

# Formulation of the Social Workers' Problem in Quadratic Unconstrained Binary Optimization Form and Solve It on a Quantum Computer

Atchade Parfait Adelomou\* , Elisabet Golobardes Ribé, Xavier Vilasis Cardona

Research Group in Data Science for the Digital Society (DS4DS), Barcelona, Spain

Email: \*parfait.atchade@salle.url.edu

**How to cite this paper:** Adelomou, A.P., Ribé, E.G. and Cardona, X.V. (2020) Formulation of the Social Workers' Problem in Quadratic Unconstrained Binary Optimization Form and Solve It on a Quantum Computer. *Journal of Computer and Communications*, 8, 44-68.

<https://doi.org/10.4236/jcc.2020.811004>

**Received:** September 28, 2020

**Accepted:** November 9, 2020

**Published:** November 12, 2020

Copyright © 2020 by author(s) and

Scientific Research Publishing Inc.

This work is licensed under the Creative

Commons Attribution-NonCommercial

International License (CC BY-NC 4.0).

<http://creativecommons.org/licenses/by-nc/4.0/>



Open Access

---

## Abstract

The problem of social workers visiting their patients at home is a class of combinatorial optimization problems and belongs to the class of problems known as NP-Hard. These problems require heuristic techniques to provide an efficient solution in the best of cases. In this article, in addition to providing a detailed resolution of the social workers' problem using the Quadratic Unconstrained Binary Optimization Problems (QUBO) formulation, an approach to mapping the inequality constraints in the QUBO form is given. Finally, we map it in the Hamiltonian of the Ising model to solve it with the Quantum Exact Solver and Variational Quantum Eigensolvers (VQE). The quantum feasibility of the algorithm will be tested on IBMQ computers.

## Keywords

QUBO, Quantum Algorithms, Variational Quantum Eigensolvers, Combinatorial Optimization Algorithms

---

## 1. Introduction

The social workers' problem is the problem defined by social workers who visit patients at home to provide personalized assistant care according to a schedule, time and duration of the visit based on the patient's pathology. This problem combines, on the one hand, a routing problem such as Vehicle Routing Problem (VRP) [1] [2] [3] and, on the other, a planning problem [4]. Hence the exact solution can drive an exponential computation time for growing scale input data where the need for another approach to solving these problems is veritably crucial. The complexity and above all, the importance of these problems involved the scientific community in investigating efficient methods to solve them [5].

The social workers' problem, as the combination of VRP and planning problem, is NP-Hard. This definitely, leads us to explore new approaches for the large-scale social workers and one of these approaches to take into account is Quantum computing [6].

Quantum computing could help us in a change of the degree of complexity of the problem, empowered by its high computation capacity. Within the large fields where quantum computing is called to stand out, is the field of combinatorial optimization. And one of the most special algorithms in this field is the Quadratic Unconstrained Binary Optimization Problems (QUBO).

The QUBO [7] [8] [9] is one of the most relevant categories of optimization problems. It was designed to solve quantum annealing problems and maps to perfection with D-wave technology [7]. QUBO refers to a pattern matching technique used in machine learning and optimization, which involves minimizing a quadratic polynomial over binary variables [7]. It must be ruled out that QUBO is NP-hard [10]. And also, because this formulation does not admit restrictions, it could significantly limit its use when modelling systems that need constraints (especially inequality). However, many famous combinatorial optimization problems take advantage of the QUBO formulation to be solved [7] [8] [9] and also to be used in quantum computers based on gate model [7].

This article provides a detailed resolution of the social workers' problem using the Quadratic Unconstrained Binary Optimization Problems (QUBO) formulation. We also propose an approach to mapping the inequality constraints in the QUBO form and finally, we map it in the Hamiltonian of the Ising model to solve it with the Quantum Exact Solver and Variational Quantum Eigensolvers (VQE) [11]. The quantum feasibility of the algorithm will be tested on IBMQ computers.

The paper is organized as follows. In Section 2, the quantum computers, the Adiabatic Quantum Computing (AQC), and the VQE will be explored by focusing on the two dominant configurations and their techniques. In Section 3, we introduce the question we want to solve and the approaches. In Section 4, 5 and 6 we outline our proposed formulation to solve the problem, we describe details of its algebraic QUBO formulation and we translate it into Ising form ready to be solved with any quantum solver, in this paper, mainly with the VQE. Finally, in Section 7, we present the results of our experimental analysis. In the discussion, we give a summary, scale our new formulation over some open problems.

## 2. Quantum Techniques

It is essential to understand the leading quantum techniques that exist to know what steps to follow for the implementation of an algorithm in quantum. And especially for this work that uses the QUBO formulation to solve the social workers' problem.

There are several techniques for building a quantum computer, and the way it is currently done is by combining multiple various multicore processors [9]. All

of this brings to life numerous models of quantum computing: theoretical models, quantum circuit models, adiabatic quantum computing, measurement-based quantum computing, and topological quantum computing, being equivalent to each other within the reduction of polynomial-time. The most widespread and considerably developed model is the circuit model for gate-based quantum computation. The conceptual generalization of Boolean logic gates used for classical computing works for quantum computing as well [12]. And, with the combination of these basic gates and the appropriate memory structures based on architecture, it gives life to the quantum computer. Quantum annealing has a somewhat different software stack structure than gate-based model quantum computers. The annealing-based computer must be viewed as a specific case of a quantum accelerator based on quantum gate algorithms. So instead of a quantum circuit, the level of abstraction is Ising's classic model.

Currently, there are two dominant configurations for quantum computing: Quantum Annealing (D-Wave), in which problems are coded in quantum Hamiltonians and the natural dynamics of physical systems, and the Gate Model Quantum Calculus (IBM) [7] [12] [13], in which the calculation is made through a series of discrete gate operations.

In quantum annealing, optimization is achieved by mapping the Hamiltonian optimization problem of a controllable quantum system so that the lowest-energy states correspond to optimal solutions. The quantum superconducting circuit analyzers produced by D-Wave Systems Inc are the most mature [7] [11]. So, we can visualize a D-Wave quantum computer like is a physical representation of the Ising model like the one in Equation (1).

For gate-based machines, one of the most promising algorithms [13] [14] for optimization is known by Ansatz Alternative Quantum Operator, also known as the Approximate Quantum Optimization Algorithm (QAOA) [11]. QAOA is exclusively designed to run in polynomial time on Noisy Intermediate-Scale Quantum (NISQ) [11] devices and find optimal solutions to optimization problems. This algorithm is called to solve critical optimization problems that classically take exponential computational complexity to solve exactly. Although, in principle, QAOA could be considered a gate model or a continuous-time configuration [15] [16], also like the Variational Quantum Eigensolvers (VQE) [11].

As the experiment designed in this paper goes through the QUBO formulation, the Adiabatic Quantum Computing will be analyzed, which is an essential step for the implementation of our algorithm.

The Adiabatic Quantum Computing uses a concept of quantum physics known as the adiabatic theorem [17]. The adiabatic theorem states that as long as this transformation occurs slow enough, the system has time to adapt and will remain in the lowest energy configuration [7].

The process followed by the AQC can be summarized in two very recognizable steps:

1) The first step is to prepare a system and initialize it to its lowest energy state known as the ground state.

2) The second step allows us to transform/map our problem in the system.

The researchers H. Nishimori and T. Kadowaki demonstrated that it is necessary to use Ising's transversal model to code the problem to be optimized and activate it slowly. Equation (1) illustrates how we can use the adiabatic theorem for computation [8].

$$H_{\text{Ising}} = -A(t) \left( \sum_{i < j} J_{ij} Z_i Z_j + \sum_i h_i Z_i \right) + B(t) \sum_i h_i X_i \quad (1)$$

with  $X_i$  and  $Z_i$  the Pauli matrices in  $X$  and  $Z$  respectively.

Taking into account that:

$$A(t)|_{t=0} = 0 \quad \text{and} \quad B(0)|_{t=0} = 1 \quad (2)$$

As mentioned above, if we want the system to be in the ground state, the initial state must be

$$|\psi(t)\rangle|_{t=0} = |+_1\rangle_x \otimes |+_2\rangle_x \otimes \dots \otimes |_{+_{n-1}}\rangle_x \otimes |+_n\rangle_x \quad (3)$$

with  $|+_i\rangle_x$  is the proper state of  $\sigma_i^x(X_i)$ .

Now suppose that quantum annealing ends at  $T \gg 1$ , where  $A(t)|_{t=T} = 1$  and  $B(t)|_{t=T} = 0$ . The adiabatic theorem ensures that  $|\psi(t)\rangle|_{t=T}$  will be the ground state of  $H(t)$  such that the energy of the system will be the energy destined for the ground state. This is the basis of AQC. There are several derivatives of this technique, but it is not in line with this paper [7] [10] [14] [18].

As the given the social workers' problem is purely combinatorial, and the recommended steps to follow when designing a combinatorial quantum algorithm are:

- 1) define the model in the QUBO formulation,
- 2) apply the change of variables to bring it to the Ising model,
- 3) and then use a variational algorithm to find the optimal solutions the VQE will be analyzed.

Unfortunately, we are still in the NISQ [11] era. Even, this era allowed the scientific community to develop hybrid algorithms that use the potential of quantum computing and especially the experience of classical algorithms in the areas of optimization—leaving quantum computers operations that require a very high computational cost.

One of the most famous algorithms is the Variational Quantum Eigensolver (VQE) [15] [16] [19] which is based on the variational principle [15]. The native VQE will be used to find optimal solutions to our social workers' problem.

VQE is a hybrid quantum-classical algorithm that combines quantum mechanical aspects to the classical algorithm. And its goal is to find approximate solutions to combinatorial problems. One of the fundamental approaches is to map the combinatorial problems onto a physics problem (Hamiltonian). Namely onto a

problem that can be phrased in terms of a Hamiltonian of the Ising model. So, identify the solution to the combinatorial problem is related by finding the ground state of this physics problem. Thus, the goal is to find the ground state of this Hamiltonian. The approach we follow is related to the annealing algorithm.

The unknown eigenvectors are prepared by varying the experimental parameters and calculating the Rayleigh-Ritz ratio [20] in a classical minimization (Figure 1). At the end of the algorithm, the reconstruction of the eigenvector that is stored in the final set of experimental parameters that define the state will be done.

From the variational principle the following equation  $\langle H \rangle_{\psi(\vec{\theta})} \geq \lambda_i$  can be reached out, with  $\lambda_i$  as eigenvector and  $\langle H \rangle_{\psi(\vec{\theta})}$  as the expected value. By this way, the VQE finds (4) such an optimal choice of parameters  $\vec{\theta}$ , that the expected value is minimized and that a lower eigenvalue is located.

$$\langle H \rangle = \langle \psi(\theta) | H | \psi(\theta) \rangle \tag{4}$$

### 3. Definition of the Social Workers Problem

Let  $n$  be the number of patients (users) and considering a weekly calendar of visits for each of them. Our objective is to find an optimal meeting’s timing which minimizes the cost of time travel; hence, money and maximize the number of visits to the patients in a work schedule.

In our case study, the daily schedule (Table 1), is set at 8 hours, and the distance between patients is at least 15 minutes.

- A set of social workers (  $N_1, N_2, N_3, \dots$  )

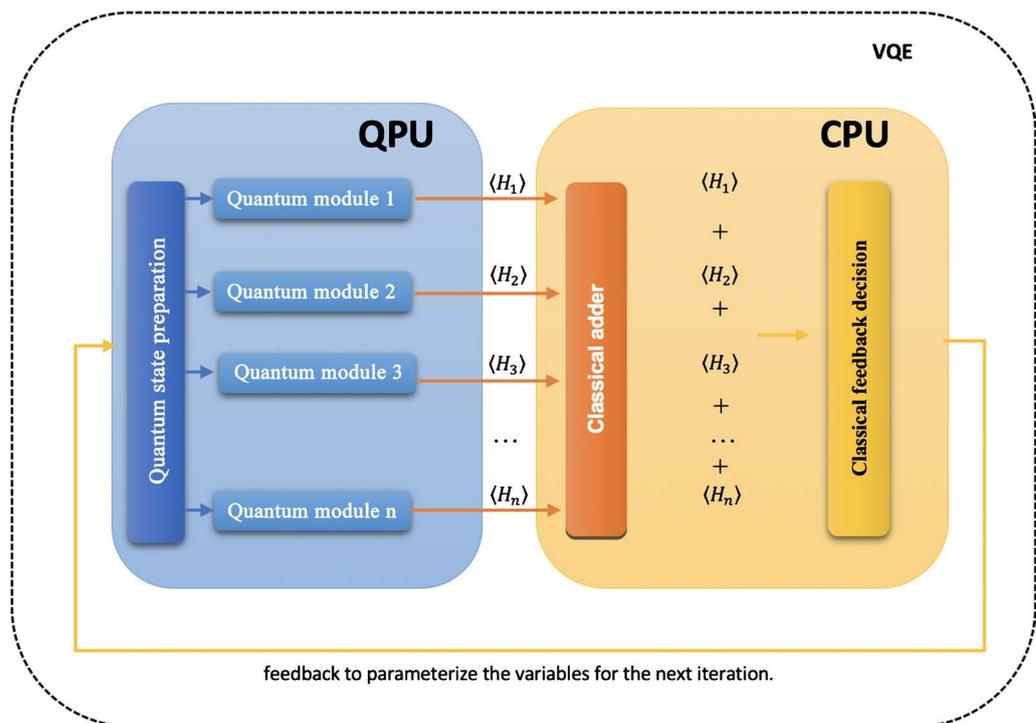


Figure 1. The variational quantum eigensolver diagram.

**Table 1.** Schedule of patient visits without any association with social workers.

	<b>M</b>	<b>T</b>	<b>W</b>	<b>TH</b>	<b>F</b>
9:00-10:00	$U_1$	$U_1$	$U_1$	$U_1$	$U_1$
9:30-10:30	$U_4$				
10:15-11:15			$U_4$		
11:30-12:30					$U_3 U_4$
11:45-12:45	$U_5$				
12:00-13:00	$U_2$				
14:45-15:45		$U_2$			
15:00-16:00				$U_3$	
15:15-16:15	$U_3$				
15:45-16:45			$U_3$		
16:00-17:00			$U_2$		
16:30-17:30		$U_3$			
17:00-18:00					$U_3$

Where  $U_1$  to  $U_5$  are the patients (users) and equal to the variable  $i$  or  $j$  of the mathematical formulation.

- A set of patients ( $P_1, P_2, P_3, \dots$ )
- A set of visits ( $U_1, U_2, U_3, \dots$ )
  - each visit is linked to a patient: a patient can have multiple appointments on a day
  - for each visit, we know the start time and duration
- Social workers can work at most 8 hours per day (that means, social workers cannot do a visit the first visit at 9:00 and end the last tour at 18:00) (removed to reduce the number of variables, so this can be solved using Qiskit<sup>1</sup>)
- We know the cost of travelling between each pair of patients. The cost can be seen as a function of travel time and distance.

The objective is the following:

- Find a schedule where each visit is assigned to a social worker
- We are minimizing the travel cost while also respecting that a social worker does not work more than 8 hours per day.

We design this formulation keeping in mind that our device does not admit inequality restrictions. For this, we will encode in some way the time information that will represent the limits of the constraints in the said formula.

#### 4. Formulation of the Social Workers' Problem

What we pursue is to avoid using the inequality constraints and use the least number of qubits. Within the real environments (not simulation) we have very few real qubits to test our algorithm. Therefore, we will base our algorithm on techniques already consolidated to achieve efficiency in several qubits.

<sup>1</sup>[www.qiskit.org](http://www.qiskit.org).

Let  $G=(V, E, c)$  be a complete graph directed with  $V = \{0, 1, 2, \dots, n\}$ , as the set of nodes and  $E = \{(i, j) : i, j \in V, i \neq j\}$  as the set of edges, where node 0 represents the central, for a team of  $k$  social workers with the same capacity  $\rho$  and  $n$  remaining nodes that represent geographically dispersed patients.

Each patient  $i \in V - \{0\}$  has a specific demand for visits  $d_i \leq \rho$ . The non-negative travel cost  $W_{ij}$  is associated with each arc  $(i, j) \in E$ . To simplify, we consider that the distances are symmetrical. Where  $x_{ij}$  are the decision variables of the paths between two patients. The minimum number of social workers needed to care for all patients is  $\frac{\sum_{i=1}^n d_i}{Q}$ .

Minimize:

$$\sum_{k=1}^m \sum_{i=1}^l \sum_{j=1, i \neq j}^n W_{ij} x_{ijk}, \tag{5}$$

Subject to:

$$\sum_{k=1}^m \sum_{j=1}^n x_{ijk} = 1 \quad \forall i \in \{1, \dots, n\}, \tag{6}$$

$$\sum_{k=1}^m \sum_{i=1}^n x_{jik} = 1 \quad \forall j \in \{1, \dots, n\}, \tag{7}$$

$$\sum_{i=1}^l d_i \sum_{j=1}^n x_{ijk} \leq q \quad \forall k \in \{1, \dots, m\}, \tag{8}$$

$$\sum_{j=1}^n x_{0jk} = K \quad \forall k \in \{1, \dots, m\}, \tag{9}$$

$$\sum_{j=1}^n x_{j0k} = K \quad \forall k \in \{1, \dots, m\}, \tag{10}$$

$$\sum_{i=1}^n x_{ihk} - \sum_j x_{hjk} = 0 \quad \forall h \in \{1, \dots, n\} \text{ and } \forall k \in \{1, \dots, m\} \tag{11}$$

$$x_{ijk} \in \{0, 1\}, \quad \forall i, j \in \{0, \dots, n\}, \quad i \neq j \quad \forall k \in \{1, \dots, m\} \tag{12}$$

In this formulation, the objective function (5) minimizes total travel savings taking into account the new cost function with the time window. The restrictions of Equation (6) impose that exactly the arcs  $k$  leave the plant; (9) and (10) are the restrictions on the degree of entry and exit of the social worker from the central office. The constraints (11) are the route of continuity and the elimination of sub-courses, which ensure that the solution does not contain a sub-route disconnected from the exchange. Restrictions (12) are mandatory.

To solve the posed scheduling problem, we need a time variable. The Introduction of the time restriction (schedule) in the original QUBO formulation is a significant obstacle, as discussed above [10].

So, the strategy we follow is to incorporate the schedule information (calendar) of **Table 1** in the weight variable. Equations (13) and (14) describe the temporal evolution of each social worker equivalent to achieve the scheduling parameters.

$$W_{ij} = d_{ij} + f(t_{ij}) \tag{13}$$

$$f(t_{ij}) = \gamma \frac{(\tau_i - \tau_j)^2}{d_{\max} - d_{\min}} \tag{14}$$

where  $W_{ij}$  is our weight and time window function,  $d_{ij}$  is the distance between the patient  $i$  and  $j$  and  $f(t_{ij})$  is our time window's function. The function  $f(t_{ij})$  is a growing function, and we model it by a quadratic function to weigh short distances concerning large ones. We are taking into account that the first weight function  $W_{ij} = d_{ij}$  is a distance function, we want to make  $f(t_{ij})$  behave like  $d_{ij}$ , and thus, be able to take full advantage of the behaviour of the primary objective function.

$\gamma > 0$  It is a weighted degree parameter of our time window function;  $\tau_i$  is the start time of a time slot for patient  $i$  and  $\tau_j$  for the patient  $j$ . where  $d_{\max}$  represents the maximum distance between all patients and,  $d_{\min}$  is the minimum distance between the gaps of all patients. The term  $T_{ij} = (\tau_i - \tau_j) > 0$  is the time window.

The simplified Hamiltonian resulting from the schedule optimization problem is as follows:

$$H = \frac{1}{d_{\max} - d_{\min}} \sum_{ij \in E} \left( (d_{\max} - d_{\min}) d_{ij} + (\tau_i - \tau_j)^2 \right) x_{i,j} + A \sum_{i=1}^n \left( 1 - \sum_{j \in \delta(i)^+} x_{i,j} \right)^2 + A \sum_{i=1}^n \left( 1 - \sum_{j \in \delta(i)^-} x_{ji} \right)^2 + A \left( k - \sum_{i \in \delta(0)^+} x_{0,i} \right)^2 + A \left( k - \sum_{j \in \delta(0)^+} x_{j,0} \right)^2 \tag{15}$$

### 5. Resolution of Social Workers' Problem Based on QUBO Formulation

Let us solve our formulation in QUBO form by considering  $n=4$  patients. Where  $Q$  is a  $2^N \times 2^N$  matrix with  $N = n(n-1)$  as the number of qubits. So, in this case,  $N = 12$  qubits. Let's remember that we are using binary variables so,  $(x_{u,v}^i)^2 = x_{u,v}^i$ .

Now, let's develop step by step each term of (15) into the QUBO form considering the definition of our cost function from (13) and (14).

$$\begin{aligned} \sum_{ij \in E} W_{ij} x_{i,j} &= \sum_i \sum_{j \neq i} W_{ij} x_{i,j} \\ &= W_{0,1} x_{0,1} + W_{0,2} x_{0,2} + W_{0,3} x_{0,3} + W_{1,0} x_{1,0} + W_{1,2} x_{1,2} + W_{1,3} x_{1,3} \\ &\quad + W_{2,0} x_{2,0} + W_{2,1} x_{2,1} + W_{2,3} x_{2,3} + W_{3,0} x_{3,0} + W_{3,1} x_{3,1} + W_{3,2} x_{3,2} \end{aligned} \tag{16}$$

$$\begin{aligned} &A \sum_{i=1}^3 \left( 1 - \sum_{j \in \delta(i)^-} x_{ji} \right)^2 \\ &= A \left( 1 - x_{0,1} - x_{2,1} - x_{3,1} + 1 - x_{0,2} - x_{1,2} - x_{3,3} + 1 - x_{0,3} - x_{1,3} - x_{2,3} \right)^2 \\ &= A \left( 3 - x_{0,1} - x_{2,1} - x_{3,1} - x_{0,2} - x_{1,2} - x_{3,2} - x_{0,3} - x_{1,3} - x_{2,3} \right)^2 \end{aligned}$$

$$\begin{aligned}
 &= A(3^2 + (x_{0,1})^2 + (x_{2,1})^2 + (x_{3,1})^2 + (x_{0,2})^2 + (x_{1,2})^2 + (x_{3,2})^2 + (x_{0,3})^2 \\
 &\quad + (x_{1,3})^2 + (x_{2,3})^2 - 6x_{0,1} - 6x_{2,1} - 6x_{3,1} - 6x_{0,2} - 6x_{1,2} - 6x_{3,2} - 6x_{0,3} \\
 &\quad - 6x_{1,3} - 6x_{2,3} + 2x_{0,1}x_{2,1} + 2x_{0,1}x_{3,1} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{1,2} + 2x_{0,1}x_{3,2} \\
 &\quad + 2x_{0,1}x_{0,3} + 2x_{0,1}x_{1,3} + 2x_{0,1}x_{2,3} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{0,2} + 2x_{2,1}x_{1,2} \\
 &\quad + 2x_{2,1}x_{3,2} + 2x_{2,1}x_{0,3} + 2x_{2,1}x_{1,3} + 2x_{2,1}x_{2,3} + 2x_{3,1}x_{0,2} + 2x_{3,1}x_{1,2} \\
 &\quad + 2x_{3,1}x_{3,2} + 2x_{3,1}x_{0,3} + 2x_{3,1}x_{1,3} + 2x_{3,1}x_{2,3} + 2x_{0,2}x_{1,2} + 2x_{0,2}x_{3,2} \\
 &\quad + 2x_{0,2}x_{0,3} + 2x_{0,2}x_{1,3} + 2x_{0,2}x_{2,3} + 2x_{1,2}x_{3,2} + 2x_{1,2}x_{0,3} + 2x_{1,2}x_{1,3} \\
 &\quad + 2x_{1,2}x_{2,3} + 2x_{3,2}x_{0,3} + 2x_{3,2}x_{1,3} + 2x_{3,2}x_{2,3} + 2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} + 2x_{1,3}x_{2,3}) \tag{17}
 \end{aligned}$$

$$\begin{aligned}
 &A \sum_{i=1}^3 \left( 1 - \sum_{j \in \delta(i)^+} x_{i,j} \right)^2 \\
 &= A(1 - x_{1,0} - x_{1,2} - x_{1,3} + 1 - x_{2,0} - x_{2,1} - x_{2,3} + 1 - x_{3,0} - x_{3,1} - x_{3,2})^2 \\
 &= A(3 - x_{1,0} - x_{1,2} - x_{1,3} - x_{2,0} - x_{2,1} - x_{2,3} - x_{3,0} - x_{3,1} - x_{3,2})^2 \\
 &= A(3^2 + (x_{1,0})^2 + (x_{1,2})^2 + (x_{1,3})^2 + (x_{2,0})^2 + (x_{2,1})^2 + (x_{2,3})^2 + (x_{3,0})^2 \\
 &\quad + (x_{3,1})^2 + (x_{3,2})^2 - 6x_{1,0} - 6x_{1,2} - 6x_{1,3} - 6x_{2,0} - 6x_{2,1} - 6x_{2,3} - 6x_{3,0} \\
 &\quad - 6x_{3,1} - 6x_{3,2} + 2x_{1,0}x_{1,2} + 2x_{1,0}x_{1,3} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{2,1} + 2x_{1,0}x_{2,3} \\
 &\quad + 2x_{1,0}x_{3,0} + 2x_{1,0}x_{3,1} + 2x_{1,0}x_{3,2} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,0} + 2x_{1,2}x_{2,1} \\
 &\quad + 2x_{1,2}x_{2,3} + 2x_{1,2}x_{3,0} + 2x_{1,2}x_{3,1} + 2x_{1,2}x_{3,2} + 2x_{1,3}x_{2,0} + 2x_{1,3}x_{2,1} \\
 &\quad + 2x_{1,3}x_{2,3} + 2x_{1,3}x_{3,0} + 2x_{1,3}x_{3,1} + 2x_{1,3}x_{3,2} + 2x_{2,0}x_{2,1} + 2x_{2,0}x_{2,3} \\
 &\quad + 2x_{2,0}x_{3,0} + 2x_{2,0}x_{3,1} + 2x_{2,0}x_{3,2} + 2x_{2,1}x_{2,3} + 2x_{2,1}x_{3,0} + 2x_{2,1}x_{3,1} \\
 &\quad + 2x_{2,1}x_{3,2} + 2x_{2,3}x_{3,0} + 2x_{2,3}x_{3,1} + 2x_{2,3}x_{3,2} + 2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2} + 2x_{3,1}x_{3,2}) \tag{18}
 \end{aligned}$$

$$\begin{aligned}
 &A \left( k - \sum_{i \in \delta(0)^+} x_{0,i} \right)^2 = A(k - x_{0,1} - x_{0,2} - x_{0,3})^2 \\
 &= A(k^2 + (x_{0,1})^2 + (x_{0,2})^2 + (x_{0,3})^2 - 2kx_{0,1} - 2kx_{0,2} \\
 &\quad - 2kx_{0,3} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3}) \tag{19}
 \end{aligned}$$

$$\begin{aligned}
 &A \left( k - \sum_{j \in \delta(0)^+} x_{j,0} \right)^2 = A(k - x_{1,0} - x_{2,0} - x_{3,0})^2 \\
 &= A(k^2 + (x_{1,0})^2 + (x_{2,0})^2 + (x_{3,0})^2 - 2kx_{1,0} - 2kx_{2,0} \\
 &\quad - 2kx_{3,0} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0}) \tag{20}
 \end{aligned}$$

Grouping the terms (16) to (20) we reach out to the following expression:

$$\begin{aligned}
 &W_{0,1}x_{0,1} + W_{0,2}x_{0,2} + W_{0,3}x_{0,3} + W_{1,2}x_{1,2} + W_{1,3}x_{1,3} + W_{2,1}x_{2,1} + W_{2,3}x_{2,3} \\
 &+ A(3^2 + (x_{0,1})^2 + (x_{2,1})^2 + (x_{3,1})^2 + (x_{0,2})^2 + (x_{1,2})^2 + (x_{3,2})^2 + (x_{0,3})^2 \\
 &+ (x_{1,3})^2 + (x_{2,3})^2 - 6x_{0,1} - 6x_{2,1} - 6x_{3,1} - 6x_{0,2} - 6x_{1,2} - 6x_{3,2} - 6x_{0,3} \\
 &- 6x_{1,3} - 6x_{2,3} + 2x_{0,1}x_{2,1} + 2x_{0,1}x_{3,1} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{1,2} + 2x_{0,1}x_{3,2} \\
 &+ 2x_{0,1}x_{0,3} + 2x_{0,1}x_{1,3} + 2x_{0,1}x_{2,3} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{0,2} + 2x_{2,1}x_{1,2}
 \end{aligned}$$

$$\begin{aligned}
& + 2x_{2,1}x_{3,2} + 2x_{2,1}x_{0,3} + 2x_{2,1}x_{1,3} + 2x_{2,1}x_{2,3} + 2x_{3,1}x_{0,2} + 2x_{3,1}x_{1,2} \\
& + 2x_{3,1}x_{3,2} + 2x_{3,1}x_{0,3} + 2x_{3,1}x_{1,3} + 2x_{3,1}x_{2,3} + 2x_{0,2}x_{1,2} + 2x_{0,2}x_{3,2} \\
& + 2x_{0,2}x_{0,3} + 2x_{0,2}x_{1,3} + 2x_{0,2}x_{2,3} + 2x_{1,2}x_{3,2} + 2x_{1,2}x_{0,3} + 2x_{1,2}x_{1,3} \\
& + 2x_{1,2}x_{2,3} + 2x_{3,2}x_{0,3} + 2x_{3,2}x_{1,3} + 2x_{3,2}x_{2,3} + 2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} + 2x_{1,3}x_{2,3} \\
& + A(3^2 + (x_{1,0})^2 + (x_{1,2})^2 + (x_{1,3})^2 + (x_{2,0})^2 + (x_{2,1})^2 + (x_{2,3})^2 + (x_{3,0})^2 \\
& + (x_{3,1})^2 + (x_{3,2})^2 - 6x_{1,0} - 6x_{1,2} - 6x_{1,3} - 6x_{2,0} - 6x_{2,1} - 6x_{2,3} - 6x_{3,0} \\
& - 6x_{3,1} - 6x_{3,2} + 2x_{1,0}x_{1,2} + 2x_{1,0}x_{1,3} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{2,1} + 2x_{1,0}x_{2,3} \\
& + 2x_{1,0}x_{3,0} + 2x_{1,0}x_{3,1} + 2x_{1,0}x_{3,2} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,0} + 2x_{1,2}x_{2,1} \\
& + 2x_{1,2}x_{2,3} + 2x_{1,2}x_{3,0} + 2x_{1,2}x_{3,1} + 2x_{1,2}x_{3,2} + 2x_{1,3}x_{2,0} + 2x_{1,3}x_{2,1} \\
& + 2x_{1,3}x_{2,3} + 2x_{1,3}x_{3,0} + 2x_{1,3}x_{3,1} + 2x_{1,3}x_{3,2} + 2x_{2,0}x_{2,1} + 2x_{2,0}x_{2,3} \\
& + 2x_{2,0}x_{3,0} + 2x_{2,0}x_{3,1} + 2x_{2,0}x_{3,2} + 2x_{2,1}x_{2,3} + 2x_{2,1}x_{3,0} + 2x_{2,1}x_{3,1} \\
& + 2x_{2,1}x_{3,2} + 2x_{2,3}x_{3,0} + 2x_{2,3}x_{3,1} + 2x_{2,3}x_{3,2} + 2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2} + 2x_{3,1}x_{3,2} \\
& + A(k^2 + (x_{0,1})^2 + (x_{0,2})^2 + (x_{0,3})^2 - 2kx_{0,1} - 2kx_{0,2} - 2kx_{0,3} + 2x_{0,1}x_{0,2} \\
& + 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3}) + A(k^2 + (x_{1,0})^2 + (x_{2,0})^2 + (x_{3,0})^2 - 2kx_{1,0} - 2kx_{2,0} \\
& - 2kx_{3,0} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0})
\end{aligned} \tag{21}$$

Now let's apply the binary variable property  $x_{i,j}^2 = x_{i,j}$  so,

$$\begin{aligned}
& W_{0,1}x_{0,1} + W_{0,2}x_{0,2} + W_{0,3}x_{0,3} + W_{1,2}x_{1,2} + W_{1,3}x_{1,3} + W_{2,1}x_{2,1} + W_{2,3}x_{2,3} \\
& + A(3^2 + x_{0,1} + x_{2,1} + x_{3,1} + x_{0,2} + x_{1,2} + x_{3,2} + x_{0,3} + x_{1,3} + x_{2,3} \\
& - 6x_{0,1} - 6x_{2,1} - 6x_{3,1} - 6x_{0,2} - 6x_{1,2} - 6x_{3,2} - 6x_{0,3} - 6x_{1,3} - 6x_{2,3} \\
& + 2x_{0,1}x_{2,1} + 2x_{0,1}x_{3,1} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{1,2} + 2x_{0,1}x_{3,2} + 2x_{0,1}x_{0,3} \\
& + 2x_{0,1}x_{1,3} + 2x_{0,1}x_{2,3} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{0,2} + 2x_{2,1}x_{1,2} + 2x_{2,1}x_{3,2} \\
& + 2x_{2,1}x_{0,3} + 2x_{2,1}x_{1,3} + 2x_{2,1}x_{2,3} + 2x_{3,1}x_{0,2} + 2x_{3,1}x_{1,2} + 2x_{3,1}x_{3,2} \\
& + 2x_{3,1}x_{0,3} + 2x_{3,1}x_{1,3} + 2x_{3,1}x_{2,3} + 2x_{0,2}x_{1,2} + 2x_{0,2}x_{3,2} + 2x_{0,2}x_{0,3} \\
& + 2x_{0,2}x_{1,3} + 2x_{0,2}x_{2,3} + 2x_{1,2}x_{3,2} + 2x_{1,2}x_{0,3} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,3} \\
& + 2x_{3,2}x_{0,3} + 2x_{3,2}x_{1,3} + 2x_{3,2}x_{2,3} + 2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} + 2x_{1,3}x_{2,3} \\
& + A(3^2 + x_{1,0} + x_{1,2} + x_{1,3} + x_{2,0} + x_{2,1} + x_{2,3} + x_{3,0} + x_{3,1} + x_{3,2} \\
& - 6x_{1,0} - 6x_{1,2} - 6x_{1,3} - 6x_{2,0} - 6x_{2,1} - 6x_{2,3} - 6x_{3,0} - 6x_{3,1} - 6x_{3,2} \\
& + 2x_{1,0}x_{1,2} + 2x_{1,0}x_{1,3} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{2,1} + 2x_{1,0}x_{2,3} + 2x_{1,0}x_{3,0} \\
& + 2x_{1,0}x_{3,1} + 2x_{1,0}x_{3,2} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,0} + 2x_{1,2}x_{2,1} + 2x_{1,2}x_{2,3} \\
& + 2x_{1,2}x_{3,0} + 2x_{1,2}x_{3,1} + 2x_{1,2}x_{3,2} + 2x_{1,3}x_{2,0} + 2x_{1,3}x_{2,1} + 2x_{1,3}x_{2,3} \\
& + 2x_{1,3}x_{3,0} + 2x_{1,3}x_{3,1} + 2x_{1,3}x_{3,2} + 2x_{2,0}x_{2,1} + 2x_{2,0}x_{2,3} + 2x_{2,0}x_{3,0} \\
& + 2x_{2,0}x_{3,1} + 2x_{2,0}x_{3,2} + 2x_{2,1}x_{2,3} + 2x_{2,1}x_{3,0} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{3,2} \\
& + 2x_{2,3}x_{3,0} + 2x_{2,3}x_{3,1} + 2x_{2,3}x_{3,2} + 2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2} + 2x_{3,1}x_{3,2} \\
& + A(k^2 + x_{0,1} + x_{0,2} + x_{0,3} - 2kx_{0,1} - 2kx_{0,2} - 2kx_{0,3} + 2x_{0,1}x_{0,2} \\
& + 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3}) + A(k^2 + x_{1,0} + x_{2,0} + x_{3,0} - 2kx_{1,0} - 2kx_{2,0} \\
& - 2kx_{3,0} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0})
\end{aligned} \tag{22}$$

Grouping similar terms:

$$\begin{aligned}
 &W_{0,1}x_{0,1} + W_{0,2}x_{0,2} + W_{0,3}x_{0,3} + W_{1,2}x_{1,2} + W_{1,3}x_{1,3} + W_{2,1}x_{2,1} + W_{2,3}x_{2,3} \\
 &+ A(3^2 - 5x_{0,1} - 5x_{2,1} - 5x_{3,1} - 5x_{0,2} - 5x_{1,2} - 5x_{3,2} - 5x_{0,3} - 5x_{1,3} - 5x_{2,3} \\
 &+ 2x_{0,1}x_{2,1} + 2x_{0,1}x_{3,1} + 2x_{0,1}x_{0,2} + 2x_{0,1}x_{1,2} + 2x_{0,1}x_{3,2} + 2x_{0,1}x_{0,3} \\
 &+ 2x_{0,1}x_{1,3} + 2x_{0,1}x_{2,3} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{0,2} + 2x_{2,1}x_{1,2} + 2x_{2,1}x_{3,2} \\
 &+ 2x_{2,1}x_{0,3} + 2x_{2,1}x_{1,3} + 2x_{2,1}x_{2,3} + 2x_{3,1}x_{0,2} + 2x_{3,1}x_{1,2} + 2x_{3,1}x_{3,2} \\
 &+ 2x_{3,1}x_{0,3} + 2x_{3,1}x_{1,3} + 2x_{3,1}x_{2,3} + 2x_{0,2}x_{1,2} + 2x_{0,2}x_{3,2} + 2x_{0,2}x_{0,3} \\
 &+ 2x_{0,2}x_{1,3} + 2x_{0,2}x_{2,3} + 2x_{1,2}x_{3,2} + 2x_{1,2}x_{0,3} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,3} \\
 &+ 2x_{3,2}x_{0,3} + 2x_{3,2}x_{1,3} + 2x_{3,2}x_{2,3} + 2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} + 2x_{1,3}x_{2,3}) \\
 &+ A(3^2 - 5x_{1,0} - 5x_{1,2} - 5x_{1,3} - 5x_{2,0} - 5x_{2,1} - 5x_{2,3} - 5x_{3,0} - 5x_{3,1} - 5x_{3,2} \\
 &+ 2x_{1,0}x_{1,2} + 2x_{1,0}x_{1,3} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{2,1} + 2x_{1,0}x_{2,3} + 2x_{1,0}x_{3,0} \\
 &+ 2x_{1,0}x_{3,1} + 2x_{1,0}x_{3,2} + 2x_{1,2}x_{1,3} + 2x_{1,2}x_{2,0} + 2x_{1,2}x_{2,1} + 2x_{1,2}x_{2,3} \\
 &+ 2x_{1,2}x_{3,0} + 2x_{1,2}x_{3,1} + 2x_{1,2}x_{3,2} + 2x_{1,3}x_{2,0} + 2x_{1,3}x_{2,1} + 2x_{1,3}x_{2,3} \\
 &+ 2x_{1,3}x_{3,0} + 2x_{1,3}x_{3,1} + 2x_{1,3}x_{3,2} + 2x_{2,0}x_{2,1} + 2x_{2,0}x_{2,3} + 2x_{2,0}x_{3,0} \\
 &+ 2x_{2,0}x_{3,1} + 2x_{2,0}x_{3,2} + 2x_{2,1}x_{2,3} + 2x_{2,1}x_{3,0} + 2x_{2,1}x_{3,1} + 2x_{2,1}x_{3,2} \\
 &+ 2x_{2,3}x_{3,0} + 2x_{2,3}x_{3,1} + 2x_{2,3}x_{3,2} + 2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2} + 2x_{3,1}x_{3,2}) \\
 &+ A(k^2 + x_{0,1} + x_{0,2} + x_{0,3} - 2kx_{0,1} - 2kx_{0,2} - 2kx_{0,3} + 2x_{0,1}x_{0,2} \\
 &+ 2x_{0,1}x_{0,3} + 2x_{0,2}x_{0,3}) + A(k^2 + x_{1,0} + x_{2,0} + x_{3,0} - 2kx_{1,0} - 2kx_{2,0} \\
 &- 2kx_{3,0} + 2x_{1,0}x_{2,0} + 2x_{1,0}x_{3,0} + 2x_{2,0}x_{3,0}) \tag{23}
 \end{aligned}$$

Now let's group the linear terms as following:

$$\begin{aligned}
 &2A3^2 + 2Ak^2 + (W_{0,1} + A(1 - 5 - 2k))x_{0,1} + (W_{0,2} + A(1 - 5 - 2k))x_{0,2} \\
 &+ (W_{0,3} + A(1 - 5 - 2k))x_{0,3} + A(W_{1,0} - A(1 - 5 - 2k))x_{1,0} \\
 &+ A(W_{2,0} - A(1 - 5 - 2k))x_{2,0} + A(W_{3,0} - A(1 - 5 - 2k))x_{3,0} \tag{24} \\
 &+ (W_{1,2} - 10A)x_{1,2} + (W_{1,3} - 10A)x_{1,3} + (W_{2,1} - 10A)x_{2,1} \\
 &+ (W_{2,3} - 10A)x_{2,3} - (W_{3,1} - 10A)x_{3,1} - (W_{3,2} - 10A)x_{3,2}
 \end{aligned}$$

Now let's group quadratic terms.

$$\begin{aligned}
 &+ 4Ax_{0,1}x_{0,2} + 4Ax_{0,1}x_{0,3} + 4Ax_{0,2}x_{0,3} + 2Ax_{0,1}x_{1,2} + 2Ax_{0,1}x_{1,3} + 2Ax_{0,1}x_{2,1} \\
 &+ 2Ax_{0,1}x_{2,3} + 2Ax_{0,1}x_{3,1} + 2Ax_{0,1}x_{3,2} + 4Ax_{1,0}x_{2,0} + 4x_{1,0}x_{3,0} + 4Ax_{2,0}x_{3,0} \\
 &+ 2Ax_{1,0}x_{1,2} + 2Ax_{1,0}x_{1,3} + 2Ax_{1,0}x_{2,1} + 2Ax_{1,0}x_{2,3} + 2Ax_{1,0}x_{3,1} + 2Ax_{1,0}x_{3,2} \\
 &+ 2Ax_{2,1}x_{1,2} + 2Ax_{2,1}x_{1,3} + 2Ax_{2,1}x_{3,0} + 4Ax_{2,1}x_{3,1} + 4Ax_{2,1}x_{3,2} + 2Ax_{2,1}x_{0,2} \\
 &+ 2Ax_{2,1}x_{0,3} + 4Ax_{2,1}x_{2,3} + 2Ax_{1,2}x_{0,3} + 2Ax_{1,2}x_{2,0} + 2Ax_{1,2}x_{2,1} + 2Ax_{1,2}x_{3,0} \\
 &+ 2Ax_{1,2}x_{3,1} + 4Ax_{1,2}x_{1,3} + 4Ax_{1,2}x_{3,2} + 4Ax_{1,2}x_{2,3} + 2Ax_{3,1}x_{0,2} + 2Ax_{3,1}x_{1,2} \\
 &+ 4Ax_{3,1}x_{3,2} + 2Ax_{3,1}x_{0,3} + 2Ax_{3,1}x_{1,3} + 2Ax_{3,1}x_{2,3} + 2Ax_{1,3}x_{2,0} + 2Ax_{1,3}x_{2,1} \\
 &+ 4Ax_{1,3}x_{2,3} + 2Ax_{1,3}x_{3,0} + 2Ax_{1,3}x_{3,1} + 2Ax_{1,3}x_{3,2} + 2Ax_{0,2}x_{1,2} + 2Ax_{0,2}x_{3,2} \\
 &+ 2Ax_{0,2}x_{1,3} + 2Ax_{0,2}x_{2,3} + 2Ax_{2,0}x_{2,1} + 2Ax_{2,0}x_{2,3} + 2Ax_{2,0}x_{3,1} + 2Ax_{2,0}x_{3,2} \\
 &+ 2Ax_{3,2}x_{0,3} + 2Ax_{3,2}x_{1,3} + 2Ax_{3,2}x_{2,3} + 2Ax_{2,3}x_{3,0} + 2Ax_{2,3}x_{3,1} + 2Ax_{2,3}x_{3,2} \tag{25} \\
 &+ 2x_{0,3}x_{1,3} + 2x_{0,3}x_{2,3} + 2x_{3,0}x_{3,1} + 2x_{3,0}x_{3,2}
 \end{aligned}$$

Recalling the quadratic form  $x^T Qx + g^T x + C$  with the following terms.

$$X = \begin{bmatrix} x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{1,0} \\ x_{1,2} \\ x_{1,3} \\ x_{2,0} \\ x_{2,1} \\ x_{2,3} \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 4 & 4 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 4 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 0 & 2 & 2 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 2 & 2 & 4 & 2 & 2 & 4 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 4 & 2 & 2 & 4 & 2 & 2 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 4 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 2 & 4 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$g = \begin{bmatrix} (W_{0,1} - A(4 + 2k)) \\ (W_{0,2} - A(4 + 2k)) \\ (W_{0,3} - A(4 + 2k)) \\ (W_{1,0} - A(4 + 2k)) \\ (W_{1,2} - 10A) \\ (W_{1,3} - 10A) \\ (W_{2,0} - A(4 + 2k)) \\ (W_{2,1} - 10A) \\ (W_{2,3} - 10A) \\ (W_{3,0} - A(4 + 2k)) \\ (W_{3,1} - 10A) \\ (W_{3,2} - 10A) \end{bmatrix} \quad \text{and } C = 2Ak^2 + 2A(n-1)^2$$
(26)

We put Equations (13) and (14) together, and we arrive at the following expression:

$$W_{ij} = d_{ij} + \gamma \frac{(\tau_i - \tau_j)^2}{d_{\max} - d_{\min}}$$
(27)

There are several strategies to implement the objective function with inequality constraints. But each of these strategies requires additional variables that end up working as a stack. The approach most frequently used in the scientific community is to use auxiliary binary variables to convert inequality to equality and then proceed, as usual, by squaring the equality constraint following penalty theory [21].

These additional variables are translated into qubits; extra qubits that today are scarce for some experiments.

To do this, we define a strategy that allows us to solve combinatorial optimization problems with inequality constraints without the need to increase the number of qubits required.

Our strategy is based on coding the variables of time inequality, the time

window following the formulation  $W_{ij} = d_{ij} + \gamma \frac{(\tau_i - \tau_j)^2}{d_{\max} - d_{\min}}$ .

However, IBM’s significant contribution<sup>2</sup> opens up promising horizons in quadratic programming with inequality constraints (with more additional qubits).

Now we only have to code the information in **Table 1**, the data (distance, costs and correction) of each patient and generate our weight matrix, which in turn will serve to calculate our variables linear  $g$ .

With all this, we already have all the components for one, write our objective function in the form QUBO and second solve it efficiently on quantum annealing computers.

### 6. QUBO to Ising Hamiltonian Formulation

As we already have our objective function as a QUBO in the form  $\langle x^T | Q | x \rangle$ , now we can map our QUBO to Ising Hamiltonian formulation leads to calculating the values of  $J_{ij}$  and  $h_i$ .

The transformation between Ising Hamiltonian and QUBO is  $Z = 2x - 1$ , with  $Z$  as Pauli-Z matrix. This means that by writing an algorithm for QUBO with this single variable change, we will have the algorithm in Ising form. That is very useful to have the algorithm for various platforms that are based on quantum gates (meanly IBM Q) or quantum annealing (meanly D-Wave) in case of going from the Hamiltonian form.

**Table 2.** QUBO Q matrix for the social workers problem.

	$x_{0,1}$	$x_{0,2}$	$x_{0,3}$	$x_{1,0}$	$x_{1,2}$	$x_{1,3}$	$x_{2,0}$	$x_{2,1}$	$x_{2,3}$	$x_{3,0}$	$x_{3,1}$	$x_{3,2}$
$x_{0,1}$	0	4	4	0	2	2	0	2	2	0	2	2
$x_{0,2}$	0	0	4	0	2	2	0	2	2	0	2	2
$x_{0,3}$	0	0	0	0	2	2	0	2	2	0	2	2
$x_{1,0}$	0	0	0	0	2	2	4	2	2	4	2	2
$x_{1,2}$	0	0	0	0	0	4	2	2	4	2	2	4
$x_{1,3}$	0	0	0	0	0	0	2	2	4	2	2	2
$x_{2,0}$	0	0	0	0	0	0	0	2	2	4	2	2
$x_{2,1}$	0	0	0	0	0	0	0	0	4	2	4	4
$x_{2,3}$	0	0	0	0	0	0	0	0	0	2	2	2
$x_{3,0}$	0	0	0	0	0	0	0	0	0	0	2	2
$x_{3,1}$	0	0	0	0	0	0	0	0	0	0	0	4
$x_{3,2}$	0	0	0	0	0	0	0	0	0	0	0	0

<sup>2</sup><https://medium.com/qiskit/towards-quantum-advantage-for-optimization-with-qiskit-9a564339ef26>.

Next, we calculate these variables. We start with  $J_{ij}$  as is summarized in **Table 3**.

$$J_{ij} = \begin{cases} \frac{q_{i,j} + q_{j,i}}{4} & \text{if } i < j \\ 0 & \text{otherwise} \end{cases} \tag{28}$$

Let's calculate the external forces  $h_i$ :

$$h_i = \frac{1}{4} \sum_{k=1}^N (q_{i,k} + q_{k,i})$$

$$h_i = \frac{1}{4} [(q_{i,1} + q_{1,i}) + (q_{i,2} + q_{2,i}) + (q_{i,3} + q_{3,i}) + (q_{i,4} + q_{4,i}) + (q_{i,5} + q_{5,i}) + (q_{i,6} + q_{6,i}) + (q_{i,7} + q_{7,i}) + (q_{i,8} + q_{8,i}) + (q_{i,9} + q_{9,i}) + (q_{i,10} + q_{10,i}) + (q_{i,11} + q_{11,i}) + (q_{i,12} + q_{12,i})] \tag{29}$$

Now let calculate  $i = 1$ .

$$h_1 = \frac{1}{4} [(q_{1,1} + q_{1,1}) + (q_{1,2} + q_{2,1}) + (q_{1,3} + q_{3,1}) + (q_{1,4} + q_{4,1}) + (q_{1,5} + q_{5,1}) + (q_{1,6} + q_{6,1}) + (q_{1,7} + q_{7,1}) + (q_{1,8} + q_{8,1}) + (q_{1,9} + q_{9,1}) + (q_{1,10} + q_{10,1}) + (q_{1,11} + q_{11,1}) + (q_{1,12} + q_{12,1})] \tag{30}$$

$$= \frac{1}{4} [(0+0) + (4+0) + (4+0) + (0+0) + (2+0) + (2+0) + (0+0) + (2+0) + (2+0) + (0+0) + (2+0) + (2+0)]$$

$$= 5$$

**Table 3.**  $J_{ij}$  interaction forces between grid neighbours. We assume that  $J_{ij} = 0$  for  $i$  and  $j$  are not adjacent.

	$J_1$	$J_2$	$J_3$	$J_4$	$J_5$	$J_6$	$J_7$	$J_8$	$J_9$	$J_{10}$	$J_{11}$	$J_{12}$
$J_1$	0	1	1	0	0.5	0.5	0	0.5	0.5	0	0.5	0.5
$J_2$	0	0	1	0	0.5	0.5	0	0.5	0.5	0	0.5	0.5
$J_3$	0	0	0	0	0.5	0.5	0	0.5	0.5	0	0.5	0.5
$J_4$	0	0	0	0	0.5	0.5	1	0.5	0.5	1	0.5	0.5
$J_5$	0	0	0	0	0	4	2	0.5	4	0.5	0.5	1
$J_6$	0	0	0	0	0	0	0.5	0.5	1	0.5	0.5	0.5
$J_7$	0	0	0	0	0	0	0	0.5	0.5	1	0.5	0.5
$J_8$	0	0	0	0	0	0	0	0	1	0.5	1	1
$J_9$	0	0	0	0	0	0	0	0	0	0.5	0.5	0.5
$J_{10}$	0	0	0	0	0	0	0	0	0	0	0.5	0.5
$J_{11}$	0	0	0	0	0	0	0	0	0	0	0	1
$J_{12}$	0	0	0	0	0	0	0	0	0	0	0	0

where  $h_1 = 5$ .

Now let calculate  $i = 2$ .

$$\begin{aligned}
 h_2 &= \frac{1}{4} \left[ (q_{2,1} + q_{1,2}) + (q_{2,2} + q_{2,2}) + (q_{2,3} + q_{3,2}) + (q_{2,4} + q_{4,2}) \right. \\
 &\quad + (q_{2,5} + q_{5,2}) + (q_{2,6} + q_{6,2}) + (q_{2,7} + q_{7,2}) + (q_{2,8} + q_{8,2}) \\
 &\quad \left. + (q_{2,9} + q_{9,2}) + (q_{2,10} + q_{10,2}) + (q_{2,11} + q_{11,2}) + (q_{2,12} + q_{12,2}) \right] \\
 &= \frac{1}{4} \left[ (0+4) + (0+0) + (4+0) + (0+0) + (2+0) + (2+0) \right. \\
 &\quad \left. + (0+0) + (2+0) + (2+0) + (0+0) + (2+0) + (2+0) \right] \\
 &= 5
 \end{aligned} \tag{31}$$

where  $h_2 = 5$ .

Now let calculate  $i = 3$ .

$$\begin{aligned}
 h_3 &= \frac{1}{4} \left[ (q_{3,1} + q_{1,3}) + (q_{3,2} + q_{2,3}) + (q_{3,3} + q_{3,3}) + (q_{3,4} + q_{4,3}) \right. \\
 &\quad + (q_{3,5} + q_{5,3}) + (q_{3,6} + q_{6,3}) + (q_{3,7} + q_{7,3}) + (q_{3,8} + q_{8,3}) \\
 &\quad \left. + (q_{3,9} + q_{9,3}) + (q_{3,10} + q_{10,3}) + (q_{3,11} + q_{11,3}) + (q_{3,12} + q_{12,3}) \right] \\
 &= \frac{1}{4} \left[ (0+4) + (0+4) + (0+0) + (0+0) + (2+0) + (2+0) \right. \\
 &\quad \left. + (0+0) + (2+0) + (2+0) + (0+0) + (2+0) + (2+0) \right] \\
 &= 5
 \end{aligned} \tag{32}$$

where  $h_3 = 5$ .

Now let calculate  $i = 4$ .

$$\begin{aligned}
 h_4 &= \frac{1}{4} \left[ (q_{4,1} + q_{1,4}) + (q_{4,2} + q_{2,4}) + (q_{4,3} + q_{3,4}) + (q_{4,4} + q_{4,4}) \right. \\
 &\quad + (q_{4,5} + q_{5,4}) + (q_{4,6} + q_{6,4}) + (q_{4,7} + q_{7,4}) + (q_{4,8} + q_{8,4}) \\
 &\quad \left. + (q_{4,9} + q_{9,4}) + (q_{4,10} + q_{10,4}) + (q_{4,11} + q_{11,4}) + (q_{4,12} + q_{12,4}) \right] \\
 &= \frac{1}{4} \left[ (0+0) + (0+0) + (0+0) + (0+0) + (2+0) + (2+0) \right. \\
 &\quad \left. + (4+0) + (2+0) + (2+0) + (4+0) + (2+0) + (2+0) \right] \\
 &= 5
 \end{aligned} \tag{33}$$

where  $h_4 = 5$ .

Now let calculate  $i = 5$ .

$$\begin{aligned}
 h_5 &= \frac{1}{4} \left[ (q_{5,1} + q_{1,5}) + (q_{5,2} + q_{2,5}) + (q_{5,3} + q_{3,5}) + (q_{5,4} + q_{4,5}) \right. \\
 &\quad + (q_{5,5} + q_{5,5}) + (q_{5,6} + q_{6,5}) + (q_{5,7} + q_{7,5}) + (q_{5,8} + q_{8,5}) \\
 &\quad \left. + (q_{5,9} + q_{9,5}) + (q_{5,10} + q_{10,5}) + (q_{5,11} + q_{11,5}) + (q_{5,12} + q_{12,5}) \right] \\
 &= \frac{1}{4} \left[ (0+2) + (0+2) + (0+2) + (0+2) + (0+0) + (4+0) \right. \\
 &\quad \left. + (0+2) + (2+0) + (4+0) + (2+0) + (2+0) + (4+0) \right] \\
 &= 7
 \end{aligned} \tag{34}$$

where  $h_5 = 7$ .

Now let calculate  $i = 6$ .

$$\begin{aligned}
 h_6 &= \frac{1}{4} \left[ (q_{6,1} + q_{1,6}) + (q_{6,2} + q_{2,6}) + (q_{6,3} + q_{3,6}) + (q_{6,4} + q_{4,6}) \right. \\
 &\quad + (q_{6,5} + q_{5,6}) + (q_{6,6} + q_{6,6}) + (q_{6,7} + q_{7,6}) + (q_{6,8} + q_{8,6}) \\
 &\quad \left. + (q_{6,9} + q_{9,6}) + (q_{6,10} + q_{10,6}) + (q_{6,11} + q_{11,6}) + (q_{6,12} + q_{12,6}) \right] \\
 &= \frac{1}{4} \left[ (0+2) + (0+2) + (0+2) + (0+2) + (0+4) + (0+0) \right. \\
 &\quad \left. + (2+0) + (2+0) + (4+0) + (2+0) + (2+0) + (2+0) \right] \\
 &= \frac{13}{2}
 \end{aligned} \tag{35}$$

where  $h_6 = 6.5$ .

Now let calculate  $i = 7$ .

$$\begin{aligned}
 h_7 &= \frac{1}{4} \left[ (q_{7,1} + q_{1,7}) + (q_{7,2} + q_{2,7}) + (q_{7,3} + q_{3,7}) + (q_{7,4} + q_{4,7}) \right. \\
 &\quad + (q_{7,5} + q_{5,7}) + (q_{7,6} + q_{6,7}) + (q_{7,7} + q_{7,7}) + (q_{7,8} + q_{8,7}) \\
 &\quad \left. + (q_{7,9} + q_{9,7}) + (q_{7,10} + q_{10,7}) + (q_{7,11} + q_{11,7}) + (q_{7,12} + q_{12,7}) \right] \\
 &= \frac{1}{4} \left[ (0+0) + (0+0) + (0+0) + (0+4) + (0+2) + (0+2) \right. \\
 &\quad \left. + (0+0) + (2+0) + (2+0) + (4+0) + (2+0) + (2+0) \right] \\
 &= 5
 \end{aligned} \tag{36}$$

where  $h_7 = 5$ .

Now let calculate  $i = 8$ .

$$\begin{aligned}
 h_8 &= \frac{1}{4} \left[ (q_{8,1} + q_{1,8}) + (q_{8,2} + q_{2,8}) + (q_{8,3} + q_{3,8}) + (q_{8,4} + q_{4,8}) \right. \\
 &\quad + (q_{8,5} + q_{5,8}) + (q_{8,6} + q_{6,8}) + (q_{8,7} + q_{7,8}) + (q_{8,8} + q_{8,8}) \\
 &\quad \left. + (q_{8,9} + q_{9,8}) + (q_{8,10} + q_{10,8}) + (q_{8,11} + q_{11,8}) + (q_{8,12} + q_{12,8}) \right] \\
 &= \frac{1}{4} \left[ (0+2) + (0+2) + (0+2) + (0+2) + (0+2) + (0+2) \right. \\
 &\quad \left. + (0+2) + (0+0) + (4+0) + (2+0) + (4+0) + (4+0) \right] \\
 &= 7
 \end{aligned} \tag{37}$$

where  $h_8 = 7$ .

Now let calculate  $i = 9$ .

$$\begin{aligned}
 h_9 &= \frac{1}{4} \left[ (q_{9,1} + q_{1,9}) + (q_{9,2} + q_{2,9}) + (q_{9,3} + q_{3,9}) + (q_{9,4} + q_{4,9}) \right. \\
 &\quad + (q_{9,5} + q_{5,9}) + (q_{9,6} + q_{6,9}) + (q_{9,7} + q_{7,9}) + (q_{9,8} + q_{8,9}) \\
 &\quad \left. + (q_{9,9} + q_{9,9}) + (q_{9,10} + q_{10,9}) + (q_{9,11} + q_{11,9}) + (q_{9,12} + q_{12,9}) \right] \\
 &= \frac{1}{4} \left[ (0+2) + (0+2) + (0+2) + (0+2) + (0+4) + (0+4) \right. \\
 &\quad \left. + (0+2) + (0+4) + (0+0) + (2+0) + (2+0) + (2+0) \right] \\
 &= 7
 \end{aligned} \tag{38}$$

where  $h_9 = 7$ .

Now let calculate  $i = 10$ .

$$\begin{aligned}
 h_{10} &= \frac{1}{4} \left[ (q_{10,1} + q_{1,10}) + (q_{10,2} + q_{2,10}) + (q_{10,3} + q_{3,10}) + (q_{10,4} + q_{4,10}) \right. \\
 &\quad + (q_{10,5} + q_{5,10}) + (q_{10,6} + q_{6,10}) + (q_{10,7} + q_{7,10}) + (q_{10,8} + q_{8,10}) \\
 &\quad \left. + (q_{10,9} + q_{9,10}) + (q_{10,10} + q_{10,10}) + (q_{10,11} + q_{11,10}) + (q_{10,12} + q_{12,10}) \right] \\
 &= \frac{1}{4} \left[ (0+0) + (0+0) + (0+0) + (0+4) + (0+2) + (0+2) \right. \\
 &\quad \left. + (0+4) + (0+2) + (0+2) + (0+0) + (2+0) + (2+0) \right] \\
 &= \frac{9}{2}
 \end{aligned} \tag{39}$$

where  $h_{10} = 4.5$ .

Now let calculate  $i = 11$ .

$$\begin{aligned}
 h_{11} &= \frac{1}{4} \left[ (q_{11,1} + q_{1,11}) + (q_{11,2} + q_{2,11}) + (q_{11,3} + q_{3,11}) + (q_{11,4} + q_{4,11}) \right. \\
 &\quad + (q_{11,5} + q_{5,11}) + (q_{11,6} + q_{6,11}) + (q_{11,7} + q_{7,11}) + (q_{11,8} + q_{8,11}) \\
 &\quad \left. + (q_{11,9} + q_{9,11}) + (q_{11,10} + q_{10,11}) + (q_{11,11} + q_{11,11}) + (q_{11,12} + q_{12,11}) \right] \\
 &= \frac{1}{4} \left[ (0+2) + (0+2) + (0+2) + (0+2) + (0+2) + (0+2) \right. \\
 &\quad \left. + (0+2) + (0+4) + (0+2) + (0+2) + (0+0) + (4+0) \right] \\
 &= \frac{13}{2}
 \end{aligned} \tag{40}$$

where  $h_{11} = 6.5$ .

Now let calculate  $i = 12$ .

$$\begin{aligned}
 h_{12} &= \frac{1}{4} \left[ (q_{12,1} + q_{1,12}) + (q_{12,2} + q_{2,12}) + (q_{12,3} + q_{3,12}) + (q_{12,4} + q_{4,12}) \right. \\
 &\quad + (q_{12,5} + q_{5,12}) + (q_{12,6} + q_{6,12}) + (q_{12,7} + q_{7,12}) + (q_{12,8} + q_{8,12}) \\
 &\quad \left. + (q_{12,9} + q_{9,12}) + (q_{12,10} + q_{10,12}) + (q_{12,11} + q_{11,12}) + (q_{12,12} + q_{12,12}) \right] \\
 &= \frac{1}{4} \left[ (0+2) + (0+2) + (0+2) + (0+2) + (0+4) + (0+2) \right. \\
 &\quad \left. + (0+2) + (0+4) + (0+2) + (0+2) + (0+4) + (0+0) \right] \\
 &= 7
 \end{aligned} \tag{41}$$

where  $h_{12} = 7$ .

With the calculated  $J_{ij}$  and  $h_i$  (Table 2 and Table 4), we can now solve our Social Worker Problem with VQE ( $\langle \psi(\theta) | H | \psi(\theta) \rangle$ ) or with the Quantum Exact Solver or with any variational method. We are considering that the superposition state ( $N$  qubits) of  $Q = q_1 \cdots q_N$  is described by  $|\psi\rangle = |\psi_1\rangle \cdots |\psi_N\rangle$ .

$$H_P |\psi\rangle = \left( \sum_i h_i \sigma_i^z + \sum_{i < j} J_{ij} \sigma_i^z \sigma_j^z \right) |\psi\rangle \tag{42}$$

Where the energy function is as follows and where the notation  $\sigma_i^z$  means that the Pauli-Z operator is applied to the single-qubit  $|\psi\rangle = |\psi_1 \cdots \psi_i \cdots\rangle$ .

**Table 4.** Calculated values of the external force  $h_i$ .

$h_1$	5
$h_2$	5
$h_3$	5
$h_4$	5
$h_5$	7
$h_6$	6,5
$h_7$	5
$h_8$	7
$h_9$	7
$h_{10}$	4,5
$h_{11}$	6,5
$h_{12}$	7

$$\epsilon_j = \sum_{i \in V} h_i (-1)^{w_i} + \sum_{(i,j) \in V} J_{ij} (-1)^{w_i} (-1)^{w_j} \tag{43}$$

with

$$\sigma_i^z \rightarrow \prod_{1^{st} \text{ position}} \otimes \dots \otimes \underbrace{\sigma^z}_{i^{th} \text{ position}} \otimes \dots \otimes \prod_{N^{th} \text{ position}} \tag{44}$$

and

$$\sigma_i^z \sigma_j^z \xrightarrow{i < j} \prod_{1^{st} \text{ position}} \otimes \dots \otimes \underbrace{\sigma^z}_{i^{th} \text{ position}} \otimes \dots \otimes \underbrace{\sigma^z}_{j^{th} \text{ position}} \otimes \dots \otimes \prod_{N^{th} \text{ position}} \tag{45}$$

### 7. Results and Discussion

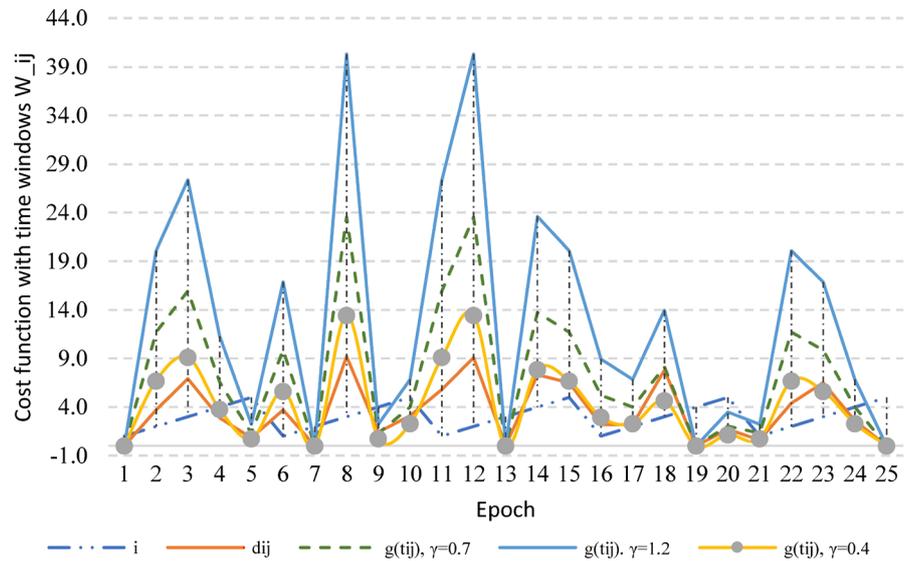
The first observation is that both the proposed cost function and the social workers’ problem formulation behave much better than we expected. Form the posed cost function, the quadratic function (27) acts very well for small numbers and also for large quantities. For intermediate numbers of  $\gamma$ , we use the value of  $\gamma = 0.73$  to correct the resulting function. **Figure 2** describes the analysis done in terms of  $\gamma$ .

The second comment is that we have considered a specific problem with a very generic vision that combines both a routing and scheduling problem. We have presented and detailed it to step by step in the QUBO formulation and to adapt the limitation of the latter. But we were making our formulation generic and above all making the time restrictions (time window) in the form of social workers’ schedules not grow proportionally in the number of qubits. This allows us to use the few useful qubits that we have for the most suitable operations.

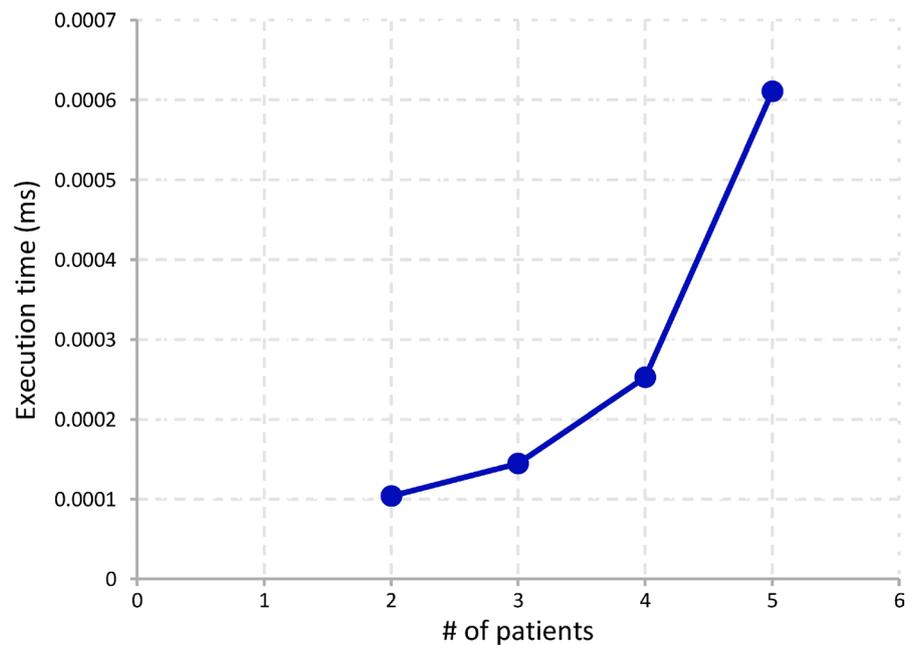
The formulation that we have provided is feasible for any quantum architecture, however, due to the feasibility offered by IBMQ and its very active com-

munity, the viability of the code and experimentation have been done on Qiskit under its library aqua<sup>3</sup>.

All experiments performed, 100 in total, we have seen that the exact quantum solver behaves in the same way, in terms of the output result, as the classical solver that we have used (MIP Solver). On the one hand, the results obtained are

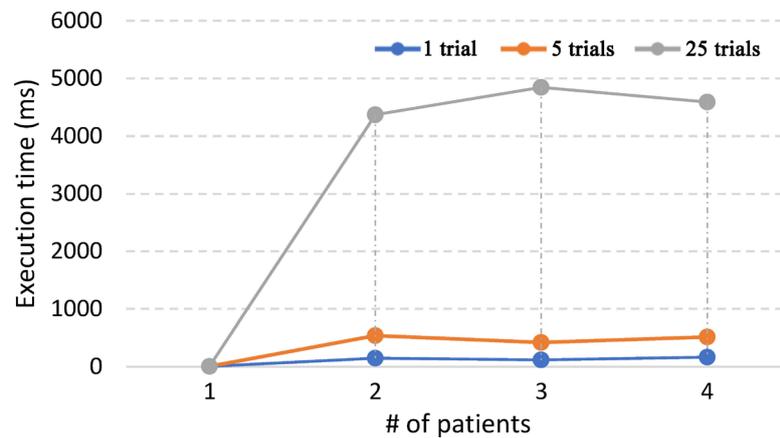


**Figure 2.** Comparison of performance with standard deviation error bar on the three mappings of our proposed formulation ((11), (12)). The Standard Deviation expected total anneal time for 99% percent success for each mapping value, with the best  $\epsilon$  for each shoot are shown. Our optimal case is for  $\gamma = 0.7$ . Our most representative cases are for  $\gamma = 0.4$ ,  $\gamma = 0.7$  and  $\gamma = 1.2$ .



**Figure 3.** Backtracking’s behaviour depending on the execution time and the number of patients.

<sup>3</sup>[https://qiskit.org/documentation/apidoc/qiskit\\_aqua.html](https://qiskit.org/documentation/apidoc/qiskit_aqua.html).



**Figure 4.** VQE's behaviour depending on the execution time and the number of patients and number of trials.

identical in more significant computational time for the quantum Exact Solver and QASM\_simulator. Conceptually it is expected, since the priority of the era of quantum computing in which we are (NISQ), the most important thing is to compare the complexity and the good functioning of the quantum algorithm, not its execution time. **Figure 3** and **Figure 4** plot what we just mentioned.

On the other hand, the QASM\_simulator, to obtain the same result we need to increase the number of seeds considerably up to 10,598 and max\_trials close to 2000 within the VQE configuration. Therefore, we can see that in the case of Monday and Wednesday (**Figure 5**), the algorithm finds three paths instead of the two that it should have found.

It should be added that **Figures 5-7** are the results for the case of 5 patients and two social workers with the values of seed = 10,598 and max\_trials = 300 for the QASM\_simulator.

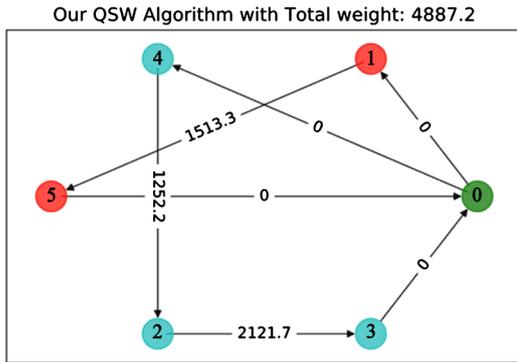
The three banks of figures show us the results when using the classical solver as a cplex, or backtracking, when using the exact quantum solver and when using the ibmq\_qasm\_simulator from IBMQ<sup>4</sup>.

We can see that the three results are identical, which ensures that our algorithm works well. We can observe that our quantum algorithm responds to the classical solver. And in the experiments we did (**Figure 3** and **Figure 4**), we were able to attend that, while the classical solver exhibits exponential behaviour as the number of patients (which would be the nodes of the graph) increases, the VQE algorithm, without taking into account the cost of evaluation and algorithm calibration, has logarithm growth. This, as the number of patients grows, will offer more significant advantages than a classic algorithm, such as Backtracking, since its temporary cost will be much lower for more complex problems. In other words, the quantum algorithm has a linear complexity compared to a classical solver.

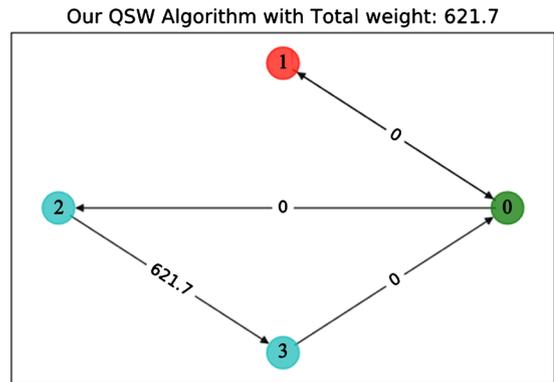
### 7.1. Results from QUBO Form with Classical Solver

<sup>4</sup><https://quantum-computing.ibm.com/>.

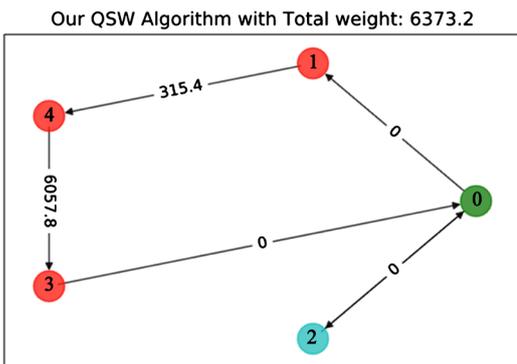
Social Worker TimeTable  
 -----  
 Social Worker 1:  
 Patient 1: 09:00-10:00  
 Patient 5: 11:45-12:45  
 Social Worker 2:  
 Patient 4: 09:30-10:30  
 Patient 2: 12:00-13:00  
 Patient 3: 15:15-16:15



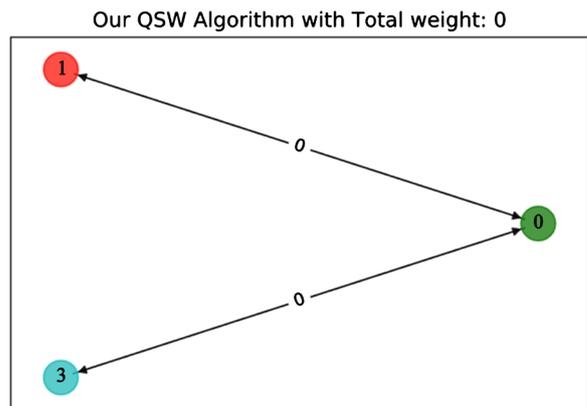
Social Worker TimeTable  
 -----  
 Social Worker 1:  
 Patient 1: 09:00-10:00  
 Social Worker 2:  
 Patient 2: 14:45-15:45  
 Patient 3: 16:30-17:30



Social Worker TimeTable  
 -----  
 Social Worker 1:  
 Patient 1: 09:00-10:00  
 Patient 4: 10:15-11:15  
 Patient 3: 15:45-16:45  
 Social Worker 2:  
 Patient 2: 16:00-17:00



Social Worker TimeTable  
 -----  
 Social Worker 1:  
 Patient 1: 09:00-10:00  
 Social Worker 2:  
 Patient 3: 15:00-16:00



Social Worker TimeTable  
 -----  
 Social Worker 1:  
 Patient 1: 09:00-10:00  
 Patient 5: 11:30-12:30  
 Patient 3: 17:00-18:00  
 Social Worker 2:  
 Patient 4: 11:30-12:30

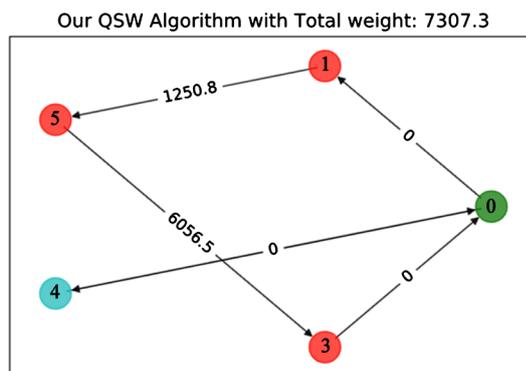
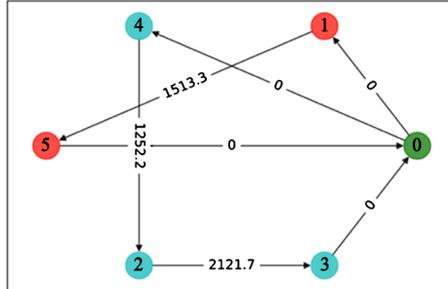


Figure 5. Result of the social workers' algorithm by using classical solver.

## 7.2. Results from Ising Model with Quantum Exact Eigen Solver

Result of Monday by using Quantum ExactEigensolver

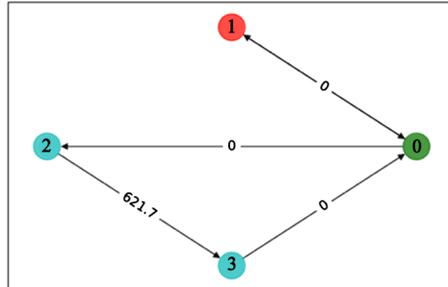


Result of Monday's path  
 energy: -499459.44999999984  
 objective: 54091.80000000214  
 solution: [1 1 0 0 0 1 0 0 0 0 1 0 1 0 0 1 1]  
 Social Worker TimeTable

Sociak Worker 1:  
 Patient 1: 09:00-10:00  
 Patient 5: 11:45-12:45

Sociak Worker 2:  
 Patient 4: 09:30-10:30  
 Patient 2: 12:00-13:00  
 Patient 3: 15:15-16:15

Result of Tuesday by using Quantum ExactEigensolver

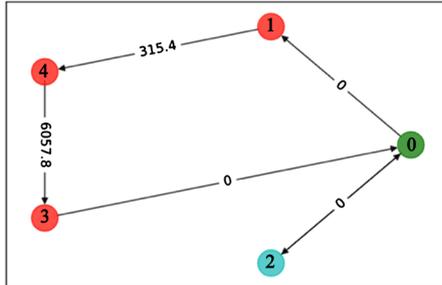


Result of Tuesday's path  
 energy: -119600.49999999996  
 objective: 37613.299999999916  
 solution: [1 1 0 1 0 0 0 1 1]  
 Social Worker TimeTable

Sociak Worker 1:  
 Patient 1: 09:00-10:00

Sociak Worker 2:  
 Patient 2: 14:45-15:45  
 Patient 3: 16:30-17:30

Result of Wednesday by using Quantum ExactEigensolver

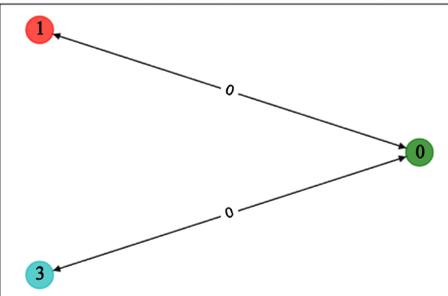


Result of Wednesday's path  
 energy: -360614.3  
 objective: 70197.20000000001  
 solution: [1 0 0 1 0 1 0 0 0 1 0 1 1]  
 Social Worker TimeTable

Sociak Worker 1:  
 Patient 1: 09:00-10:00  
 Patient 4: 10:15-11:15  
 Patient 3: 15:45-16:45

Sociak Worker 2:  
 Patient 2: 16:00-17:00

Result of Thursday by using Quantum ExactEigensolver

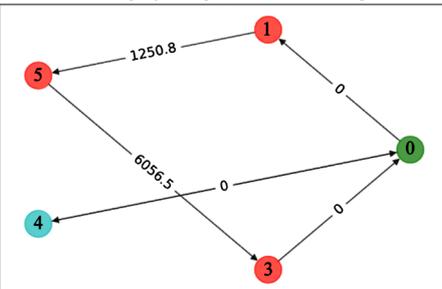


Result of Thursday's path  
 energy: -32435.049999999992  
 objective: 14415.799999999997  
 solution: [1 1 1 0 1]  
 Social Worker TimeTable

Sociak Worker 1:  
 Patient 1: 09:00-10:00

Sociak Worker 2:  
 Patient 3: 15:00-16:00

Result of Friday by using Quantum ExactEigensolver



Result of Friday's path  
 energy: -308090.05000000005  
 objective: 62159.099999999744  
 solution: [1 0 1 0 0 1 0 0 0 1 1 0 1]  
 Social Worker TimeTable

Sociak Worker 1:  
 Patient 1: 09:00-10:00  
 Patient 5: 11:30-12:30  
 Patient 3: 17:00-18:00

Sociak Worker 2:  
 Patient 4: 11:30-12:30

Figure 6. Result of the social workers' algorithm by using the quantum exact eigensolver.

### 7.3. Results from the Ising Model with Quantum ASM Simulator

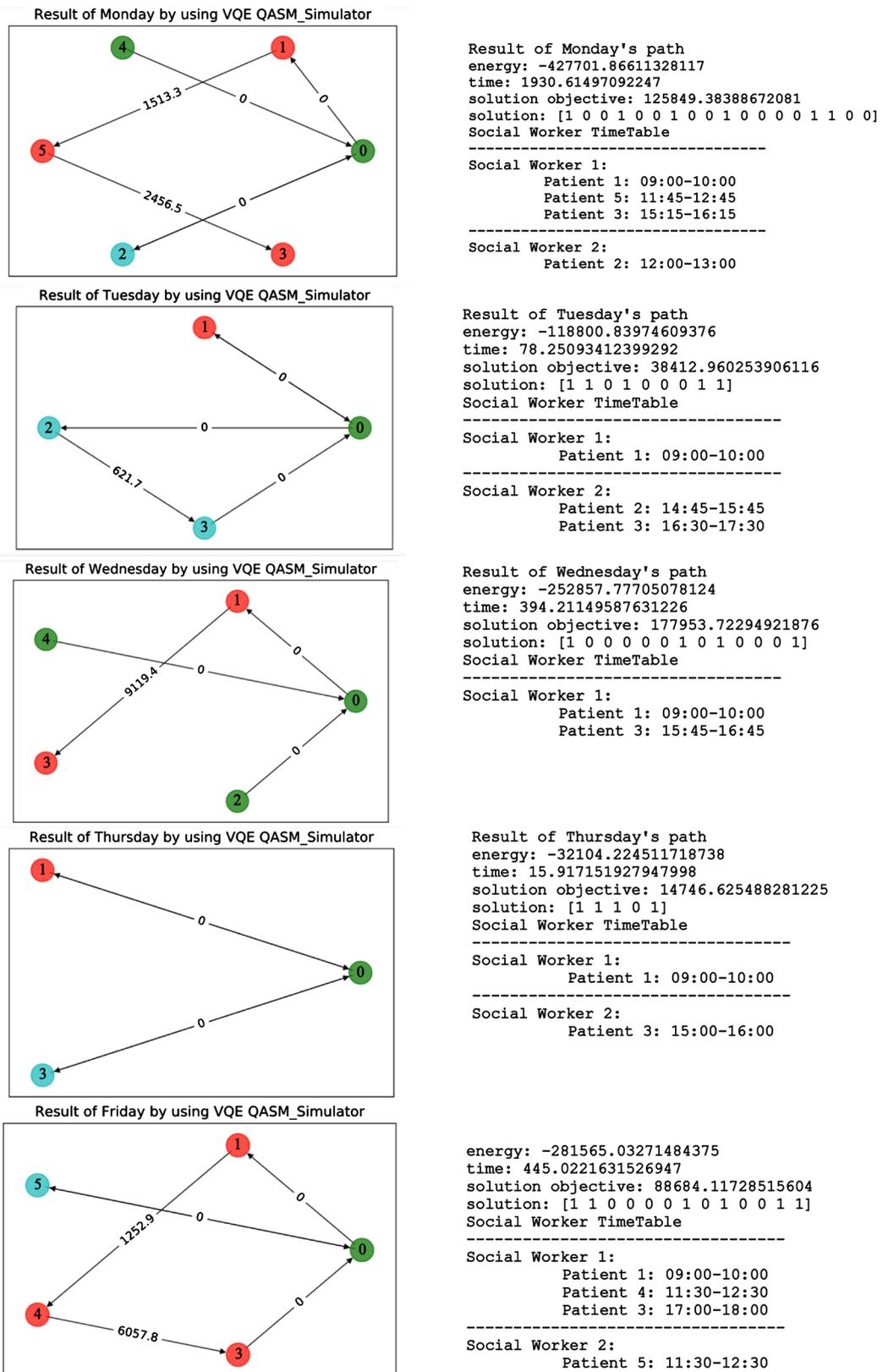


Figure 7. Result of the social workers' algorithm by using the ibmq\_simulator.

## 8. Conclusions and Further Work

In this work, we have defined, written and solved a time-constrained combinatorial optimization problem in the form QUBO. The question of social workers has inequality constraints that are very difficult to tackle within the QUBO framework. To do this, we have defined a strategy to solve it in this NISQ era. We found the quantum efficient solution using Exact solver and VQE.

The future direction of this work will be modelling the social workers' problem into the IBM docplex. And compare the complexity of quantum algorithms in the face of the classic ones. Also, it is worth saying that our proposal is not limited exclusively to the case of social workers, but to all combinatorial optimization problems that require routing and scheduling features.

## Acknowledgements

The authors greatly thank the IBMQ team and, mainly Steve Wood and Jennifer Ramírez Molino for their valuable discussions that have made this work possible.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

- [1] Clarke, G. and Wright, J.W. (1964) Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, **12**, 568-581.  
<https://doi.org/10.1287/opre.12.4.568>
- [2] Laporte, G. (1992) The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithm. *European Journal of Operational Research*, **59**, 345-358.  
[https://doi.org/10.1016/0377-2217\(92\)90192-C](https://doi.org/10.1016/0377-2217(92)90192-C)
- [3] Toth, P. and Vigo, D. (2002) The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia.  
<https://doi.org/10.1137/1.9780898718515>
- [4] Bożejko, W., Pempera, J. and Smutnicki, C. (2009) Parallel Simulated Annealing for the Job Shop Scheduling Problem. *International Conference on Computational Science*, Baton Rouge, 25-27 May 2009, 631-640.  
[https://doi.org/10.1007/978-3-642-01970-8\\_62](https://doi.org/10.1007/978-3-642-01970-8_62)
- [5] Brucker, P. (2007) Scheduling Algorithms. Springer, New York.
- [6] Wolf, R.D. (2019) Quantum Computing: Lecture Notes. QuSoft, CWI and University of Amsterdam, Amsterdam.
- [7] McGeoch, C.C. (2014) Adiabatic Quantum Computation and Quantum Annealing: Theory and Practice (Synthesis Lectures on Quantum Computing). Morgan & Claypool, San Rafael. <https://doi.org/10.2200/S00585ED1V01Y201407QMC008>
- [8] Kadowaki, T. and Nishimori, H. (2008) Quantum Annealing in the Transverse Ising Model. Department of Physics, Tokyo Institute of Technology, O-okayama, Me-guro-ku, Tokyo, 152-8551.
- [9] Bertels, K., Sarkar, A., Mouedenne, A., Hubregtsen, T., Yadav, A., Krol, A. and

- Ashraf, I. (2019) Quantum Computer Architecture: Towards Full-Stack Quantum Accelerators. *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Grenoble, 9-13 March 2020, 1-6.  
<https://doi.org/10.23919/DATE48585.2020.9116502>
- [10] Fowler, A. (2017) Improve QUBO Formulation for D-Wave Quantum Computing.
- [11] Preskill, J. (2018) Quantum Computing in the NISQ Era and Beyond. *Quantum: The Open Journal for Quantum Science*, **2**, 79.  
<https://doi.org/10.22331/q-2018-08-06-79>
- [12] Nielsen, M.A. and Chuang, I.L. (2000) Quantum Computation and Quantum Information. Cambridge University Press, Cambridge.
- [13] McDonald, K.T. (2017) Ph410 Physics of Quantum Computation. Princeton University, Princeton.
- [14] Rieffel, E.G., Venturelli, D., O’Gorman, B., Do, M.B., Prystay, E. and Smelyanskiy, V.N. (2014) A Case Study in Programming a Quantum Annealer for Hard Operational Planning Problems. *Quantum Information Processing*, **14**, 1-36.  
<https://doi.org/10.1007/s11128-014-0892-x>
- [15] Peruzzo, A., McClean, J., Shadbolt, P., Yung, M.-H., Zhou, X.-Q., Love, P.J., Aspuru-Guzik, A. and O’Brien, J.L. (2013) A Variational Eigenvalue Solver on a Quantum Processor. *Nature Communications*, **5**, Article No. 4213.  
<https://doi.org/10.1038/ncomms5213>
- [16] Wang, D., Higgott, O. and Brierley, S. (2019) Accelerated Variational Quantum Eigensolver. *Physical Review Letters*, **122**, Article ID: 140504.  
<https://doi.org/10.1103/PhysRevLett.122.140504>
- [17] Lechner, W., Hauke, P. and Zoller, P. (2015) A Quantum Annealing Architecture with All-to-All Connectivity from Local Interactions. *Science Advances*, **1**, e1500838.  
<https://doi.org/10.1126/sciadv.1500838>
- [18] Tran, T.T., Do, M., Rieffel, E.G., Frank, J., Wang, Z., O’Gorman, B., Venturelli, D. and Beck, J. (2016) A Hybrid Quantum-Classical Approach to Solving Scheduling Problems. *Proceedings of the Ninth International Symposium on Combinatorial Search*, Vancouver, 20 June 2016, 98-106.
- [19] Grimsley, H.R., Economou, S.E., Barnes, E. and Mayhall, N.J. (2019) An Adaptive Variational Algorithm for Exact Molecular Simulations on a Quantum Computer. *Nature Communications*, **10**, 3007. <https://doi.org/10.1038/s41467-019-10988-2>
- [20] Wu, C.W. (2005) On Rayleigh-Ritz Ratios of a Generalized Laplacian Matrix of Directed Graphs. *Linear Algebra and Its Applications*, **402**, 207-227.  
<https://doi.org/10.1016/j.laa.2004.12.014>
- [21] Lewis, M., Kochenberger, G. and Metcalfe, J. (2019) Robust Optimization of Unconstrained Binary Quadratic Problems. <https://doi.org/10.1504/IJOR.2019.104050>