

Neural-powered unit disk graph embedding: qubits connectivity for some QUBO problems

Original

Neural-powered unit disk graph embedding: qubits connectivity for some QUBO problems / Vercellino, C; Viviani, P; Vitali, G; Scionti, A; Scarabosio, A; Terzo, O; Giusto, E; Montrucchio, B. - (2022), pp. 186-196. (Intervento presentato al convegno 2022 IEEE International Conference on Quantum Computing and Engineering (QCE) tenutosi a Broomfield, CO (USA) nel September 18–23, 2022) [10.1109/QCE53715.2022.00038].

Availability:

This version is available at: 11583/2974759 since: 2023-01-18T09:16:40Z

Publisher:

IEEE COMPUTER SOC

Published

DOI:10.1109/QCE53715.2022.00038

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2022 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

Neural-powered unit disk graph embedding: qubits connectivity for some QUBO problems

Chiara Vercellino^{1*} Paolo Viviani¹ Giacomo Vitali¹ Alberto Scionti¹
Andrea Scarabosio¹ Olivier Terzo¹ Edoardo Giusto²
Bartolomeo Montrucchio²

¹*LINKS Foundation, Torino, Italy*

²*DAUIN, Politecnico di Torino, Torino, Italy*

**chiara.vercellino@linksfoundation.com*

November 25, 2022

Abstract

Graph embedding is a recurrent problem in quantum computing, for instance, quantum annealers need to solve a minor graph embedding in order to map a given Quadratic Unconstrained Binary Optimization (QUBO) problem onto their internal connectivity pattern. This work presents a novel approach to constrained *unit disk* graph embedding, which is encountered when trying to solve combinatorial optimization problems in QUBO form, using quantum hardware based on neutral Rydberg atoms. The qubits, physically represented by the atoms, are excited to the Rydberg state through laser pulses. Whenever qubits pairs are closer together than the blockade radius, entanglement can be reached, thus preventing entangled qubits to be simultaneously in the excited state. Hence, the blockade radius determines the adjacency pattern among qubits, corresponding to a unit disk configuration. Although it is straightforward to compute the adjacency pattern given the qubits' coordinates, identifying a feasible unit disk arrangement that matches the desired QUBO matrix is, on the other hand, a much harder task. In the context of quantum optimization, this issue translates into the physical placement of the qubits in the 2D/3D register to match the machine's Ising-like

Hamiltonian with the QUBO formulation of the optimization problems. The proposed solution exploits the power of neural networks to transform an initial embedding configuration, which does not match the quantum hardware requirements or does not account for the unit disk property, into a feasible embedding properly representing the target optimization problems. Experimental results show that this new approach overcomes in performance *Gurobi* solver.

Keywords:

Quantum computing
Graphs
Embedding
Neural networks
Optimization

1 Introduction

Quantum computers bear promises as tools to accelerate specific computations like prime factorization, eigenvalues decomposition, combinatorial problem-solving etc. However, in the noisy intermediate-scale quantum (NISQ) era, practical applications of canonical quantum algorithms (*e.g.*, Shor's algorithm) are still out of reach due to low qubits

count, limited coherence time, and gates fidelity; in this context different approaches emerged, like hybrid quantum-classical algorithms and specialised architectures (*e.g.*, *quantum annealers*, *quantum simulators*), that showed promising results for specific problems, but still present a number of implementation issues.

The goal of this work is to provide a suitable methodology to port real-life applications to these kinds of machines, with a specific focus on solving discrete optimization problems on optical quantum simulators based on neutral atoms, by developing a Machine Learning (ML) technique for embedding graphs into the suitable machine representation (register).

This quantum computing architecture operates at room temperature and it is, in principle, flexible enough to implement quantum gates as well as manipulate a given target system Hamiltonian through laser pulses [1]. In this paper, gates-based operations will not be considered, instead, attention will be paid to the analog quantum processing mode, which involves the use of laser pulses to induce dipole-dipole interactions between Rubidium atoms (*i.e.*, Rydberg atoms [2]) in *Rydberg state*, arranged arbitrarily in 2D or 3D arrays.

The interactions between Rydberg atoms can, in turn, be mapped into a spin Hamiltonian (*e.g.*, Ising): this capability of neutral atoms machines is interesting because a significant set of NP-hard combinatorial optimization problems can be mapped to the Ising Hamiltonian [3] or to the equivalent Quadratic Unconstrained Binary Optimization (QUBO) forms [4], popularized by Quantum Annealers. In principle, once a set of Rydberg atoms is organised in the space to reproduce the desired Hamiltonian, it is possible to solve the associated optimization problem using the Quantum Approximate Optimization Algorithm [5], [6].

While the idea is somewhat straightforward, the hardware requirements of the experimental machine characterize the optimization problems that can be approached. The starting point is to model the optimization problem at hand in the QUBO form, which means building a square matrix $Q \in \mathbb{R}^{n \times n}$ whose dimensionality n is determined by the number of binary variables. In the proposed embedding solution, each

binary variable is represented by a physical qubit. The diagonal elements of Q identify the penalty/gain associated with each binary variable when it assumes the value 1; due to the restriction of the machine lasers to operate only at the global level (*i.e.*, exciting all atoms with the same Rabi frequency [1]), these diagonal elements should have all the same value. The off-diagonal elements of Q represent quadratic interactions between the variables, which are represented by atoms coupled through the Rydberg blockade effect [7] which scales as $\propto 1/d_{ij}^6$, with d_{ij} the Euclidean distance between two qubits i and j . Thus, as the distance is necessarily positive, the off-diagonal elements of Q should have all the same sign.

If a QUBO problem satisfies the requirements above, it is in principle straightforwardly mapped into the effective Hamiltonian that the real machine is able to reproduce

$$\mathcal{H} = \sum_{i=1}^n \frac{\hbar\Omega}{2} \sigma_i^x - \sum_{i=1}^n \frac{\hbar\delta}{2} \sigma_i^z + \sum_{j>i} \frac{C_6}{d_{ij}^6} n_i n_j \quad (1)$$

where $n_i = (1 + \sigma_i^z)/2$ is the Rydberg state occupancy, $\sigma_i^{x,z}$ are the Pauli matrices of the spin of i -th qubit, Ω is the Rabi frequency of the laser pulse and δ is the detuning of the laser pulse. While a feasible formulation of the QUBO matrix based on these constraints can be achieved (*e.g.*, for graph coloring problems [4]), the magnitude of the off-diagonal elements of Q is bound to the two-qubits coupling coefficient C_6/d_{ij}^6 , therefore to the relative distances of qubits in a 2D/3D register. In particular, a threshold-like effect is reached when the distances between qubits' pairs are shorter than the blockade radius, *i.e.*, a critical distance at which the strength of the interactions balances with the Rabi frequency, thus yielding opposite entanglement [8] on neutral atoms: atoms with a pair distance shorter than the blockade radius cannot be simultaneously in the excited state.. Therefore, the interactions between qubits in the register can be represented by a unit disk (UD) graph [9] as in figure 1, and the problem of mapping an arbitrary QUBO/Ising problem onto a neutral atoms quantum simulator can be formulated as a unit disk graph embedding problem.

This paper will describe a novel heuristic to obtain

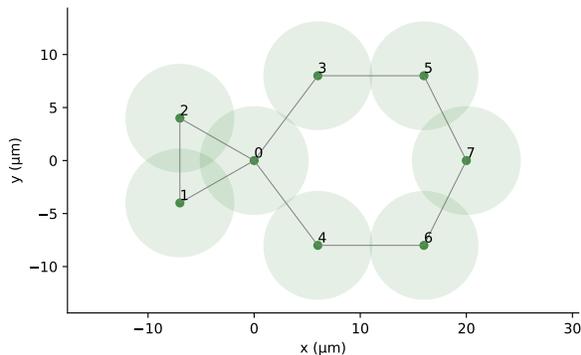


Figure 1: Representation of a qubit register composed of neutral atoms arranged in a 2D array. Interaction between qubits is regulated by the Rydberg blockade effect: only atoms falling within the Rydberg radius are subject to interaction.

the coordinates of a unit disk embedding for a given QUBO problem.

2 Related Work

The qubits placement problem is well-known in the context of quantum computing. Despite the specific technologies, the design of a proper embedding to exploit the hardware potential is of major importance. In the particular case of neutral atoms based devices, the embedding for QUBO problems has been approached in different ways.

In [6], non-convex constrained formulations of the UD graph problem are provided, anyway, solving the UD graph problem with classical solvers can become pretty difficult as the size of the QUBO problems increases or the adjacency pattern gets more complicated: the computation of the unit disk graph could become possibly harder than retrieving the QUBO problem solution. To fully understand the complexity of the unit disk graph problem, it is important to notice that even the corresponding recognition problem (*i.e.*, determining if a given graph has a realization that maps vertices to points of a unit disk configuration) is NP-hard [10].

Further results on UD graphs are presented in [11]

with a focus on the non-approximability of the unit disk (and d-quasi unit disk [12]) graphs, moreover, a bound on the quality of the embedding is computed. In [13], S. Bhore et al. investigate the unit disk graph recognition problem for subclasses of planar graphs, stating that even for outerplanar [14] and trees [15] graphs this task is NP-hard.

Given the complexity of the problem, to overcome the limitations of an exact solution, embedding heuristics have been proposed, but their application to the neutral atoms quantum hardware generally needs more physical qubits or greater sophistication in the hardware design. Several studies approach the embedding issue through local fields' manipulation. In [16], local interactions enhance the quantum annealing architecture with controllable all-to-all connectivity between logical qubits, the physical qubits are instead placed according to a square-lattice geometry. In [17], the authors illustrate a three-dimensional embedding based on Ising ferromagnetic quantum wires to couple distant qubits and produce a regular cubic lattice architecture. As additional requirements to build quantum wires, multiple physical qubits are used to represent a single variable and local fields play a key role. Further work on quantum wires is presented in [18], where Kim et al. overcome the issue of local fields in quantum wires for maximum independent set problems, anyway, the proposed architecture still requires auxiliary wire atoms in a 3D space and constrains the embedding to accomplish constant distances among adjacent qubits, thus introducing a discretization of the embedding space and not exploiting the full potential of the blockade effect.

QUBO problems' embedding has also been approached through graph minors [19] search. The D-Wave system exploits heuristic techniques to find graph minors with a probabilistic setting to avoid exhaustive search [20]. Their quantum hardware is designed with a fixed topology where qubits adjacency is modelled through couplers. The mapping of the QUBO problems is feasible whenever the graph of binary optimization variables interactions is a minor of the graph of qubit interactions in the D-Wave hardware. A similar approach was presented also by V. Choi [21], with a focus on embedding Ising Hamiltonian for solving the maximum independent set prob-

lem via adiabatic quantum computation. Again the reduction through minor-embedding in the quantum hardware graph relies on parameters settings of qubit biases and coupler strengths.

3 Methodology

Starting from a properly representable, *i.e.*, compliant to the Hamiltonian formulation of eq. (1), QUBO model, a feasible embedding for the qubits should be retrieved. The adjacency matrix A , associated with matrix Q , defines the adjacency pattern among the qubits, but the concept of proximity/distance of the atoms in the register cannot be defined at will. Qubits placement has to satisfy strong requirements in order to reproduce the blockade effect that underlies the entanglement between them and, therefore to enhance optimization problems solution with quantum technology.

To model these embedding constraints, the following notation will be used: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is the graph associated with matrix A , with \mathcal{V} vertexes and \mathcal{E} edges, $\mathcal{G}' = (\mathcal{V}, \bar{\mathcal{E}})$ is the complement graph of \mathcal{G} . Finally, each vertex in \mathcal{V} represents a qubit and x_i, y_i and possibly z_i are the coordinates of the qubits into the 2D/3D domain, $\forall i \in \mathcal{V}$.

3.1 From QPU requirements to embedding constraints

The register on which qubits have to be placed identifies the domain for the qubits' positions. In the considered device (Pasqal's R&D prototype, Chadoq2[22]), this domain can cover at most a circular area with a maximum radius of $50 \mu m$, so qubits must be placed with a maximum pair distance d_{ij} of $100 \mu m$. Moreover, in this machine setting, two atoms cannot be closer than $4 \mu m$. Hence, the constraints:

$$d_{ij} \leq 100 \mu m \quad \forall i, j \in \mathcal{V}, i < j \quad (2)$$

$$d_{ij} \geq 4 \mu m \quad \forall i, j \in \mathcal{V}, i < j \quad (3)$$

Then, the unit disk representation requires vertexes, *i.e.*, qubits, that share an edge, to be closer than non-adjacent vertexes, *i.e.*, the ones not connected by an

edge. In principle, this constraint could be written as:

$$\max_{(i,j) \in \mathcal{E}} d_{ij} < \min_{(h,k) \in \bar{\mathcal{E}}} d_{hk} \quad (4)$$

But this formulation could lead to an embedding configuration that prevents the blockade effect to take place. In fact, the Rydberg blockade effect occurs in the regime of strong interactions, which determines the blockade radius r_b [23] as

$$r_b = \sqrt[6]{\frac{C_6}{\hbar\Omega}} \quad (5)$$

Moreover, to obtain the opposite entanglement [24] between pairs of qubits that fall into the r_b radius, the laser pulses should have a time duration that is

$$t = \frac{\pi}{\sqrt{2}\Omega} \quad (6)$$

This is the time duration at which the maximally entangled state is reached [8]. From equation (6), it is possible to compute the minimum Rabi frequency that can be allowed given \tilde{t} , the coherence time limit of the machine. Then, exploiting eq. (5) the maximum blockade radius \tilde{r}_b can be obtained as follows:

$$\tilde{r}_b = \sqrt[6]{\frac{\sqrt{2}|C_6|\tilde{t}}{\hbar\pi}} \quad (7)$$

This leads to a reformulation of constraint (4) that considers also the time limit \tilde{t} of the machine:

$$d_{ij} \leq \tilde{r}_b \quad \forall (i, j) \in \mathcal{E} \quad (8)$$

$$d_{hk} > \tilde{r}_b \quad \forall (h, k) \in \bar{\mathcal{E}} \quad (9)$$

The described constraints define a constrained UD graph problem, whose solution, both in the 2D and 3D cases, provides realistic coordinates for qubits embedding in the real quantum register. Providing qubits' positions, while avoiding fixed geometries, is enhanced by the capability to control individual atoms trapped in optical tweezers, through a real-time control system [25].

3.2 Traditional approaches as a starting point

Starting from the above-mentioned constraints, the first approach to the UD embedding problem could be to solve the corresponding quadratic constrained non-convex optimization problem that arises using the positions of the qubits as variables. For these kinds of problems, several solvers have been proposed (*e.g.*, [26, 27, 28, 29]), but the UD embedding problem in the general case is NP-hard, so these approaches did not provide feasible solutions as the number of qubits increases or as the connectivity pattern becomes more complicated.

Other approaches to achieving embedding representation take inspiration from force-directed algorithms, like the Fruchterman-Reingold method [30] [31]: this force-directed layout algorithm models attractive and repulsive forces between vertexes pairs, according to the adjacency pattern described by A ; repulsive forces intervene on all vertexes pairs with module k^2/d_{ij}^2 , whilst attractive forces intervene only on adjacent pairs and have module d_{ij}/k . The traditional algorithm, in this case, is designed to find the embedding in a square domain and takes as a parameter the optimal distance k at which the two forces balance for adjacent pairs [31], which in our case could be set to a value in the range $[4, \tilde{r}_b]$. There exists an easy way to modify the domain of the embedding to match our requirement expressed by eq. (2), that is by just changing the positions' projector into a circular domain with a radius of $50 \mu m$ instead of the current square domain projector, anyway, a further modification to match all other constraints (eqs. (3), (8), (9)) is not straightforward.

Finally, it is worth mentioning that some optimization problems inherit their connectivity patterns in the QUBO formulation from their own topology, such as antennas' positions in a PCI (Physical Cell Identifier) problem [32], [33]. In these cases, some initial positions are yet provided and they can be scaled to match the register limit imposed by eq. (2), anyway the matching of all other constraints is not guaranteed.

All these approaches provide some initial positions, $(x_i, y_i) \forall i \in \mathcal{V}$, for the UD embedding in the regis-

ter, which is yet an improvement over a randomly set embedding, nevertheless, further manipulation is required to match the optimization problem structure.

3.3 GEAN: Graph Embedding Autoencoder Network

3.3.1 Model architecture

The proposed solution implements a Neural Network based model, reported in figure 2, which has two main components:

i) The autoencoder part follows the typical symmetric architecture, with an input layer whose nodes are associated with the initial positions $(x_i, y_i) \forall i \in \mathcal{V}$ and an output layer, *CoordL*, that provides the transformed positions $(x'_i, y'_i) \forall i \in \mathcal{V}$. The *CoordL* is equipped with an activation function that is defined as $F_a(t) := 50 \tanh(t)$, thus the transformed coordinates belong to the square domain $(-50, 50)^2$ that inscribes the circular domain of interest. As the considered domain is two-dimensional, the input and the output layers have $2n$ nodes each, the first n nodes of each layer represent x-coordinates, and the other n nodes are y-coordinates. The hidden layers of the model respectively have 64, 36, 18, 9, 18, 36 and 64 hidden nodes, they use ReLU as activation functions, and they are interspersed with dropout layers, with dropout probability set to 0.5, for regularization purposes. All the weights of the autoencoder are trainable and the bias nodes' contributions are also taken into account. The choice of this particular architecture was driven by two main observations. On one hand, as the order of the elements in the input vector does not imply local correlations, Convolutional Neural Networks (CNNs) have been discarded, as fully-connected architecture allowed a more general interpretation of the elements in the input vector, we opted for that choice; on the other hand, autoencoder architecture was proposed in [34], in this work, an autoencoder is used to retrieve coordinates given the distance matrix in a protein modelling use case, even though the same architecture was not straightly usable for our use case, we do not have feasible pair distances a priori, Eguchi et al. study suggested us that our neural network architecture could benefit

from a latent embedding representation.

ii) The second component makes the model aware that the output of the autoencoder, $(x'_i, y'_i) \forall i \in \mathcal{V}$, are Cartesian coordinates. This component is in charge of computing Euclidean distances between all the qubits' pairs, which is obtained by adding a sparse layer that connects the *CoordL* to nodes in the *DiffL* layer, which consists of $2\binom{n}{2} = n(n-1)$ nodes. The *DiffL* layer has no contribution from a bias node and non-trainable weights in the sparse layer are set to ± 1 , such that the nodes α in the *DiffL* reflect differences between pairs of coordinates:

$$\alpha_{(i-1)(n-1)-\binom{i-1}{2}+j-i} = x'_i - x'_j \quad i, j \in \mathcal{V}, i < j \quad (10)$$

$$\alpha_{(n-1)\binom{n}{2}+i-1-\binom{i-1}{2}+j-i} = y'_i - y'_j \quad i, j \in \mathcal{V}, i < j \quad (11)$$

Then a square activation function $F_a(t) := t^2$ is applied to *DiffL*. Finally, the squared pairs' distances are retrieved by adding another sparse layer, with fixed weights of value +1 that allows the $\binom{n}{2}$ nodes, β , in the output layer, *DistL*, to assume the following values:

$$\beta_k = \alpha_k^2 + \alpha_{\binom{n}{2}+k}^2 \quad k \in \left\{1, 2, \dots, \binom{n}{2}\right\} \quad (12)$$

At last, the $F_a(t) := \sqrt{t}$ activation function intervenes on the *DistL* layer to make the outputs of the GEAN model, d , become the pairs' distances d_{ij} , $i < j \in \mathcal{V}$.

3.3.2 Loss function

Once the pairs' distances are computed, it is possible to model the loss function to target a feasible UD embedding. In this case, the training of the model has a different meaning from the one of a typical machine learning task: the mini-batch of the training set consists of just one sample, *i.e.*, the initial positions $(x_i, y_i) \forall i \in \mathcal{V}$, and the minimization of the loss function does not correspond to the intent of prediction, but it pursues a proper approximation of the domain transformation to achieve the desired UD connectivity.

As there are four kinds of constraints that the UD embedding must satisfy (eqs. (2), (3), (8), (9)), the proposed loss function is four-folded and each part of it is applied to the distances output vector d : in accordance to eq. (2), the qubits' pairs that are more distant than $100 \mu m$ should be penalized, so the first contribution to the loss function, $loss_1$, is computed as follows:

$$loss_1(d) = \text{avg}_{i < j \in \mathcal{V}} |\max(100, d_{ij}) - 100| \quad (13)$$

To match constraint (3), which defines the minimum distance allowed between qubits' pairs, $loss_2$ contribution is designed as:

$$loss_2(d) = \text{avg}_{i < j \in \mathcal{V}} |\min(4, d_{ij}) - 4| \quad (14)$$

Then, $loss_3$ expresses constraint (8), and it penalizes pairs' distances of adjacent qubits that are greater than \tilde{r}_b :

$$loss_3(d) = \text{avg}_{i < j \in \mathcal{V}} A_{ij} |\max(\tilde{r}_b, d_{ij}) - \tilde{r}_b| \quad (15)$$

Finally, $loss_4$ is the counterpart for constraint (9), and it penalizes pairs' distances of non-adjacent qubits that are shorter than $\tilde{r}_b + \epsilon$, where parameter ϵ is set to 0.1 to account for the strictly greater sign in the constraint:

$$loss_4(d) = \text{avg}_{i < j \in \mathcal{V}} (1 - A_{ij}) |\min(\tilde{r}_b + \epsilon, d_{ij}) - (\tilde{r}_b + \epsilon)| \quad (16)$$

The overall loss function sums up all the previous contributions. The absence of weighting constants in the final loss formulation is justified by the fact that all constraints violations have the same importance; they all would imply an unfeasible embedding:

$$loss(d) = loss_1(d) + loss_2(d) + loss_3(d) + loss_4(d) \quad (17)$$

In order to minimize the *loss* function, the *PyTorch* implementation of the AdamW algorithm [35] has been exploited, in combination with a constant learning rate equal to $1e^{-3}$. The maximum number of epochs allowed for each graph embedding has been set to 5000, but a stopping criterion based on the achievement of a feasible embedding prevents the training to go through all the epochs when it is not necessary.

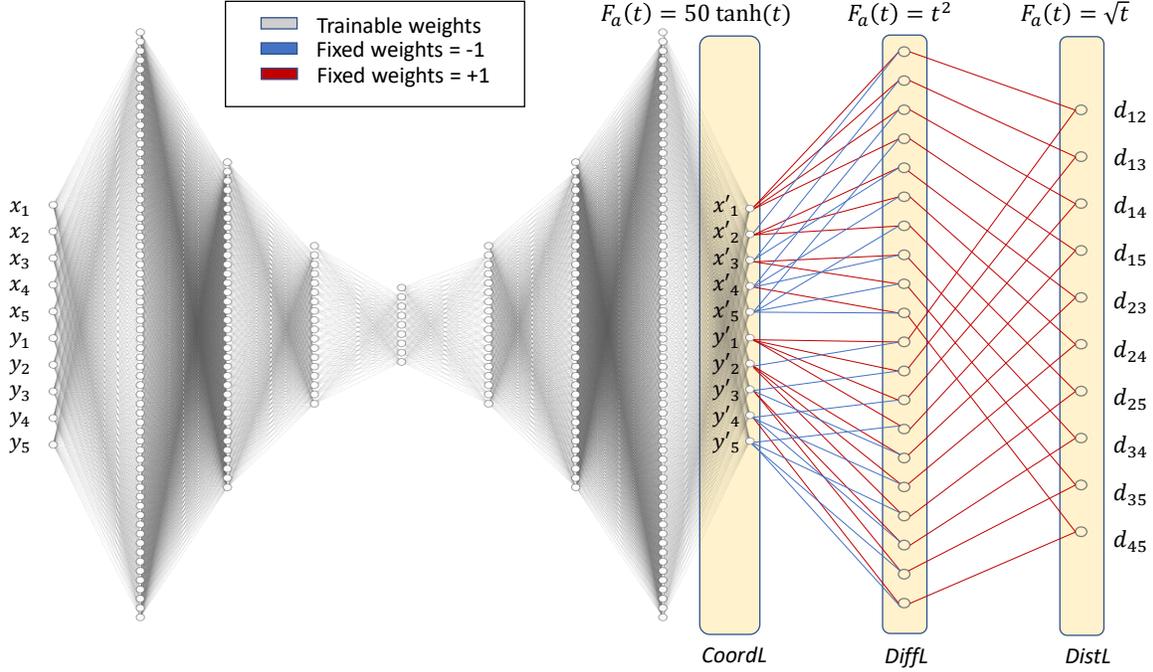


Figure 2: GEAN architecture: 5 qubits embedding in 2D, the initial coordinates $(x_1, y_1), \dots, (x_5, y_5)$ are transformed into feasible coordinates $(x'_1, y'_1), \dots, (x'_5, y'_5)$. The output layer of the model, $DistL$, provides all pairs' distances between nodes.

3.4 The limitations of the 2D domain

This UD embedding methodology has to be subject to the limitations of 2-dimensional space, thus the connectivity of the graphs that can be embedded is not general. As they are of particular interest to identify which QUBO problems could be embedded into the 2D register, two necessary conditions have been identified: one concerning the maximal clique and the other one regarding the maximal degree of graph \mathcal{G} . These two connectivity-related properties should be complemented by the other requirements expressed in the 1 section: constant values in the diagonal of Q and constant sign for its off-diagonal elements. The key concept underlying these results comes from *Thue's Theorem*, which states that *the regular hexagonal packing is the densest circle packing in the plane* [36]. Thus, the densest packing for

the considered qubits' register has the structure reported in figure 3b). Considering constraint (3), the hexagon side of the densest embedding is $4 \mu m$. Furthermore, the time limit of the machine determines the maximum \tilde{t} , and consequently \tilde{r}_b , for this experimental setting $\tilde{t} = 3 \mu s$ and $\tilde{r}_b \approx 10.26 \mu m$. From these considerations, the following properties are obtained:

Property 1 (Maximum clique property) *The maximum N , such that a complete graph (or clique), K_N , with N vertices, can be embedded into the register is $N = 7$.*

Property 2 (Maximum degree property) *Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the maximal degree $d_{max}(\mathcal{G})$ allowed for a feasible embedding is 18.*

Fig. 3 represents both these properties. To embed a clique, all the vertices of the graph, K_N , must lie

within a circle of maximum radius set to $\frac{\tilde{r}_b}{2}$ (fig. 3a)) and the maximum degree for a graph, $d_{max}(\mathcal{G})$, implies all the adjacent qubits to lie within a maximum radius \tilde{r}_b . Anyway this bound on the maximum degree relies on the hypothesis that the adjacency pattern among all other qubits within the circle matches the adjacency matrix A one.

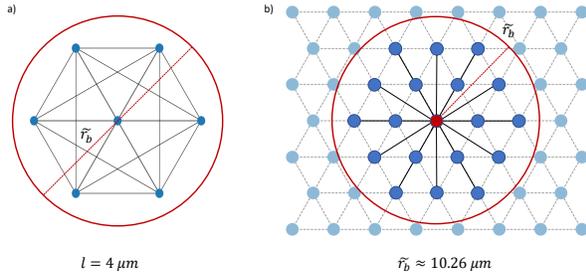


Figure 3: Representation of the maximal clique, K_7 , fig. 3a), and of the maximal degree, fig. 3b), that can be embedded on the register. The baseline embedding is the regular hexagonal packing with $l = 4 \mu m$ side and the Rydberg blockade radius is the maximum available on the machine $\tilde{r}_b \approx 10.26 \mu m$.

3.5 Adding new dimensions: the 3D embedding case

To be able to embed a greater set of QUBO problems and with a view to a future quantum machine able to work with a 3D register, the GEAN model can be modified to obtain also feasible z-coordinates. This implies some modification to the architecture presented in Fig. 2 but does not require loss function changes, given that the constraints on pairs' distances remain the same. The architecture changes concern:

i) the addition of nodes in the input and in the *CoordL* layers to represent the z-coordinates, z_i and $z'_i, \forall i \in \mathcal{V}$, so the size of these layers passes from $2n$ to $3n$ nodes. When the initial z_i coordinates are not available, they can be initialized to 0;

ii) the *DiffL* layer should be modified to account also for the z-coordinates pairs' differences, so it con-

tains $3\binom{n}{2}$ nodes, and the weights in the fully connected layer are modified such that the last $\binom{n}{2}$ α nodes in *DiffL* are

$$\alpha_{(n-1)(n+i-1)-\binom{i-1}{2}+j-i} = z'_i - z'_j \quad i, j \in \mathcal{V}, \quad i < j \quad (18)$$

iii) the *DistL* layer does not change the size, as it still computes pairs' distances between qubits, but the increment of the *DiffL* layer requires it to consider the contribution also from the z-coordinates square differences, so β nodes input values become

$$\beta_k = \alpha_k^2 + \alpha_{\binom{n}{2}+k}^2 + \alpha_{2\binom{n}{2}+k}^2 \quad k \in \left\{1, 2, \dots, \binom{n}{2}\right\} \quad (19)$$

4 Results and Discussion

4.1 Classical solver as a benchmark

In order to provide benchmarks, we formulated the Unit Disk graph problem and attempted to find solutions for feasible embeddings with state-of-the-art solvers. In particular, we opted for the *Gurobi* solver [37] and handled the implementation of the programming model through the *Pyomo*¹ Python library. The choice of a classical solver for comparison, instead of a heuristic approach based on fixed geometries, is intended to try and fully exploit at most all the degrees of freedom available for the placements of the atoms in the register.

Since the considered optimization problem is NP-hard, the computing time for retrieving the optimal solution is linked to the dimension of the problem, *i.e.*, the n number of qubits. Accordingly, as n increases it becomes necessary to set up a time limit for the solver, thus we set a maximum walltime for the classical solver to 2 minutes, as it is the maximum time required by the GEAN model to retrieve a feasible solution on the largest instance.

In the definition of the programming model, which is quadratic and non-convex, we will refer to \mathcal{P} as the set of all unordered pairs in \mathcal{V} , thus $|\mathcal{P}| = \frac{n(n-1)}{2}$, the embedding dimensionality is $N \in \{2, 3\}$, and

¹<https://pyomo.readthedocs.io>

the positions of the qubits will be represented by N -dimensional vector $\vec{p}_i, \forall i \in \mathcal{V}$. The retrieval of a feasible embedding is modelled through γ_{ij} binary variables, described as follows.

$$\gamma_{ij} := \begin{cases} 1 & \text{pair distance is unfeasible} \\ 0 & \text{pair distance is feasible} \end{cases} \quad (20)$$

Hereafter, we present the overall constrained Unit Disk graph problem, with reference to the specifications of the considered Quantum machine.

$$\begin{aligned} \min_{\vec{p}, \gamma} \quad & \sum_{\{i,j\} \in \mathcal{P}} \gamma_{ij} \\ \text{s.t.} \quad & d_{ij}^2 \leq \tilde{r}_b^2 + ((100\sqrt{2})^2 - \tilde{r}_b^2)\gamma_{ij} & (i,j) \in \mathcal{E}, \\ & d_{ij}^2 \geq (1 - \gamma_{ij})4^2 & (i,j) \in \mathcal{E}, \\ & d_{ij}^2 \leq 100^2 + 100^2\gamma_{ij} & (i,j) \notin \mathcal{E}, \\ & d_{ij}^2 \geq (1 - \gamma_{ij})(\tilde{r}_b + \epsilon)^2 & (i,j) \notin \mathcal{E}, \\ & d_{ij}^2 = \|\vec{p}_i - \vec{p}_j\|_2^2 & \{i,j\} \in \mathcal{P}, \\ & \vec{p} \in [-50, +50]^{n \times N}, \\ & \gamma \in \{0, 1\}^{|\mathcal{P}|} \end{aligned} \quad (21)$$

4.2 QUBO instances description

To evaluate the proposed methodology, different QUBO problem instances have been investigated. The studied problems all allow for a binary variables representation that can be mapped to the machine Hamiltonian (1), thus, even if in the classical nomenclature of optimization problems, they are referred to with different names (*i.e.*, graph coloring, protein folding problems), all problems that can be formulated as Maximum Independent Set problems [38] can take advantage of the proposed embedding methodology.

Starting from previous work on graph coloring problems [39], solved through an iterative QUBO Maximum Independent Set (MIS) approach [6], the *antennas dataset*, has been created: this dataset is natively UD-based, as it is built from the original

positions of antennas² in the city of Turin and it presents an adjacency pattern set using conflict distance D_c : all antennas that are nearer than D_c are in conflict and as a consequence introduce a quadratic penalty term in the MIS QUBO formulation. The *antennas dataset* contains at most 87 antennas, and as each antenna is represented by a qubit in the MIS QUBO formulation, this implies that at most 87 qubits should be embedded. Anyway, according to the distance D_c , the *antennas dataset* could be separated into connected components and the respective QUBO instances can be embedded separately. The *antennas dataset* provides an example for QUBO problems that rely on a precise topology to obtain starting positions $(x_i, y_i), \forall i \in \mathcal{V}$ scaled to the quantum register domain. Problems' instances of this kind will be hereafter named *QUBO MIS antennas* problems.

Then, other not UD-based problems have been investigated. For instance, the protein folding problem aims to find the folding of a protein chain made of hydrophilic and hydrophobic amino acids. The chain naturally folds to bring as many hydrophobic acids as possible close together, this is modelled by maximizing the number of hydrophobic acid matchings, which in the optimization problem are modelled as binary variables δ_{ij} . So, starting from the *protein folding* formulation in [40], the equivalent QUBO formulation is proposed:

$$\begin{aligned} \min_{\delta_{ij}} \quad & - \sum_{i>j} \delta_{ij} + P \sum_{ij,kh \in S} \delta_{ij}\delta_{kh} \\ \text{s.t.} \quad & \delta_{ij} \in \{0, 1\} \end{aligned} \quad (22)$$

$$S = \{ij, kh \mid i \leq f < j, f \neq \frac{i+j-1}{2}, f = \frac{k+h-1}{2} \in \mathbb{N}\} \quad (23)$$

The QUBO instances belonging to this class of problems will be referred to as *QUBO protein folding* samples and the possible matchings δ_{ij} will be named after the hydrophobic acids positions.

Finally, a QUBO formulation that fits the Hamiltonian formulation of eq. (1), can be provided also for a one-hot-encoding modelled graph coloring problem

²<https://opencellid.org/>

[41]. Problems' instances belonging to this class will be referred to as *QUBO graph coloring* problems.

For both the *QUBO protein folding* and the *QUBO graph coloring* problems, the initial positions of the qubits (x_i, y_i) , $\forall i \in \mathcal{V}$ are computed through the Fruchterman-Reingold method.

To describe the characteristics of the embeddings, the following notation is introduced

- $|K_{max}|$, the size of the estimated maximum clique
- r_{min}^0 , minimum distance among qubits in the initial configuration
- r_{min}^f , minimum distance among qubits in the final configuration
- r_{max}^0 , maximum distance among qubits in the initial configuration
- r_{max}^f , maximum distance among qubits in the final configuration
- r_{adj}^0 , maximum distance among adjacent qubits in the initial configuration
- r_{adj}^f , maximum distance among adjacent qubits in the final configuration
- r_{adj}^0 , minimum distance among non-adjacent qubits in the initial configuration
- r_{adj}^f , minimum distance among non-adjacent qubits in the final configuration
- E_c , number of epochs used by the GEAN model to find a feasible embedding
- *Gurobi_sol*, whether the *Ipopt* solver is able to retrieve a feasible solution

4.3 QUBO instances embeddings

The first set of results concerns *QUBO MIS antennas* problems, embedded in the 2D register. Setting a conflict distance $D_c = 130 m$, all the connected components of the corresponding MIS QUBO problems were successfully embedded onto the quantum

register. In particular, 10 out of the 87 initial vertexes were isolated nodes, the other vertexes could be divided into 17 connected components. Among the connected components, there were 6 K_2 , 1 K_3 and 1 K_5 graphs, whose feasible embedding can be trivially obtained and 3 graphs with 3 vertexes that had yet a feasible embedding from the scaling operation. All other connected components required the intervention of the GEAN model to reach feasibility as summarized in Table 1. From the results, it can be noticed that the unfeasible condition of the initial position is related to too near qubits that violate constraint (3). However, applying a uniform magnification to work around the issue with the r_{min}^0 would just shift the issue: for instance, considering graph \mathcal{G}_c , r_{min}^0 is 3 order of magnitude below the minimum allowed distance of $4 \mu m$, a uniform magnification would make the constraint (3) satisfied, but would violate constraints (2) and (8). Fig. 4 shows the embedding transformations on the hardest to embed, in terms of E_c , connected components of the *QUBO MIS antennas* problem.

Concerning the instances of *QUBO protein folding* problems, feasible embeddings were reached for all the proposed instances. The graphs associated with the QUBO problems will be referred to as \mathcal{G}_{aa_ha} , with *aa* as the number of the amino acids of the entire chain and *ha* the number of the hydrophobic ones. In particular, the results concerned the following instances:

- $\mathcal{G}_{12.6}$ with hydrophobic amino acids placed at positions 1, 2, 3, 5, 11, 12
- $\mathcal{G}_{17.7}$ with hydrophobic amino acids placed at positions 1, 2, 5, 6, 10, 12, 17
- $\mathcal{G}_{22.8}$ with hydrophobic amino acids placed at positions 1, 3, 5, 6, 9, 10, 11, 17

In these cases, the initialization of the coordinates $(x_i, y_i) \forall i \in \mathcal{V}$ is performed through the Fruchterman-Reingold method with the equilibrium point between the attractive and the repulsive forces set to $k = 4 \mu m$. As can be noticed from Table 2, the initial positions lack the unit disk property, as $r_{adj}^0 > r_{adj}^0$. Moreover, the absence of the control on

Table 1: Results obtained with the GEAN model to embed the *QUBO MIS antennas* instances with a conflict distance D_c set to 130 m.

\mathcal{G}	$ \mathcal{V} $	d_{max}	$ K_{max} $	r_{min}^0 (μm)	r_{min}^f (μm)	r_{max}^0 (μm)	r_{max}^f (μm)	r_{adj}^0 (μm)	r_{adj}^f (μm)	r_{adj}^0 (μm)	r_{adj}^f (μm)	E_c	<i>Gurobi_sol</i>
\mathcal{G}_a	13	8	5	0.454	4.006	19.457	26.725	7.070	9.984	7.193	10.634	733	Yes
\mathcal{G}_b	7	4	4	2.634	4.004	15.956	21.265	6.873	10.090	7.265	11.155	71	Yes
\mathcal{G}_c	6	4	4	0.001	4.013	14.663	22.046	6.899	8.943	8.731	11.462	384	Yes
\mathcal{G}_d	6	5	3	1.960	4.028	11.855	16.650	7.067	8.724	7.334	11.273	374	Yes
\mathcal{G}_e	4	3	3	1.198	4.037	7.799	8.547	6.952	6.511	7.799	8.547	98	Yes
\mathcal{G}_f	5	4	4	0.855	4.061	7.255	13.473	6.670	9.865	7.255	13.473	261	Yes
\mathcal{G}_g	7	5	4	2.559	4.015	13.137	21.496	6.572	8.616	7.418	11.139	281	Yes

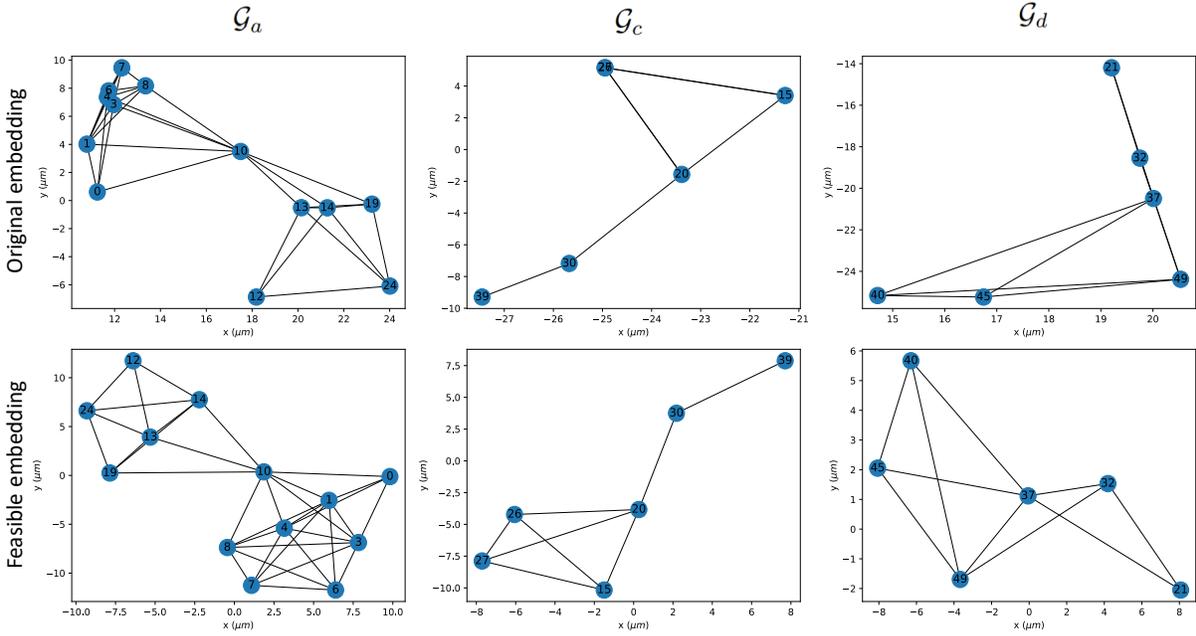


Figure 4: Transformation into feasible 2D embeddings of the \mathcal{G}_a , \mathcal{G}_c and \mathcal{G}_d connected components for the *QUBO MIS antennas* problem.

the maximum distance between adjacent vertexes leads to $r_{adj}^0 > \tilde{r}_b$. The difficulty of the UD embed-

ding task is reflected in the loss function behaviour, see Fig. 5, most of the loss contribution comes

from the adjacency pattern requirements. In fact, averaging on the overall *loss* score, 37% and 31% contributions come respectively from *loss*₃ and *loss*₄.

Sometimes the limitations on the 2D embeddings do not allow for a feasible UD embedding. Hence, here are reported the instances whose adjacency pattern could not be fully respected in a 2D configuration, but that achieved feasibility through the exploitation of the third dimension. An example comes from the *QUBO MIS antennas* case with an increased conflict distance $D_c = 250\text{ m}$, which makes the graph of the corresponding QUBO model form a unique connected component, so the UD embedding considers 87 qubits to place into a sphere domain of radius $50\text{ }\mu\text{m}$, see Fig. 6.

The increased dimensionality allows the embedding to deal with a more complex graph that has $d_{max} = 14$ and $|K_{max}| = 10$. The effect of the GEAN model on the coordinates transformation has a significant impact on the feasibility of the embedding: the minimum distance among qubits is increased by 3 orders of magnitude, from $r_{min}^0 = 0.001\text{ }\mu\text{m}$ to $r_{min}^f = 4.007\text{ }\mu\text{m}$, and the distance among adjacent qubits is reduced, from $r_{adj}^0 = 13.552\text{ }\mu\text{m}$ to $r_{adj}^f = 10.134\text{ }\mu\text{m}$, within 2120 epochs. Moreover, the UD property is still preserved, as r_{adj}^f is $10.415\text{ }\mu\text{m}$ and the maximum distance constraint is largely satisfied as $r_{max}^f = 53.300\text{ }\mu\text{m}$. The *Gurobi* solver could not provide a feasible solution within a comparable time.

Another example is from the *QUBO graph coloring* problem: from the red graph in Fig. 7, a three colors graph coloring QUBO problem was formulated. In this case, the graph \mathcal{G}_{GC} that is associated with the QUBO formulation has 21 vertexes, $d_{max} = 6$, $|K_{max}| = 3$, corresponding to the clique that is associated with the exclusivity constraint of colors assigned to a vertex. The effect of the GEAN model

transformation can be summed up as follows:

$$\begin{aligned} r_{min}^0 = 8.948\text{ }\mu\text{m} &\longrightarrow r_{min}^f = 4.183\text{ }\mu\text{m} \\ r_{max}^0 = 67.980\text{ }\mu\text{m} &\longrightarrow r_{max}^f = 28.545\text{ }\mu\text{m} \\ r_{adj}^0 = 19.217\text{ }\mu\text{m} &\longrightarrow r_{adj}^f = 10.242\text{ }\mu\text{m} \\ r_{adj}^0 = 9.137\text{ }\mu\text{m} &\longrightarrow r_{adj}^f = 10.265\text{ }\mu\text{m} \end{aligned}$$

The feasible UD embedding was reached within 767 epochs, whilst the *Gurobi* solver did not reach a feasible embedding within the available wall-time.

A final observation about the power of this embedding approach concerns its running speed: since the input of each training step consists of just one sample, even many epochs can be performed in a short time. Implementing and running the model on a standard consumer machine not equipped with dedicated acceleration hardware, it was possible to obtain solutions for both the 2D and the 3D embeddings instances within 2 minutes also for the biggest and most complex instance, *i.e.*, embedding of *QUBO MIS antennas* in 3D. This capability to retrieve solution in such short times represent an advantage over traditional approaches for the solution of UD graph problems, as a matter of fact, the *Gurobi* solver was not able to provide a feasible embedding for instances considering 10, 21 and 87 qubits.

5 Conclusion

Summing up, the GEAN model shortens the gap between the theoretical QUBO problems formulation and the actual quantum optimization, by allowing UD embedding to match with the machine’s effective Hamiltonian. The proposed model relies on a strong characterization of a specific quantum register, but it can be easily adapted to match other requirements, at different levels. Nevertheless, our solution is not constrained to a fixed lattice structure and requires the minimum number of physical qubits to provide feasible embeddings. Thus, it provides an efficient tool to test QUBO problems of non-trivial scale on quantum hardware that cannot handle many qubits. The success of the GEAN model over other classical approaches is enhanced by the tight control

Table 2: Results obtained with the GEAN model to embed instances of the *QUBO protein folding* problem.

\mathcal{G}	$ \mathcal{V} $	d_{max}	$ K_{max} $	r_{min}^0 (μm)	r_{min}^f (μm)	r_{max}^0 (μm)	r_{max}^f (μm)	r_{adj}^0 (μm)	r_{adj}^f (μm)	r_{adj}^0 (μm)	r_{adj}^f (μm)	E_c	<i>Gurobi_sol</i>
$\mathcal{G}_{12.6}$	5	4	3	27.344	4.000	74.611	11.324	55.524	9.853	27.344	11.138	187	Yes
$\mathcal{G}_{17.7}$	10	9	4	11.601	4.056	64.463	20.237	32.719	10.255	22.277	10.606	492	No
$\mathcal{G}_{22.8}$	9	7	4	9.438	4.104	79.640	23.329	27.957	10.217	27.215	10.514	404	Yes

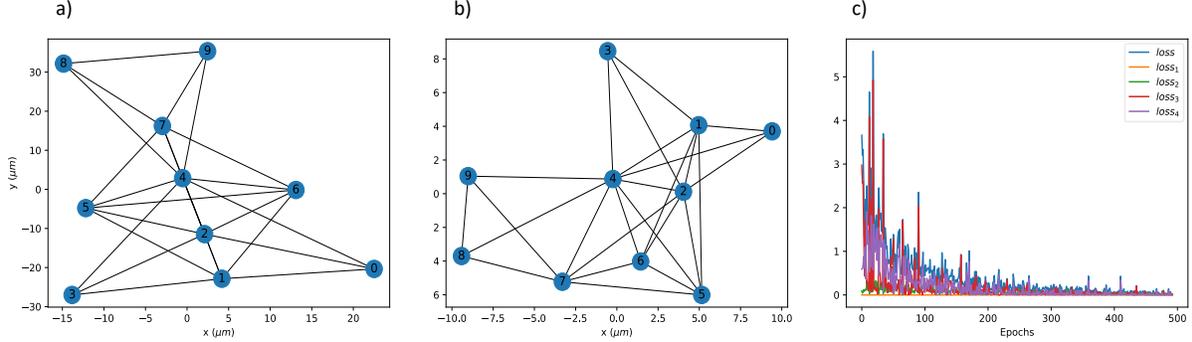


Figure 5: Embedding for the *QUBO protein folding* problem corresponding to $\mathcal{G}_{17.7}$: the initial position, fig. 5a), are computed through Fruchterman-Reingold method and are mapped into a feasible 2D UD embedding reported in fig. 5b). Fig. 5c) shows the behaviour of the *loss* function along the epochs.

of the distances, relying on loss function definition, and the exploitation of very effective optimizers for neural networks' training. Another key feature of the proposed solution is the multi-dimensional readiness, the extension from the 2D to the 3D case has yet been presented here, but a general N -dimensions definition, targeting distances' computation as outputs, is straightforward. As a matter of fact, this methodology can pave the way to other problems' solutions, not only embedding related ones.

Future directions for this work will consider a further characterization of the QUBO problems that admits a feasible UD embedding, both for the 2D and the 3D cases. A more detailed comparison with state-of-the-art solvers or heuristic approaches will be also

performed, thus gathering more insights on the scalability and performance of the GEAN model. Moreover, other stopping criteria and convergence analysis along the epochs will be explored to increase the gap between r_{adj}^f and r_{adj}^0 , with the aim of differentiating better adjacent from non-adjacent qubits in the embedding configuration. Finally, a test phase on the real machine for some classes of problems, mainly *QUBO MIS antennas*' instances, will bring further validation to the embedding methodology. The validation on the real hardware will also support a study to quantify the error tolerance for the qubits' placements, concerning the expected Rydberg blockade effect.

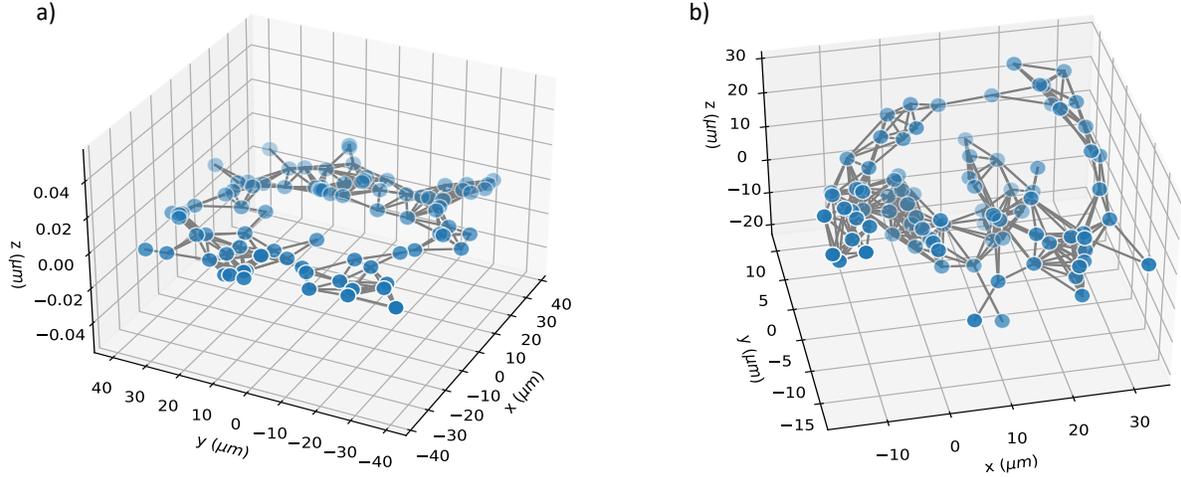


Figure 6: Embedding for the full *QUBO MIS antennas* problem in 3D: the initial positions with z -coordinates set to 0, fig. 6a), are mapped into a feasible 3D embedding, fig. 6b).

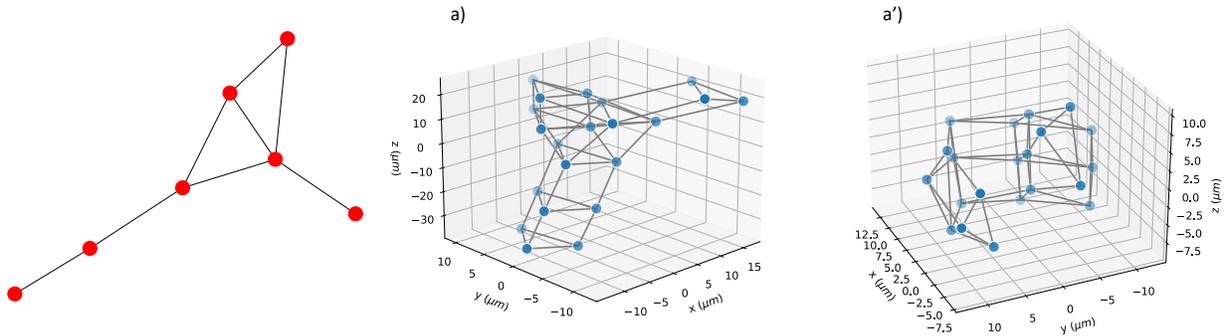


Figure 7: *QUBO graph coloring* embedding in 3D corresponding to \mathcal{G}_{GC} : starting from the red graph on the left, the one-hot-encoding QUBO formulation is built to take into account 3 colors. The initial qubits positions shown on fig. 7a) are mapped into a feasible configuration as in fig. 7a').

Acknowledgment

This work has been supported by the CINECA consortium and by Pasqal in the context of the Iskra-C project QOPS, resulting from a joint collaboration.

References

- [1] Loïc Henriët et al. “Quantum Computing with Neutral Atoms”. In: *Quantum* 4 (Sept. 2020), p. 327. ISSN: 2521-327X. DOI: 10.22331/q-2020-09-21-327.

- [2] Thomas F Gallagher. “Rydberg atoms”. In: *Reports on Progress in Physics* 51.2 (1988), p. 143.
- [3] Andrew Lucas. “Ising Formulations of Many NP Problems”. In: *Frontiers in Physics* 2 (2014). ISSN: 2296-424X. DOI: 10.3389/fphy.2014.00005.
- [4] Fred Glover, Gary Kochenberger, and Yu Du. “A tutorial on formulating and using QUBO models”. In: *arXiv preprint arXiv:1811.11538* (2018).
- [5] Michel Fabrice Serret, Bertrand Marchand, and Thomas Ayrat. “Solving Optimization Problems with Rydberg Analog Quantum Computers: Realistic Requirements for Quantum Advantage Using Noisy Simulation and Classical Benchmarks”. In: *Physical Review A* 102.5 (Nov. 2020), p. 052617. ISSN: 2469-9926, 2469-9934. DOI: 10.1103/PhysRevA.102.052617.
- [6] Constantin Dalyac et al. “Qualifying quantum approaches for hard industrial optimization problems. A case study in the field of smart-charging of electric vehicles”. In: *EPJ Quantum Technology* 8.1 (May 2021). ISSN: 2196-0763. DOI: 10.1140/epjqt/s40507-021-00100-3. URL: <http://dx.doi.org/10.1140/epjqt/s40507-021-00100-3>.
- [7] E Urban et al. “Observation of Rydberg blockade between two atoms”. In: *Nature Physics* 5.2 (2009), pp. 110–114.
- [8] T. Wilk et al. “Entanglement of Two Individual Neutral Atoms Using Rydberg Blockade”. In: *Phys. Rev. Lett.* 104 (1 Jan. 2010), p. 010502. DOI: 10.1103/PhysRevLett.104.010502. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.104.010502>.
- [9] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. “Unit disk graphs”. In: *Discrete Mathematics* 86.1 (1990), pp. 165–177. ISSN: 0012-365X. DOI: [https://doi.org/10.1016/0012-365X\(90\)90358-0](https://doi.org/10.1016/0012-365X(90)90358-0). URL: <https://www.sciencedirect.com/science/article/pii/S0012365X90903580>.
- [10] Heinz Breu and David G. Kirkpatrick. “Unit Disk Graph Recognition Is NP-hard”. In: *Computational Geometry* 9.1-2 (Jan. 1998), pp. 3–24. ISSN: 09257721. DOI: 10.1016/S0925-7721(97)00014-X.
- [11] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. “Unit Disk Graph Approximation”. In: *Proceedings of the 2004 Joint Workshop on Foundations of Mobile Computing - DIALM-POMC '04*. Philadelphia, PA, USA: ACM Press, 2004, p. 17. ISBN: 978-1-58113-921-1. DOI: 10.1145/1022630.1022634.
- [12] Fabian Kuhn, Rogert Wattenhofer, and Aaron Zollinger. “Ad-hoc networks beyond unit disk graphs”. In: *Proceedings of the 2003 joint workshop on Foundations of mobile computing*. 2003, pp. 69–78.
- [13] Sujoy Bhore et al. “Unit Disk Representations of Embedded Trees, Outerplanar and Multi-Legged Graphs”. In: *arXiv:2103.08416 [cs]* (Aug. 2021). arXiv: 2103.08416 [cs].
- [14] Maciej M Sysło. “Characterizations of outerplanar graphs”. In: *Discrete Mathematics* 26.1 (1979), pp. 47–53.
- [15] Douglas Brent West et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, 2001.
- [16] Wolfgang Lechner, Philipp Hauke, and Peter Zoller. “A quantum annealing architecture with all-to-all connectivity from local interactions”. In: *Science advances* 1.9 (2015), e1500838.
- [17] Xingze Qiu, Peter Zoller, and Xiaopeng Li. “Programmable quantum annealing architectures with Ising quantum wires”. In: *PRX Quantum* 1.2 (2020), p. 020311.
- [18] Minhyuk Kim et al. “Rydberg Quantum Wires for Maximum Independent Set Problems with Nonplanar and High-Degree Graphs”. In: *arXiv:2109.03517 [physics, physics:quant-ph]* (Sept. 2021). arXiv: 2109.03517 [physics, physics:quant-ph].

- [19] Neil Robertson and Paul D Seymour. “Graph minors. XIII. The disjoint paths problem”. In: *Journal of combinatorial theory, Series B* 63.1 (1995), pp. 65–110.
- [20] Jun Cai, William G. Macready, and Aidan Roy. “A Practical Heuristic for Finding Graph Minors”. In: *arXiv:1406.2741 [quant-ph]* (June 2014). arXiv: 1406.2741 [quant-ph].
- [21] Vicky Choi. “Minor-embedding in adiabatic quantum computation: I. The parameter setting problem”. In: *Quantum Information Processing* 7.5 (2008), pp. 193–209.
- [22] Henrique Silvério et al. “Pulser: An Open-Source Package for the Design of Pulse Sequences in Programmable Neutral-Atom Arrays”. In: *arXiv:2104.15044 [quant-ph]* (Jan. 2022). arXiv: 2104.15044 [quant-ph].
- [23] Donatella Ciampini, Oliver Morsch, and Ennio Arimondo. “Ultracold Rubidium Atoms Excited to Rydberg Levels”. In: *Journal of Atomic, Molecular, Condensed Matter and Nano Physics* 2.3 (2015), pp. 161–167.
- [24] CJ Picken et al. “Entanglement of neutral-atom qubits with long ground-Rydberg coherence times”. In: *Quantum Science and Technology* 4.1 (2018), p. 015011.
- [25] Daniel Barredo et al. “An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays”. In: *Science* 354.6315 (2016), pp. 1021–1023.
- [26] Jaehyun Park and Stephen Boyd. “General heuristics for nonconvex quadratically constrained quadratic programming”. In: *arXiv preprint arXiv:1703.07870* (2017).
- [27] Sourour Elloumi and Amélie Lambert. “Global solution of non-convex quadratically constrained quadratic programs”. In: *Optimization Methods and Software* 34.1 (2019), pp. 98–114. DOI: 10.1080/10556788.2017.1350675. eprint: <https://doi.org/10.1080/10556788.2017.1350675>. URL: <https://doi.org/10.1080/10556788.2017.1350675>.
- [28] William E Hart, Jean-Paul Watson, and David L Woodruff. “Pyomo: modeling and solving mathematical programs in Python”. In: *Mathematical Programming Computation* 3.3 (2011), pp. 219–260.
- [29] Michael L. Bynum et al. *Pyomo-optimization modeling in python*. Third. Vol. 67. Springer Science & Business Media, 2021.
- [30] Thomas MJ Fruchterman and Edward M Reingold. “Graph drawing by force-directed placement”. In: *Software: Practice and experience* 21.11 (1991), pp. 1129–1164.
- [31] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
- [32] Hakan Kavlak and Hakki Ilk. “PCI Planning Strategies for Long Term Evolution Networks”. In: *NETWORKING 2012 Workshops*. Ed. by Zdenek Becvar, Robert Bestak, and Lukas Kencl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 151–156. ISBN: 978-3-642-30039-4.
- [33] Jihong Gui, Zhipeng Jiang, and Suixiang Gao. “PCI Planning Based on Binary Quadratic Programming in LTE/LTE-A Networks”. In: *IEEE Access* 7 (2019), pp. 203–214. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2885313.
- [34] Raphael R Eguchi, Christian A Choe, and Po-Ssu Huang. “Ig-VAE: Generative Modeling of Protein Structure by Direct 3D Coordinate Generation”. In: *Biorxiv* (2022), pp. 2020–08.
- [35] Ilya Loshchilov and Frank Hutter. “Decoupled weight decay regularization”. In: *arXiv preprint arXiv:1711.05101* (2017).
- [36] Hai-Chau Chang and Lih-Chung Wang. “A simple proof of thue’s theorem on circle packing”. In: *arXiv preprint arXiv:1009.4322* (2010).
- [37] Bob Bixby. “The gurobi optimizer”. In: *Transp. Research Part B* 41.2 (2007), pp. 159–178.

- [38] Robert Endre Tarjan and Anthony E Trojanowski. “Finding a maximum independent set”. In: *SIAM Journal on Computing* 6.3 (1977), pp. 537–546.
- [39] Giacomo Vitali et al. *Towards Optimal Graph Coloring Using Rydberg Atoms*. 2021.
- [40] H Paul Williams. *Model building in mathematical programming*. John Wiley & Sons, 2013.
- [41] Fred Glover, Gary Kochenberger, and Yu Du. “Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models”. In: *4OR* 17.4 (2019), pp. 335–371.