concept of quantum computing by describing the first quantum Turing machine (Deutsch, 1985), which laid the foundations for the emergence of quantum algorithms (e.g., Deutsch and Jozsa, 1992; Simon, 1997; Grover, 1996). Additionally, Shor's algorithm, a polynomial-time quantum computer algorithm for integer factorization devised by Shor (1995), with the capacity to break the Rivest–Shamir–Adleman

(RSA) encryption system, was one of the great motivators for the development of quantum computers. A few years later, in 1998, Chuang et al. (1998) presented the first complete experimental implementation of Grover's search algorithm in a quantum computer, and in 2001, IBM and Stanford University factorized 15 in a 7-qubits quantum computer using Shor's algorithm (IBM, 2001). In 2011, D-Wave Systems Inc. (Merali, 2011) and IBM in 2016 (Santos, 2016) were the first companies to develop commercially available quantum computers.

Despite their huge capabilities, quantum computers are not a replacement for classical computers. The fundamental principles of superposition and entanglement make quantum computers faster for calculations in which one can take advantage of simultaneously evaluating multiple solutions (computational parallelism). This characteristic makes quantum computing a good option for solving NP-hard and NP-complete combinatorial problems. Today, quantum computing has applications in cryptography (Fernández-Carames and Fraga-Lamas, 2020; Mavroeidis et al., 2018), computer-aided drug design (Cao et al., 2018; Outeiral et al., 2021), finance (Egger et al., 2020; Orus et al., 2019), artificial intelligence (Choi et al., 2020; Abdelgaber and Nikolopoulos, 2020), and optimization (Harwood et al., 2021).

In this paper, we seek to contribute to the insertion of quantum computing in regional science. For this, we employ one of the best-known NP-hard problems in regional science, the maximal covering location problem (MCLP) proposed by Church and ReVelle (1974). The MCLP seeks to maximize the population covered within a desired service distance S by locating a fixed number of facilities. We start by taking the reader through the nontrivial process that begins with the transformation of the binary integer programming formulation of the MCLP to its solution in the two currently available quantum computing paradigms: adiabatic quantum computing (D-Wave) and gate-based quantum computing (IBM). We perform a computational experiment on real quantum devices and classical quantum simulators. Our main objective is to show regional scientists the potential of this technology and to point out the steps of the process in which we can contribute without expertise in quantum computing.

This article is not a simple application of quantum computing. As will be seen later, the transformation of an integer binary programming formulation into the expression required to be processed by a quantum computer is an important contribution to the state-of-the-art methods in this area. Although quantum computing is in its early stages of development, our contribution will motivate regional scientists to be better prepared to take advantage of a technology that will soon allow us to solve various optimization problems in regional science regardless of their size.

The remainder of the paper is presented as follows: the next section contains a literature review on quantum computing and the MCLP. The third section contains the general steps for solving an integer programming problem with a quantum computer. The fourth and fifth sections present a computational experiment that solves small instances of the MCLP using both the adiabatic quantum computing and the gate-based quantum computing paradigms in quantum computing. Finally, the sixth section presents the conclusions and possible future lines of research.

2 Literature Review

2.1 Quantum Computing

Quantum computation and quantum information address the study of information processing tasks that can be accomplished by using quantum mechanical systems (Nielsen and Chuang, 2010). This area of study started almost simultaneously with quantum theory, however, it was only in 1980, with the no-cloning theorem that the first relevant steps in this area were taken. Benioff (Benioff, 1980) and Feynman (Feynman, 1960, 1982) proposed the idea of making quantum mechanical simulations with the help of quantum phenomena. Later, this idea was formalized by David Deutsch in the form of quantum Turing machines (Deutsch, 1985) and quantum gates (Deutsch, 1989). The first quantum algorithms were the Deutsch-Jozsa (Deutsch and Jozsa, 1992), Grover (Grover, 1996) and Simon algorithms (Simon, 1997). In 1994, Shor's algorithm highlighted the possibilities of quantum computing in the factorization of prime numbers, which is an essential part of the worldwide RSA encryption system, and increased the interest of the scientific community in this area.

Although there are several ways in which quantum mechanics can be used in quantum computing, by far the most dominant paradigms are quantum annealing (de Falco et al., 1988) and gate-based quantum computing, also known as quantum circuits (Deutsch, 1985). Quantum annealing is based on the natural behavior of a quantum system (i.e., a network of qubits). It begins by mapping the problem into a quantum system that evolves naturally until it reaches its minimum energy level that corresponds to the optimal solution to the original optimization problem. In gate-based quantum computing, the problem is mapped onto a circuit containing qubits and quantum logic gates (Scherer, 2019). The gates perform specific transformations to the qubits until they converge to a state that corresponds to the optimal solution to the original optimization problem.

Quantum computing is a young field, and although its impact on regional science is still low, it holds a lot of promise. The areas of transport, location, and region design, to name a few, have NP-hard and NP-complete combinatorial problems that are excellent candidates for quantum computing solutions. To the best of our knowledge, Harwood et al. (2021) is the only publication that uses quantum computing to solve a problem within the area of regional science, namely, the routing problem. Harwood et al. (2021) use the IBM quantum simulators, which is a resource offered by IBM for prototyping quantum circuits and algorithms, to solve an instance with three demand nodes and with time windows originally solved in Desrochers et al. (1992).

Recently, great advances have been made in terms of the accessibility and usability of quantum computers. On one hand, python libraries have been developed, such as DOcplex (IBM, 2021), which help in the process of converting a classical optimization formulation to the quadratic unconstrained binary optimization formulation required to interact with a quantum computer (Glover et al., 2019). On the other hand, libraries have also been created in python to interact with quantum computers such as that of IBM (in the case of gate-based quantum computers) and D wave (in the case of quantum annealing computers). IBM developed the Qiskit framework that allows the simulation of quantum circuits that can then be sent to real IBM quantum computers (Santos, 2016). Amazon created Amazon Braket, a library that allows simulation and interaction with different types of quantum computers, including those of D-wave (Gonzalez, 2021).

As this article is addressed to nonexperts in quantum computing, we consider it important to briefly describe the basis of quantum computing. A classical computer performs operations using bits, which can be either one or zero. In contrast, a quantum computer uses qubits (a photon, an electron, or a nucleus), which, due to the property of superposition, can be in both states (one and zero) at the same time. Therefore, in a state of superposition, the qubit is expressed as a linear combination of two states, i.e., $|\psi\rangle = c_0|0\rangle + c_1|1\rangle, \quad c_0, c_1 \in \mathbb{C}$, where the coefficients c_0 and c_1 are complex numbers that describe how much goes into each state (Hidary, 2019). Note that if we have a classical bit, we just need a number (a piece of information) to know the state of the bit: zero or one. However, if we have a qubit, we need two pieces of information to describe its state: c_0 and c_1.

Now, let us suppose that we have two bits and two qubits. In both cases, we have four possible states: {0,0}, {0,1}, {1,0}, and {1,1}. If we want to describe the state of the two classical bits, all we need is two pieces of information: the state of the first bit and the state of the second bit (e.g., {1,0}). However, in the case of the two qubits, because of the property of superposition, {0,0},{0,1},{1,0}, and {1,1} exist at the same time. The four states are in superposition; therefore, we can write a quantum mechanical state that is a linear combination of the four basic states, i.e., : $|\psi\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle$. Thus, we need four numbers (i.e., c_{00}, c_{01}, c_{10}, and c_{11}) to determine the state of a system with two qubits. Therefore, a system with two qubits contains four bits of information. In general, a system with N qubits contains information equivalent to 2^N classical bits (Nielsen and Chuang, 2010).

Although all the states in a quantum system are in a superposition, when the system is measured (i.e., when a physical system is manipulated to yield a numerical result), all the states collapse into one of the basic states (e.g., {0,0}, {0,1}, {1,0}, or {1,1}) (Hardy and Steeb, 2001). Therefore, the big challenge in quantum computing is to design quantum systems in such a way that the final result is one of the basis states (i.e., something that can be measured).

Qubits in a quantum system can interact in a highly correlated manner, known as entanglement, a quantum phenomenon without a classical counterpart. When quantum states (qubits) are entangled, they cannot be described independently (Nielsen and Chuang, 2010). Therefore, entangled qubits lose their individuality and behave as a single entity (i.e., a change in one qubit leads to an immediate response in the others).

Finally, due to a phenomenon known as interference (Brassard et al., 1998), a quantum system can collapse into different results when observed. This makes quantum measurement a probabilistic process. Therefore, solving a problem with quantum technology requires that the problem be solved multiple times. The final solution will then be the one with the highest probability.

In summary, the quantum properties of superposition and entanglement make quantum computing extremely fast in problems in which parallelism can be taken advantage of. For example, in combinatorial optimization, one can use superposition to make an instant evaluation of all possible combinations. In other words, quantum computers allow one to perform an instant evaluation of the solution space.

2.2 The maximal covering location problem - MCLP

The maximal covering location problem (MCLP), devised by Church and ReVelle (1974), is a classic problem in location analysis that seeks to maximize the population covered within a desired service distance S by locating a fixed number of facilities. For the sake of completeness, we present the minimization version of the MCLP model.

$$\text{Minimize} \quad z = \sum_{i \in I} a_i \bar{y}_i \tag{1}$$

$$\text{S.T.} \quad \sum_{j \in N_i} x_j + \bar{y}_i \geq 1 \quad \text{for all } i \in I \tag{2}$$

$$\sum_{j \in J} x_j = p \tag{3}$$

$$x_j = (0,1) \text{ for all } j \in J$$

$$\bar{y}_i = (0,1) \text{ for all } i \in I$$

where

$I =$ denotes the set of demand nodes;

$J =$ denotes the set of facility sites;

$S =$ the distance beyond which a demand point is considered "uncovered" (the value of S can be chosen differently for each demand point if desired);

$d_{ij} =$ the shortest distance from node i to node j;

$$x_j = \begin{cases} 1 \text{ if a facility is allocated to site } j \\ 0 \text{ otherwise;} \end{cases}$$

$N_i = \{j \in J \mid d_{ij} \leq S\}$;

$a_i =$ population to be served at demand node i

$p =$ the number of facilities to be located.

$$\bar{y}_i = \begin{cases} 1 \text{ if a demand node } i \text{ is not covered by a facility within a } S \\ \text{distance} \\ 0 \text{ otherwise;} \end{cases}$$

Objective function (1) minimizes the number of people who will not be served within the maximal service distance. Constraint (2) establishes that a demand node i is not covered if there is not a facility within S distance. Constraint (3) sets the number of facilities allocated to p. Finally, all decision variables are binary.

The MCLP is related to a family of problems that are classified as nondeterministic polynomial-time complete (NP-complete) (Murray, 2016; Megiddo et al., 1983). To address this complexity, some authors have explored Lagrangian relaxation (Weaver and Church, 1986; Galvão and ReVelle, 1996), while others have proposed heuristic approaches (Murray and Church, 1996; Tong et al., 2009; Church and Murray, 2018). The contribution of our article to the MCLP literature consists of the exploration of the potential of quantum computing to solve this problem.

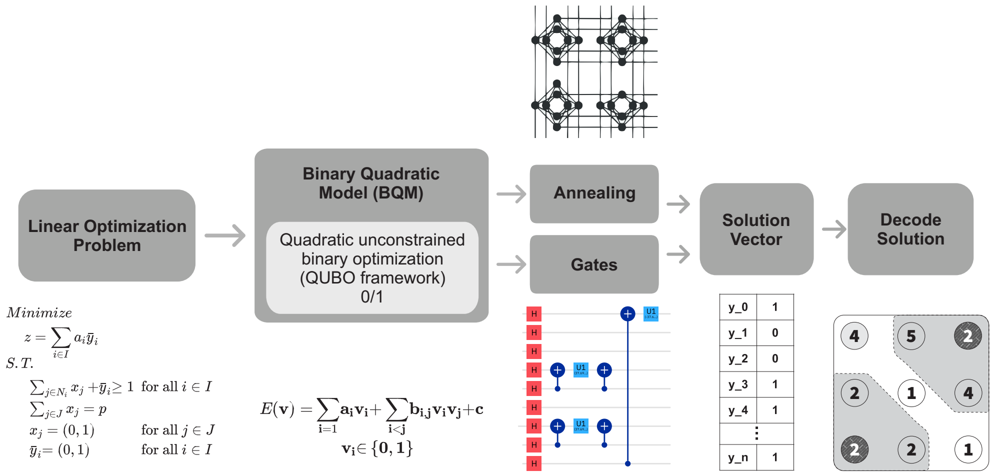


Figure 1: Main steps for solving an integer programming problem in a quantum computer

3 General steps for solving an integer programming problem in a quantum computer

The main steps for solving a linear optimization problem using a quantum computer are presented in Figure 1. In general, the process starts with a linear optimization problem (step one). Next, the problem is converted into a quadratic binary unconstrained optimization (QUBO) model, which is an expression with a linear component and a quadratic component. All variables are binary variables (step two). The QUBO expression can then be mapped onto a quantum gate or annealing computer (step three). Once the evolution of the quantum process is finished, the measurement is performed, and the solution vector is obtained (step four). Finally, the solution vector is decoded to obtain the final solution (step five). Each of these steps will be developed in more detail in the following subsections. We will use the MCLP instance presented in Figure 2 to exemplify the process (we set $p = 2$).

3.1 Linear optimization problem

The process starts with the formulation of a linear optimization problem such as the one presented in equation 4; this formulation represents the optimization MCLP problem of the instance presented in Figure 2:

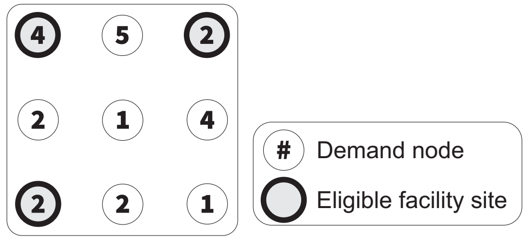


Figure 2: Example of an MCLP instance

$$
\begin{aligned}
&\text{Minimize} \\
&\qquad 4\bar{y}_0 + 5\bar{y}_1 + 2\bar{y}_2 + 2\bar{y}_3 + \bar{y}_4 + 4\bar{y}_5 + 2\bar{y}_6 + 2\bar{y}_7 + \bar{y}_8 \\
&\text{Subject to} \\
&\qquad C_1: \quad \bar{y}_0 + x_0 \geq 1 \\
&\qquad C_2: \quad \bar{y}_1 + x_0 + x_2 \geq 1 \\
&\qquad C_3: \quad \bar{y}_2 + x_2 \geq 1 \\
&\qquad C_4: \quad \bar{y}_3 + x_0 + x_6 \geq 1 \\
&\qquad C_5: \quad \bar{y}_4 \geq 1 \\
&\qquad C_6: \quad \bar{y}_5 + x_2 \geq 1 \\
&\qquad C_7: \quad \bar{y}_6 + x_6 \geq 1 \\
&\qquad C_8: \quad \bar{y}_7 + x_6 \geq 1 \\
&\qquad C_9: \quad \bar{y}_8 \geq 1 \\
&\qquad C_{10}: \quad x_0 + x_2 + x_6 = 2 \\
&\qquad\quad x_0, x_2, x_6 \quad \in \{0,1\} \\
&\qquad\quad \bar{y}_0, \bar{y}_1, \bar{y}_2, \bar{y}_3, \bar{y}_4, \bar{y}_5, \bar{y}_6, \bar{y}_7, \bar{y}_8 \quad \in \{0,1\}
\end{aligned} \tag{4}
$$

3.2 Transformation into a Quadratic Binary Unconstrained Optimization (QUBO) model

The QUBO formulation is an expression with a linear component and a quadratic component and all the variables are binary variables (Kochenberger et al., 2014). The transformation of a linear optimization problem into a QUBO model requires the following steps:

- **Transform the inequality constraints into equality constraints**

  Each inequality constraint can be converted into an equality constraint of the type $ax + y = b$ by using slack variables (Bradley et al., 1977).

Following our example, the original formulation can be transformed as follows:

Minimize

$$4\bar{y}_0 + 5\bar{y}_1 + 2\bar{y}_2 + 2\bar{y}_3 + \bar{y}_4 + 4\bar{y}_5 + 2\bar{y}_6 + 2\bar{y}_7 + \bar{y}_8$$

Subject to

$$\begin{array}{lll}
C_1: & \bar{y}_0 + x_0 \geq 1 & \xrightarrow{\text{to}} \bar{y}_0 + x_0 - s_1 = 1 \\
C_2: & \bar{y}_1 + x_0 + x_2 \geq 1 & \xrightarrow{\text{to}} \bar{y}_1 + x_0 + x_2 - s_2 = 1 \\
C_3: & \bar{y}_2 + x_2 \geq 1 & \xrightarrow{\text{to}} \bar{y}_2 + x_2 - s_3 = 1 \\
C_4: & \bar{y}_3 + x_0 + x_6 \geq 1 & \xrightarrow{\text{to}} \bar{y}_3 + x_0 + x_6 - s_4 = 1 \\
C_5: & \bar{y}_4 \geq 1 & \xrightarrow{\text{to}} \bar{y}_4 = 1 \\
C_6: & \bar{y}_5 + x_2 \geq 1 & \xrightarrow{\text{to}} \bar{y}_5 + x_2 - s_5 = 1 \\
C_7: & \bar{y}_6 + x_6 \geq 1 & \xrightarrow{\text{to}} \bar{y}_6 + x_6 - s_6 = 1 \\
C_8: & \bar{y}_7 + x_6 \geq 1 & \xrightarrow{\text{to}} \bar{y}_7 + x_6 - s_7 = 1 \\
C_9: & \bar{y}_8 \geq 1 & \xrightarrow{\text{to}} \bar{y}_8 = 1 \\
C_{10}: & x_0 + x_2 + x_6 = 2 & \xrightarrow{\text{to}} x_0 + x_2 + x_6 = 2
\end{array}$$

$$\begin{array}{l}
x_0, x_2, x_6 \quad \in \{0, 1\} \\
\bar{y}_0, \bar{y}_1, \bar{y}_2, \bar{y}_3, \bar{y}_4, \bar{y}_5, \bar{y}_6, \bar{y}_7, \bar{y}_8 \quad \in \{0, 1\} \\
s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7 \quad \in \{\mathbb{Z}^+\}
\end{array}$$

- **Transform integer variables into binary variables**

  A QUBO formulation requires all variables to be binary. For this, we suggest the bounded coefficient proposed by Karimi and Ronagh (2019). For example, let s be an integer variable such that $0 \leq s \leq 5$. Following the bounded-coefficient method, s can be encoded as a sum of three new binary variables, s_1, s_2, and s_3, with coefficients such that their sum can generate all the values of s. For this case, the coefficients would be 1, 2 and 2. Therefore, $s = s_1 + 2s_2 + 2s_3$, with $s_1, s_2, s_3 \in \{0, 1\}$. Table 1 shows how to activate s_1, s_2, and s_3 to produce the integer values from zero to five.

- **Use penalties to move the constraints to the objective function**

  This is an important step in the transformation of a classical optimization model into a QUBO expression. This is also the step in which regional scientists can contribute the most towards constructing a bridge

s	s_1	s_2	s_3
0	0	0	0
1	1	0	0
2	0	1	0
3	1	1	0
4	0	1	1
5	1	1	1

Table 1: Constructing $0 \leq s \leq 5$ with binary variables.

towards the use of quantum computing in regional science. A detailed description of the creation of penalties is available in (Glover et al., 2019). The challenge is to move each constraint in the classical optimization model to the objective function as a penalty expression. The new unconstrained optimization model consists of a unique expression (the QUBO expression) that includes the objective function and penalties. The QUBO expression has the same optimal solution as the original optimization model. Table 2 presents useful penalty expressions for certain types of constraints. In the penalties, P is an arbitrary constant. (Glover et al., 2019) recommend a value for P that is between 75% and 150% of an estimate of the original objective function value. Also note that the penalties are quadratic expressions that fall within the description of a QUBO model.

Classical constraint	Equivalent penalty
$x + y \leq 1$	$P(xy)$
$x + y \geq 1$	$P(1 - x - y + xy)$
$x + y = 1$	$P(1 - x - y + 2xy)$
$x \leq y$	$P(x - xy)$
$x_1 + x_2 + x_3 \leq 1$	$P(x_1x_2 + x_1x_3 + x_2x_3)$
$x = y$	$P(x + y - 2xy)$

* P is an arbitrary constant, x and y binary variables.

Table 2: Conversion table

The QUBO expression of our example is as follows:

$$
\begin{aligned}
-44\bar{y}_0 - 43\bar{y}_1 - 46\bar{y}_2 - 46\bar{y}_3 - 47\bar{y}_4 - 44\bar{y}_5 - 46\bar{y}_6 - 46\bar{y}_7 - 47\bar{y}_8 \\
-240x_0 - 240x_2 - 240x_6 + 48s_{00} + 48s_{10} + 48s_{11} + 48s_{20} + 48s_{30} \\
+48s_{31} + 48s_{50} + 48s_{60} + 48s_{70} + [48\bar{y}_0^2 + 96\bar{y}_0x_0 - 96\bar{y}_0s_{00} \\
+48\bar{y}_1^2 + 96\bar{y}_1x_0 + 96\bar{y}_1x_2 - 96\bar{y}_1s_{10} - 96\bar{y}_1s_{11} + 48\bar{y}_2^2 + 96\bar{y}_2x_2 \\
-96\bar{y}_2s_{20} + 48\bar{y}_3^2 + 96\bar{y}_3x_0 + 96\bar{y}_3x_6 - 96\bar{y}_3s_{30} - 96\bar{y}_3s_{31} + 48\bar{y}_4^2 \\
+48\bar{y}_5^2 + 96\bar{y}_5x_2 - 96\bar{y}_5s_{50} + 48\bar{y}_6^2 + 96\bar{y}_6x_6 - 96\bar{y}_6s_{60} + 48\bar{y}_7^2 \\
+96\bar{y}_7x_6 - 96\bar{y}_7s_{70} + 48\bar{y}_8^2 + 192x_0^2 + 192x_0x_2 + 192x_0x_6 \\
-96x_0s_{00} - 96x_0s_{10} - 96x_0s_{11} - 96x_0s_{30} - 96x_0s_{31} + 192x_2^2 \\
+96x_2x_6 - 96x_2s_{10} - 96x_2s_{11} - 96x_2s_{20} - 96x_2s_{50} + 192x_6^2 \\
-96x_6s_{30} - 96x_6s_{31} - 96x_6s_{60} - 96x_6s_{70} + 48s_{00}^2 + 48s_{10}^2 \\
+96s_{10}s_{11} + 48s_{11}^2 + 48s_{20}^2 + 48s_{30}^2 + 96s_{30}s_{31} + 48s_{31}^2 \\
+48s_{50}^2 + 48s_{60}^2 + 48s_{70}^2]/2 + 312
\end{aligned}
\quad (5)
$$

In Table 3, we show that both the classical linear model and the quadratic model have the same optimal solution. This step validates the quadratic model.

The QUBO expression can be mapped into a specific quantum computer. Each device (i.e., gate-based or annealing) has specific instructions and frameworks that allow the user to perform the mapping. In Qiskit, this work is performed using the *QuadraticProgram* from *qiskit.optimization* (IBM, 2022), in Dwave (Shin et al., 2014) the reference has specific functions to send the quadratic program into the annealer.

3.3 Solving the QUBO expression into a quantum annealer computer

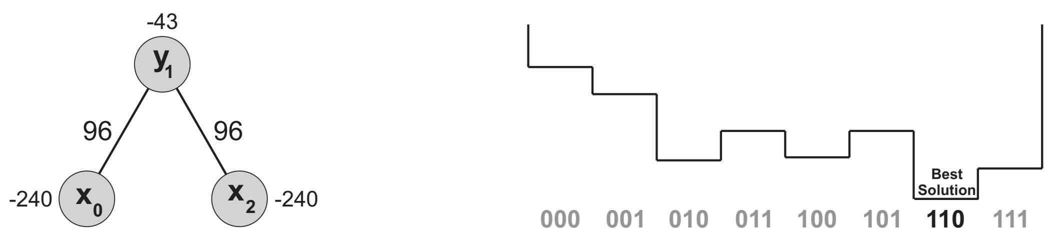
Quantum annealing computing (Albash and Lidar, 2018) uses the physical behavior of quantum systems to solve optimization and probabilistic sampling (D'Wave, 2022) problems. In this section, we focus on optimization and seek to find the optimal solution among a large, finite, set of possible solutions (as occurs in combinatorial optimization problems such as the MCLP). This technology is based on the fundamental principle of physics

Binary Variable	Linear Gurobi Solution	Quadratic Gurobi Solution
$\overline{y_0}$	1	1
$\overline{y_1}$	0	0
$\overline{y_2}$	0	0
$\overline{y_3}$	0	0
$\overline{y_4}$	1	1
$\overline{y_5}$	0	0
$\overline{y_6}$	0	0
$\overline{y_7}$	0	0
$\overline{y_8}$	1	1
x_0	0	0
x_2	1	1
x_6	1	1
s_{00}	-	0
s_{10}	-	0
s_{11}	-	0
s_{20}	-	0
s_{30}	-	0
s_{31}	-	0
s_{50}	-	0
s_{60}	-	0
s_{70}	-	0

Table 3: Gurobi classical quadratic and linear program solutions

that dictates that every object or system always seeks its minimum energy state (Das and Chakrabarti, 2008). In this case, the aim is to design an entangled quantum system in which each Qubit represents a binary variable of the QUBO expression to be solved. Furthermore, the minimum energy state of the system corresponds to the optimal solution of the problem. That is, when measuring the system, each qubit collapses to 0 or 1 as dictated by the optimal solution.

As an illustrative example, Figure 3a shows a quantum system with three qubits that represent the expression QUBO $-43y_1 - 240x_0 - 240x_2 + 96y_1x_0 + 96y_1x_2$. Each qubit represents a binary variable of the QUBO expression (i.e., X_0, X_2, and Y_1) and the quadratic relations between variables determine the entanglement relations between the qubits. Thus, the term $96y_1x_2$ implies an entanglement relationship between the qubits that represent the variables y_1 and x_2. Each basic state of this system (i.e., the different combinations of values for x_0, x_2 and y_1) has an associated energy level that allows the construction of what is known as the energy landscape. Figure 3b presents the energy landscape of our example and indicates that the lowest energy level corresponds to the basic states $x_0 = 0$, $x_2 = 0$ and $y_1 = 1$, which is the optimal solution that minimizes the QUBO expression. If the quantum system is well designed, then quantum physics takes the system to that basic state of minimum energy. The calculation of the total energy of a quantum system, given its state, is calculated with the Hamiltonian (Dirac, 1981).



(a) Quantum system (b) Energy Landscape

Figure 3: Quantum system and Energy Landscape for the QUBO expression $-43y_1 - 240x_0 - 240x_2 + 96y_1x_0 + 96y_1x_2$

In the absence of any external magnetic influence, the probability of a qubit collapsing to 0 or 1 is equal (50%, 50%). However, this probability can be altered in two ways. One way is to apply an external magnetic field to the qubit in such a way that the minimum energy state of the qubit is associated with 0 or 1. This external field is known as a bias (D'Wave, 2022). The second way is to create artificial entanglements between qubits, called couplers, in such a way that the behavior of a qubit depends on the behavior of the qubits with which it is entangled (D'Wave, 2022). Therefore, in annealing technology, the quantum system is created using couplers and biases (as dictated by the QUBO expression) so that the minimum energy state corresponds to the optimal solution.

The mathematical expression that represents the quantum annealers from D-Wave are presented in equations 6, where $\hat{\sigma}_{x,z}^{(i)}$ are Pauli matrices operating on a qubit, q_i, h_i is the qubit bias, and $J_{i,j}$ is the coupling strength. $A(s)$ and $B(s)$ are the amplitudes associated with each Hamiltonian (D'Wave, 2022). The first term is the initial Hamiltonian or “tunneling Hamiltonian” corresponds to the initial state of the system (i.e., disconnected qubits in an unbiased superposition). The second term, also called the “problem Hamiltonian”, corresponds to the energy state of the quantum system that represents the QUBO expression (i.e., with couplers and biases). In quantum annealing, the process starts in the tunnelling Hamiltonian state; the annealing process gradually introduces the problem Hamiltonian while ramping down the tunnelling Hamiltonian to zero. The idea is to stay in the minimum energy state throughout the annealing process so that, at the end, the system is in the minimum energy state of the problem Hamiltonian and, therefore, in the optimal solution. All these processes occur in 20 microseconds regardless of the size of the quantum system. During the annealing process, there is a probability for the system to jump out of the lowest energy state. Therefore, the annealing process needs to be repeated several times to define the final solution as the most likely final state. See (D’Wave, 2022) for more technical details on quantum annealing.

$$\mathrm{H(s)} = -\frac{A(s)}{2}\left(\sum_i \hat{\sigma}_x^{(i)}\right) + \frac{B(s)}{2}\left(\sum_i h_i\hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma}_z^{(i)}\hat{\sigma}_z^{(j)}\right) \tag{6}$$

3.4 Solving the QUBO expression in a gate-based quantum computer

While annealing quantum computation seeks to take advantage of the natural evolution of quantum states, the quantum gate model seeks to manipulate the way in which the state of the quantum system evolves over time. This manipulation is done by means of quantum gates that can be arranged in sequences to build a more complex algorithm. Quantum gates can be unary (one qubit), binary (two qubits), or ternary (three qubits). The main difference between classical gates (e.g., AND, OR, NAND, etc.) and quantum gates (e.g., Pauli-X, Hadamard, Phase, etc.) is that the latter are reversible (i.e., no information is lost in the process as it is with classical gates), which guarantees the preservation of entanglement and superpositions. The possibility of creating intricate circuits connecting multiple qubits with a variety of quantum gates is what makes gate-based quantum computers more applicable than quantum annealing computers. For illustrative purposes only, Figure 4 illustrates the quantum circuit of the QUBO expression $-43y_1 - 240x_0 - 240x_2 + 96y_1x_0 + 96y_1x_2$.

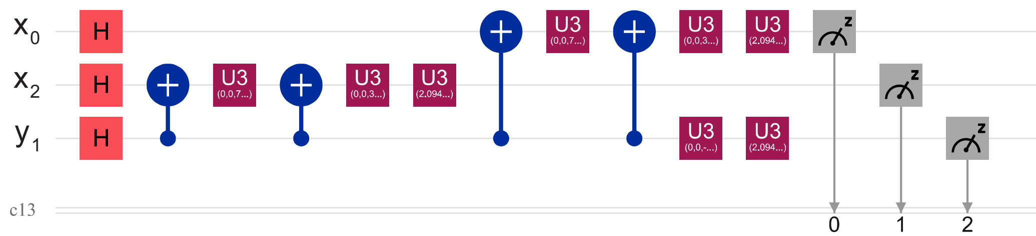


Figure 4: Example of the QAOA quantum circuit for que QUBO expression $-43y_1 - 240x_0 - 240x_2 + 96y_1x_0 + 96y_1x_2$

Currently, there are circuits specifically designed to run specific algorithms. One of them is the quantum approximate optimization algorithm (OAOA), which emulates the quantum annealing process (Farhi et al., 2014); therefore, it is of special interest within the framework of this paper.

The QAOA circuit takes a QUBO expression and encodes it into an operator that is, essentially, a quantum circuit. This circuit is composed of two parts: the cost and mixer layer. The first layer amplifies each qubit by a predefined γ parameter and rotates the quantum states of each qubit. Afterwards, the mixer layer converts each rotation into probability amplitudes amplified by a predefined β parameter; therefore, the mixer layer changes the probability of occurrence of each qubit. This process is repeated p times until it reaches an optimal value; the larger p is, the more likely it is that the result will be optimal. Finally, the measurement process is performed, and the qubits collapse into basic 0 or 1 states.

QAOA is described by equation 7 where the parameters β and γ define the unitary transformation $U(\beta, \gamma)$, and p is the number of times this transformation is applied. The β parameter is associated with the mixer layer by a unitary transformation $U(\beta) = e^{-i\beta H_B}$. Likewise, the γ parameter is associated with the cost layer by a unitary transformation $U(\gamma) = e^{-i\gamma H_P}$. The quantum state $|\psi_0\rangle$ is any suitable initial state, and is usually set to be an equally superposed state (Farhi et al., 2014). The graphical description of this circuit is shown in Figure 5.

$$|\psi(\gamma, \beta)\rangle = U\left(\beta_p\right) U\left(\gamma_p\right) \cdots U\left(\beta_1\right) U\left(\gamma_1\right) |\psi_0\rangle \tag{7}$$

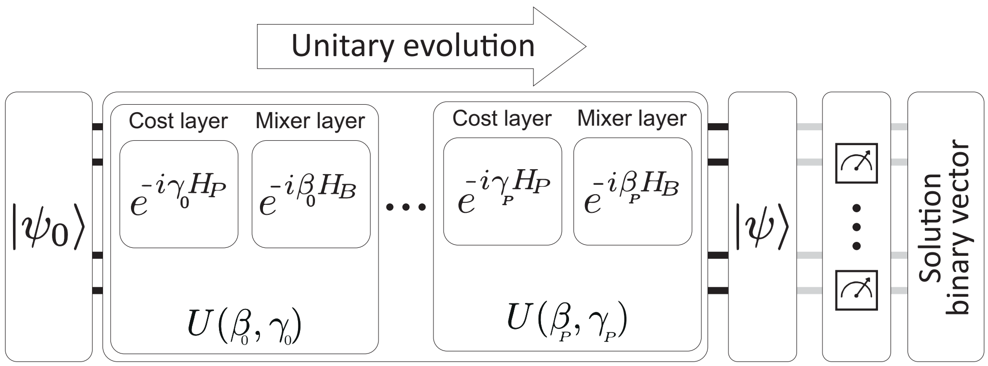


Figure 5: QAOA process

3.5 Solution vector decoding

Once the quantum process is finished, the system is measured, and all the qubits collapse into basic 0,1 states. Since each qubit represents a binary variable in the QUBO expression, we end up with a binary vector. When the original variables in the optimization problem are all binary (as in the MCLP), the binary vector is easy to interpret as it contains the optimal value of each decision variable (including the slack variables generated in the QUBO expression). If the original optimization problem contains integer variables that are transformed into binary variables in the QUBO expression, then it is necessary to decode the solution to obtain the initial decision variables, including the conversion from binary to integer. Table 4 shows the solution vectors resulting from solving our example instance from Figure 2. All 10 repetitions collapsed into the optimal solution. The table also shows the final values of the decision variables when solving the original linear formulation and the QUBO version of our example. Finally, Figure 6 shows the graphical representation of the optimal solution.

4 Computational experiment

In this section, we present the computational experiment in which we use both quantum technologies to solve small instances of the MCLP. We work with small instances because, as shown in Section 3, a small MCLP instance generates a QUBO expression with many binary variables that quickly make use of all the qubits that are available for experimentation. We hope that as

Binary variable	Lineal formulation	QUBO formulation	Annealer experiments									
			1	2	3	4	5	6	7	8	9	10
$\bar{y}_0$	1	1	1	1	1	1	1	1	1	1	1	1
$\bar{y}_1$	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{y}_2$	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{y}_3$	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{y}_4$	1	1	1	1	1	1	1	1	1	1	1	1
$\bar{y}_5$	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{y}_6$	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{y}_7$	0	0	0	0	0	0	0	0	0	0	0	0
$\bar{y}_8$	1	1	1	1	1	1	1	1	1	1	1	1
x_0	0	0	0	0	0	0	0	0	0	0	0	0
x_2	1	1	1	1	1	1	1	1	1	1	1	1
x_6	1	1	1	1	1	1	1	1	1	1	1	1
s_{00}		0	0	0	0	0	0	0	0	0	0	0
s_{10}		0	0	0	0	0	0	0	0	0	0	0
s_{11}		0	0	0	0	0	0	0	0	0	0	0
s_{20}		0	0	0	0	0	0	0	0	0	0	0
s_{30}		0	0	0	0	0	0	0	0	0	0	0
s_{31}		0	0	0	0	0	0	0	0	0	0	0
s_{50}		0	0	0	0	0	0	0	0	0	0	0
s_{60}		0	0	0	0	0	0	0	0	0	0	0
s_{70}		0	0	0	0	0	0	0	0	0	0	0

Table 4: Solution vectors of the MCLP instance presented in Figure 2 ($p = 2$).

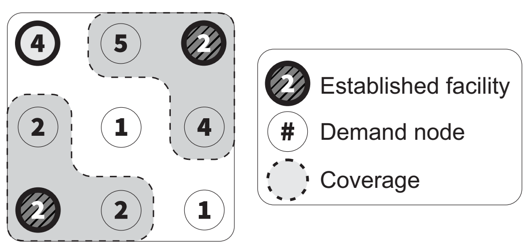


Figure 6: Optimal solution of the MCLP instance presented in Figure 2 ($p = 2$)

quantum technology evolves, it will be possible to solve larger instances of the MCLP.

Figure 7 presents the nine instances used in the experiment. They range from 9 to 20 demand nodes (I) and 3 to 7 eligible facility sites (J). In all cases, the aim was to locate 2 facilities (i.e., $p = 2$). Finally, the service distance, S, of all facilities corresponds to nearest neighbors using the rook criteria.

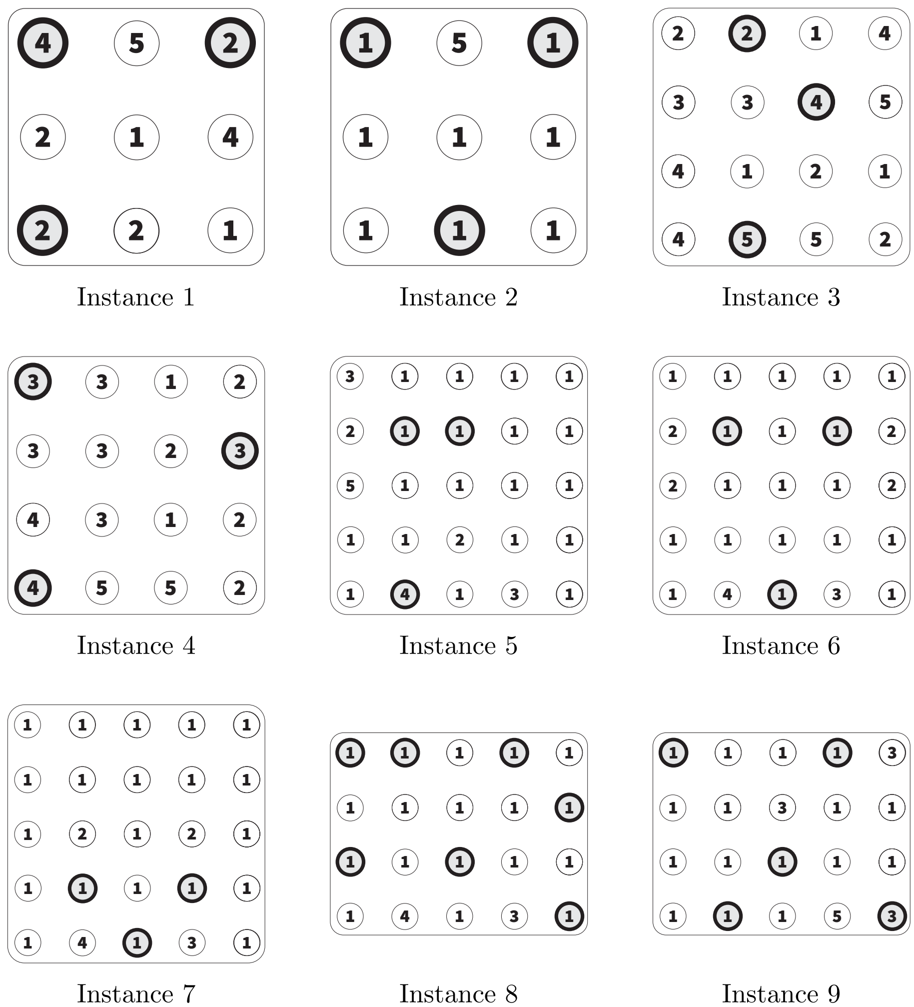


Figure 7: MCLP instances

For the annealing technology, we used a quantum computer from D-Wave (specifically the DW_2000_Q6) with 2000 Qubits and Chimera topology. For the gate technology, we used Qiskit's matrix product simulator (MPS) quantum simulator with 100 Qubits. We ran Qiskit on a classic supercomputer with 464 CPU cores and an Intel (R) Xeon (R) x8664 processor. We also ran Qiskit on Amazon Web Services (specifically ml.m5.24xlarge), with 96

CPU cores, and an Intel (R) Xeon (R) Platinum 8175M CPU, 2.50GHz. Table 5 reports the different libraries used in the experiment. We recognize that these libraries are continuously evolving, but they are a good reference for practitioners. The instances and codes utilized in this experiment are available in the following repository: https://github.com/XXX.

Process	Description	Library
Inputs and outputs	Generate ".lp" file, and classical linear and quadratic optimization	gurobiPy v.9.1.1
	Linear-quadratic conversion	docplex v.2.15.194 cplex v.12.10.0.1
Gate Based Quantum Computing	Quantum circuits	qiskit v.0.24.0 qiskit-aer v.0.7.6 qiskit-aqua v.0.8.2 qiskit-ibmq-provider v.0.12.1 qiskit-ignis v.0.5.2 qiskit-terra v.0.16.4
Quantum Annealing	Manage quadratic models	dimod v.0.9.13 pyqubo v.1.0.10
	Manage D'Waves devices	amazon-braket-ocean-plugin v.1.0.6
	Linear Optimization	docplex v.2.15.194 cplex v.12.10.0.1
	Embeding into D'Waves devices	dwave-system v.1.4.0

Table 5: Python libraries and associated function

As we explained in Section 2.1, the interference phenomenon makes it necessary for a quantum problem to be solved multiple times to find the final solution as the most likely one. For this reason, each of the nine instances of our experiment was resolved 10,000 times. In the jargon of quantum computing, each repetition is known as a "shot." Therefore, we obtain one solution out of those 10,000 shots (the one with lowest energy). To guarantee that the obtained solution was the optimal solution, we repeated the process 10 times for each instance. In summary, we solved each instance 10 x 10,000 times. This is only necessary for instances resolved with annealing technology (D-Wave), as this is where we use a real quantum computer. In the case of gate-based technology (Qiskit), the solution obtained is deterministic because a simulator on a classic computer is used.

5 Results

Figure 8 presents the graphical representation of the optimal solutions. All the instances were solved to optimality with quantum annealing technology. Only the instances ID=1, ID=2, and ID=3 were solved optimally with the gate-based quantum simulator.

5.1 Annealer Results

The results of the annealer experiments are summarized in Table 6. Columns 3 and 4 report the optimal solution (OF), which indicates the uncovered population, and the running time (in seconds) when solving the QUBO expression in Gurobi. The annealing experiments are described in columns 5 to 9. Column 5 (Optimal) indicates whether the quantum computer found the optimal solution. In this case, all instances were solved to optimality. Therefore, to the best of our knowledge, our results are the first nine instances of a real quantum computer successfully solving the MCLP. As we explained in Section 4, the probabilistic nature of quantum technology requires solving each instance multiple times to find the most likely solution. In our experiment, each instance was solved 10 times (each time we kept the solution with the lowest energy level from 10,000 shots). Column 6 (Success) indicates the number of times, out of 10 repetitions, that the quantum computer found the optimal solution. In eight instances, the optimal solution was reached in most repetitions. However, only 2 of the 10 repetitions reached the optimal solution of instance ID=8. This instability in the behavior of quantum annealing seems to be associated with the greater number of eligible facility sites, which for this instance is $J = 7$. In instance ID=9, where $J = 5$, this instability is also noticeable. Instances ID=8 and ID=9 are the most complex in terms of the number of variables in the QUBO expression. This can be seen in column 7 (Number of qubits), which indicates the number of qubits required to solve the instance. Column 8 (OF) reports the objective function (uncovered demand). We report the maximum and minimum values of the

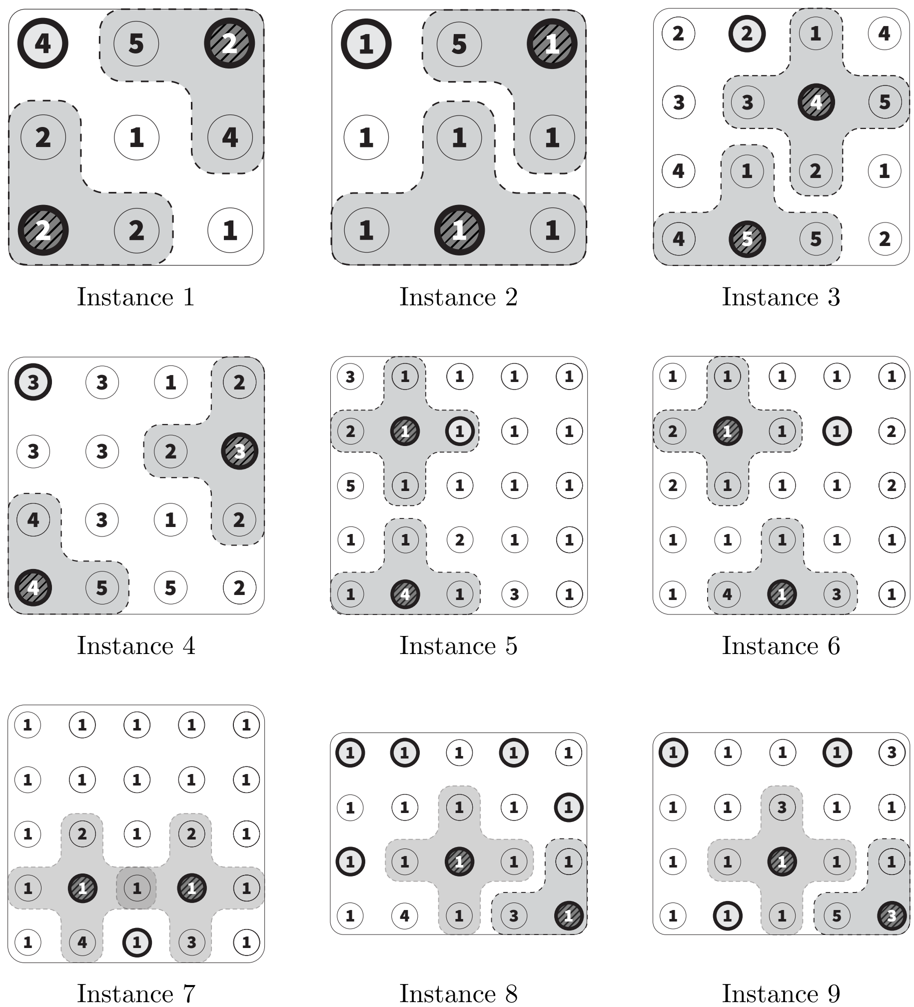


Figure 8: MCLP instance solutions

objective function in the cases in which the optimal solution was not found in all repetitions (i.e., IDs 5, 7, 8, and 9). Lastly, column 9 (Time per repetition) indicates the time required to solve the 10,000 shots. As explained in Section 3.3, the annealing process (one shot) takes $20\mu s$ regardless of the size of the instance to solve; because of this, the running time is the same for all instances.

ID	I, J, p	Gurobi		Annealing				
		OF	Time (s)	Optimal Yes/No	Success	Number of qubits	OF	Time per repetition (s)
1	9,3,2	6	0.02	Yes	10/10	21	6	0.2
2	9,3,2	2	0.01	Yes	10/10	22	2	0.2
3	16,3,2	18	0.02	Yes	10/10	32	18	0.2
4	16,3,2	24	0.01	Yes	10/10	29	24	0.2
5	25,3,2	26	0.01	Yes	8/10	42	26-27	0.2
6	25,3,2	19	0.02	Yes	10/10	42	19	0.2
7	25,3,2	16	0.01	Yes	9/10	41	16-23	0.2
8	20,7,2	15	0.03	Yes	2/10	54	15-17	0.2
9	20,5,2	14	0.03	Yes	7/10	44	14-15	0.2

Table 6: Summary of annealing experiments

5.2 QAOA Results

The challenge in this experiment is different from the annealing experiment. This is because the QAOA experiment is performed in a deterministic simulator that emulates quantum behavior in a classical computer. This deterministic nature implies that, having set the parameters, an instance must be resolved once. Therefore, the challenge is to calibrate the values of the parameters β and γ in equation 7 to allow the optimal solution to be reached. Figure 9 shows how volatile the relationship is between these two parameters and the objective function. Finally, the parameter p in equation 7, which in theory should tend to infinity, is computationally expensive (Farhi et al., 2014). Therefore, in our experiment, we set $p = 1$, as we understand that this experiment, and the paper in general, seeks to lay the foundations for future research.

Table 7 presents the results of the QAOA simulations. The first aspect that stands out is that with this technology, it was only possible to solve three of the nine instances of the experiment. As in the previous table of results (Table 6), for each instance, we report the optimal solution and computation times required by Gurobi to solve the QUBO expression to optimality. For the gate-based technology, we report whether the optimal value was found, the values of β and γ that allowed the optimal solution to be reached, the value of the objective function, the computation time, and the number of

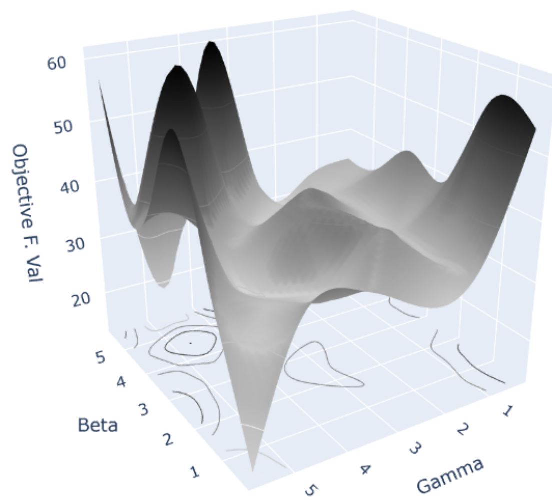


Figure 9: Landscape for objective function values from QAOA with different β (x axis) and γ (y axis) parameters in an MCLP instance.

qubits required (which coincide with those required in annealing technology because they are the same QUBO expressions). Note that the values of the parameters β and γ do not necessarily have to be different for each instance to be solved. Instances ID=2 and ID=3 reached the optimal solution with $\beta = 5.65$ and $\gamma = 0.31$, which is a positive result that invites further study of the relationship between these parameters and the optimal solution of the MCLP. However, the execution times are very high and volatile compared to those reported for the annealing technology.

6 Conclusions

This paper presents the first empirical evidence on the solution of the MCLP with quantum computers, which constitutes an important milestone in the literature of this classic problem. The computational experiments show that quantum annealing has great potential for solving this problem, which is as expected as this technology is specific for solving combinatorial optimization problems. In contrast, although it is of more general use, gate-based quantum technology did not show good performance when simulating the annealing

ID	I, J, p	Gurobi		QAOA Simulations				
		OF	Time (s)	Optimal	β and γ	OF	Time QAOA (min)	Number of qubits
1	9,3,2	6	0.02	YES	0.314/5.16	6	1.8	21
2	9,3,2	2	0.01	YES	5.65/0.31	2	68.3	22
3	16,3,2	18	0.02	YES	5.65/0.31	18	1.0	32
4	16,3,2	24	0.01	NO	N/A	N/A	N/A	29
5	25,3,2	26	0.01	NO	N/A	N/A	N/A	42
6	25,3,2	19	0.02	NO	N/A	N/A	N/A	42
7	25,3,2	16	0.01	NO	N/A	N/A	N/A	41
8	20,7,2	15	0.03	NO	N/A	N/A	N/A	54
9	20,5,2	14	0.03	NO	N/A	N/A	N/A	44

Table 7: Summary of QAOA simulations.

process with the QAOA. In both cases, it is clear that further advances in quantum technology are required to surpass classical computation in solving combinatorial optimization problems.

This paper opens new research opportunities for regional scientists (without implying that they must be experts in quantum computing). (1) We present the formulation of the QUBO expressions of other classic combinatorial problems in regional science, not only in localization but also in transportation, region design, and scheduling, among others. (2) We propose the design of new penalty expressions, such as those presented in Table 2, facilitate the formulation of QUBO expressions. (3) We propose the design of search strategies for the parameters β and γ. This includes the possibility of hybrid algorithms that combine quantum and classical computing. (4) We find the minimum depth (i.e., QAOA parameter p) that guarantees convergence to an optimal solution. All these alternatives allow regional science, as a field, to be better prepared for the day when quantum computers are more advanced.

References

Abdelgaber, N. and C. Nikolopoulos (2020). Overview on quantum computing and its applications in artificial intelligence. In *2020 IEEE Third*

International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp. 198–199. IEEE.

Albash, T. and D. A. Lidar (2018, January). Adiabatic quantum computation. *Reviews of Modern Physics 90*(1), 015002. Publisher: American Physical Society.

Benioff, P. (1980). The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics 22*(5), 563–591.

Bradley, S. P., A. C. Hax, and T. L. Magnanti (1977). *Applied Mathematical Programming*. Addison Wesley.

Brassard, G., I. Chuang, S. Lloyd, and C. Monroe (1998, September). Quantum computing. *Proceedings of the National Academy of Sciences 95*(19), 11032–11033. Publisher: National Academy of Sciences.

Cao, Y., J. Romero, and A. Aspuru-Guzik (2018). Potential of quantum computing for drug discovery. *IBM Journal of Research and Development 62*(6), 6–1.

Choi, J., S. Oh, and J. Kim (2020). The useful quantum computing techniques for artificial intelligence engineers. In *2020 International Conference on Information Networking (ICOIN)*, pp. 1–3. IEEE.

Chuang, I., N. Gershenfeld, and M. Kubinec (1998). Experimental implementation of fast quantum searching. *Physical Review Letters - PHYS REV LETT*.

Church, R. and A. Murray (2018). Location covering models. advances in spatial science.

Church, R. and C. ReVelle (1974). The maximal covering location problem. In *Papers of the regional science association*, Volume 32, pp. 101–118. Springer-Verlag.

Das, A. and B. K. Chakrabarti (2008, September). Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics 80*(3), 1061–1081. Publisher: American Physical Society.

de Falco, D., B. Apolloni, and N. Cesa-Bianchi (1988, July). A numerical implementation of quantum annealing.

Desrochers, M., J. Desrosiers, and M. Solomon (1992, April). A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows. *Operations Research 40*(2), 342–354. Publisher: INFORMS.

Deutsch, D. (1985). Quantum theory, the church–turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 400*(1818), 97–117.

Deutsch, D. and R. Jozsa (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences 439*(1907), 553–558. Publisher: Royal Society.

Deutsch, D. E. (1989). Quantum computational networks. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences 425*(1868), 73–90.

Dirac, P. A. M. P. A. M. (1981). *The principles of quantum mechanics*. Clarendon Press.

D'Wave (2022). What is Quantum Annealing? — D-Wave System Documentation documentation. `https://docs.dwavesys.com/docs/latest/c_gs_2.html`. Accesed 2022-01-09.

Egger, D. J., C. Gambella, J. Marecek, S. McFaddin, M. Mevissen, R. Raymond, A. Simonetto, S. Woerner, and E. Yndurain (2020). Quantum computing for finance: state of the art and future prospects. *IEEE Transactions on Quantum Engineering*.

Farhi, E., J. Goldstone, and S. Gutmann (2014). A quantum approximate optimization algorithm. *arXiv:1411.4028 [quant-ph]*.

Fernández-Carames, T. M. and P. Fraga-Lamas (2020). Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks. *IEEE access 8*, 21091–21116.

Feynman, R. (1960). There is plenty of room at the bottom. *Engineering and Science 5*(23), 22–36.

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics 21*(6-7), 467–488.

Galvão, R. D. and C. ReVelle (1996). A lagrangean heuristic for the maximal covering location problem. *European Journal of Operational Research 88*(1), 114–123.

Glover, F., G. Kochenberger, and Y. Du (2019). Quantum bridge analytics i: a tutorial on formulating and using qubo models. *4OR 17*(4), 335–371.

Gonzalez, C. (2021, April). Cloud based QC with Amazon Braket. *Digitale Welt 5*(2), 14–17.

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. *arXiv:quant-ph/9605043*.

Hardy, Y. and W.-H. Steeb (2001). *Classical and Quantum Computing.* Birkhäuser Basel.

Harwood, S., C. Gambella, D. Trenev, A. Simonetto, D. Bernal, and D. Greenberg (2021). Formulating and Solving Routing Problems on Quantum Computers. *IEEE Transactions on Quantum Engineering 2*, 1–17. Conference Name: IEEE Transactions on Quantum Engineering.

Hidary, J. D. (2019). *Quantum Computing: An Applied Approach.* Cham: Springer International Publishing.

IBM (2001, December). IBM's Test-Tube Quantum Computer Makes History; First Demonstration Of Shor's Historic Factoring Algorithm. `https://www.sciencedaily.com/releases/2001/12/011220081620.htm`.

IBM (2021, March). DOcplex examples. `https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/icos/12.9.0?topic=api-docplex-examples`.

IBM (January 9 2022). Quadratic Programs — Qiskit 0.24.0 documentation. `https://qiskit.org/documentation/tutorials/optimization/1\_quadratic\_program.html`.

Karimi, S. and P. Ronagh (2019). Practical integer-to-binary mapping for quantum annealers. *Quantum Information Processing 18*(4), 94.

Kochenberger, G., J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang (2014). The unconstrained binary quadratic programming problem: a survey. *Journal of combinatorial optimization 28*(1), 58–81.

Mavroeidis, V., K. Vishi, M. D. Zych, and A. Jøsang (2018). The impact of quantum computing on present cryptography. *arXiv preprint arXiv:1804.00200*.

Megiddo, N., E. Zemel, and S. L. Hakimi (1983). The maximum coverage location problem. *SIAM Journal on Algebraic Discrete Methods 4*(2), 253–261.

Merali, Z. (2011, June). First sale for quantum computing. *Nature 474*(7349), 18–18. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7349 Primary_atype: News Publisher: Nature Publishing Group Subject_term: Business;Information theory and computation;Quantum information Subject_term_id: business;information-theory-and-computation;quantum-information.

Murray, A. T. (2016). Maximal coverage location problem: impacts, significance, and evolution. *International Regional Science Review 39*(1), 5–27.

Murray, A. T. and R. L. Church (1996). Applying simulated annealing to location-planning models. *Journal of Heuristics 2*(1), 31–53.

Nielsen, M. A. and I. L. Chuang (2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press.

Orus, R., S. Mugel, and E. Lizaso (2019). Quantum computing for finance: Overview and prospects. *Reviews in Physics 4*, 100028.

Outeiral, C., M. Strahm, J. Shi, G. M. Morris, S. C. Benjamin, and C. M. Deane (2021). The prospects of quantum computing in computational molecular biology. *Wiley Interdisciplinary Reviews: Computational Molecular Science 11*(1), e1481.

Santos, A. C. (2016, September). The IBM Quantum Computer and the IBM Quantum Experience. *Revista Brasileira de Ensino de Física 39*(1). arXiv: 1610.06980.

Scherer, W. (2019). *Mathematics of Quantum Computing: An Introduction.* Springer International Publishing.

Shin, S. W., G. Smith, J. A. Smolin, and U. Vazirani (2014). How "quantum" is the d-wave machine? *arXiv:1401.7087 [quant-ph].*

Shor, P. W. (1995, 8). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer.

Simon, D. R. (1997). On the power of quantum computation. *SIAM Journal on Computing 26*(5), 1474–1483. Publisher: Society for Industrial and Applied Mathematics.

Tong, D., A. Murray, and N. Xiao (2009). Heuristics in spatial analysis: a genetic algorithm for coverage maximization. *Annals of the Association of American Geographers 99*(4), 698–711.

Weaver, J. R. and R. L. Church (1986). A location model based on multiple metrics and multiple facility assignment. *Transportation Research Part B: Methodological 20*(4), 283–296.