Pegasus: The second connectivity graph for large-scale quantum annealling hardware

Nike Dattani*
Harvard-Smithsonian Center for Astrophysics

Szilard Szalay†
Wigner Research Centre for Physics

Nicholas Chancellor‡
Durham University

Pegasus is a graph which offers substantially increased connectivity between the qubits of quantum annealing hardware compared to the graph Chimera. It is the first fundamental change in the connectivity graph of quantum annealers built by D-Wave since Chimera was introduced in 2011 for D-Wave's first commercial quantum annealer. In this article we describe an algorithm which defines the connectivity of Pegasus and we provide what we believe to be the best way to graphically visualize Pegasus in 2D and 3D in order to see which qubits couple to each other. As Supplemental Material, we provide open source codes for generating Pegasus graphs.

The 128 qubits of the first commercial quantum annealer (D-Wave One, released in 2011) were connected by a graph called Chimera, which is rather easy to describe: A 2D array of $K_{4,4}$ graphs, with one partition of each $K_{4,4}$ being also connected to the same corresponding partition on the $K_{4,4}$ cell above it, and the other partition being connected to the same corresponding partition on the $K_{4,4}$ to the right of it (see Figure 1). The degree of the graph is six, since each qubit couples to four qubits within its $K_{4,4}$ unit cell, and to one qubit in a $K_{4,4}$ above it and one qubit in a $K_{4,4}$ to the right of it. All commercial quantum annealers built to date follow this graph connectivity, with just larger and larger numbers of $K_{4,4}$ unit cells (See Table 1).

Figure 1. The Chimera graph, with open edges to show that the pattern repeats.

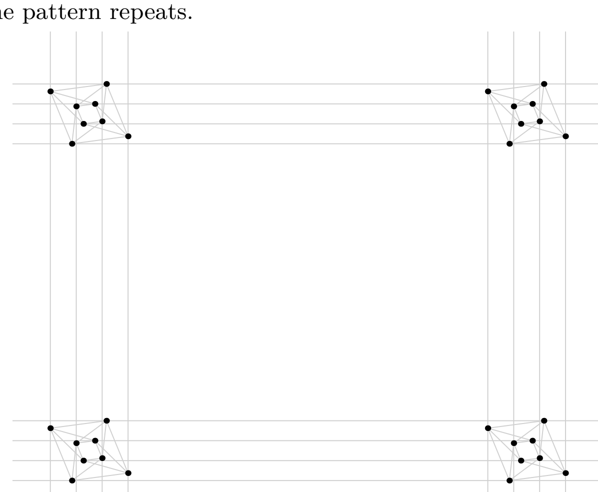


Table I. Chimera graphs in all commercial quantum computers to date.

	Array of $K_{4,4}$ unit cells	Total # of qubits
D-Wave One	4×4	128
D-Wave Two	8×8	512
D-Wave 2X	12×12	1152
D-Wave 2000Q	16×16	2048

In 2018, D-Wave announced the construction of a (not yet commercial) quantum annealer with a greater connectivity than Chimera offers, and a publicly available program (NetworkX) which allows users to generate Pegasus graphs of arbitrary size. However, no explicit description of the graph connectivity in Pegasus has been published yet, so we have had to apply the process of reverse engineering to determine it, and the following section describes our algorithm for generating Pegasus.

I. ALGORITHM FOR GENERATING PEGASUS

A. The vertices (qubits)

Start with Z layers of Chimera graphs, each being an $X \times Y$ array of $K_{4,4}$ unit cells (therefore we have an $X \times Y \times Z$ array of $K_{4,4}$ unit cells). The indices (x, y, z) will be used to describe the location of each unit cell along the indices corresponding to the dimension picked from (X, Y, Z). The values of X and Y are somewhat flexible, but for Pegasus, $Z = 3$. Each $K_{4,4}$ cell has two parts, labeled $i \in \{0, 1\}$, so that there are 4 qubits (vertices) for every (x, y, z, i). We will arbitrarily label these 4 qubits using two more labels: $(j, k) \in \{0, 1\}^2$. Therefore every qubit is associated with 6 indices: (x, y, z, i, j, k), with their ranges and descriptions given in Table II.

* n.dattani@cfa.harvard.edu
† szalay.szilard@wigner.mta.hu
‡ nicholas.chancellor@durham.ac.uk

Table II. Indices used to describe each qubit (vertex) in Pegasus

x	0 to $X-1$	Row within a Chimera layer
y	0 to $Y-1$	Column within a Chimera layer
z	0 to 2	Chimera layer
i	$0,1$	Bi-partition within $K_{4,4}$
j	$0,1$	First index within each part of $K_{4,4}$
k	$0,1$	Second index within each part of $K_{4,4}$

B. The edges (couplings) in Chimera

The $K_{4,4}$ cells are given by:

$$(x,y,z,0,j,k) \longleftrightarrow (x,y,z,1,j',k'). \tag{1}$$

This means for each $K_{4,4}$ cell, all vertices (j,k) for partition $i=0$ are coupled to all vertices (j',k') of partition $i=1$.

The horizontal lines between $K_{4,4}$ cells in Figure 1 can be described by adding 1 to x while keeping all other variables constant and setting $i=1$:

$$(x,y,z,1,j,k) \longleftrightarrow (x+1,y,z,1,j,k), \tag{2}$$

and the vertical lines can be described by adding 1 to y while keeping all other variables constant and setting $i=0$:

$$(x,y,z,0,j,k) \longleftrightarrow (x,y+1,z,0,j,k). \tag{3}$$

For each z, Equations (1), (3) and (2) define edges connecting vertices labeled by $x,y,i,$ and or j. This completes the definition of a Chimera graph. In all figures, these Chimera edges are grey.

C. The *new* edges (couplings) in Pegasus

1. *Edges within each $K_{4,4}$ cell:*

Pegasus first adds connections within each $K_{4,4}$ cell, given by simply coupling each vertex labeled as $k=0$ to its $k=1$ counterpart with all other variables unchanged:

$$(x,y,z,i,j,0) \longleftrightarrow (x,y,z,i,j,1). \tag{4}$$

These edges are drawn in black color in the figures.

2. *Edges connecting different $K_{4,4}$ cells:*

The rest of the new connections in Pegasus come from connecting the $K_{4,4}$ cells between different layers (different z) of Chimera graphs. The qubits of a $K_{4,4}$ cell located at coordinates (x,y,z) will be connected to 6 different $K_{4,4}$ cells on the other Chimera layers, and each pair of $K_{4,4}$ cells connected in this way will have 16 different connections in the form of 2 sets of $K_{2,4}$ graphs (each containing 8 connections). All of these connections are between vertices of one $K_{4,4}$ partition i the complementary partition $\bar{i}$ in a different $K_{4,4}$ (so $i=0$ vertices are coupled to $i=1$ vertices). In fact all connections will be of the form $(i,j,k) \rightarrow (\bar{i},j',k')$ where j' and k' can be any value in $0,1$.

For the $z=0,1$ layers, all $j=0$ vertices of a $K_{4,4}$ cell are connected to all j and k vertices of the opposite partition $\bar{i}$ in the $K_{4,4}$ cell with the same (x,y), but next layer $z{+}1$:

$$(x,y,z,i,0,k) \longleftrightarrow (x,y,z+1,\bar{i},j',k'), \tag{5}$$

and all $j=1$ vertices are also connected to all j and k vertices in the opposite partition $\bar{i}$ in a $K_{4,4}$ cell in the next layer $z+1$, but with x and y coordinates shifted by 1 in the following way:

$$(x,y,z,i,1,k) \longleftrightarrow (x+\bar{i},y+i,z+1,\bar{i},j',k'), \tag{6}$$

For the $z=2$ layer, all $j=0$ vertices of a $K_{4,4}$ cell are connected to the $K_{4,4}$ cells in the $z=0$ layer, but with x and y coordinates *both* shifted by 1:

$$(x,y,2,i,0,k) \longleftrightarrow (x+1,y+1,0,\bar{i},j',k'). \tag{7}$$

Equations 5-7 each describe a pair of $K_{2,4}$ graphs (16 edges in total), one connecting the $i=0$ partition of a $K_{4,4}$ cell at (x,y,z) to the $i=1$ partition of a $K_{4,4}$ on a different layer z and possibly different coordiante (x',y'), and the other $K_{2,4}$ connecting the $i=1$ partition to the $i=0$ partition of a $K_{4,4}$ on a different layer z.

All three equations 5-7 can be sumarized by one:

$$(x,y,z,i,j,k) \leftrightarrow \left(x+j\bar{i}+\delta_{z,2},\, y+ji+\delta_{z,2},(z+1)\bmod 3,\bar{i},j',k'\right).$$

A representative part of the Pegasus graph is depicted in Figure 3.

II. ALTERNATIVE GRAPHICAL REPRESENTATIONS OF PEGASUS

There are many ways that Pegasus can be drawn, so we show some of these in Figures 4 and 5.

III. COMPARISON TO CHIMERA

A. Graph degree

As the yellow vertex in Fig. ? indicates, the vertices of Pegasus (except for vertices on an edge) have a degree

Figure 2. Connections of a generic unit cell of the Pegasus graph. (grey: Chimera graph (1), (2), (3); black: (4); blue: (5) and (**??**); red: (**??**) and (**??**); green: (**??**); orange: (**??**). Solid/dashed lines start/end on the fixed (i, j, k) unit cell.)

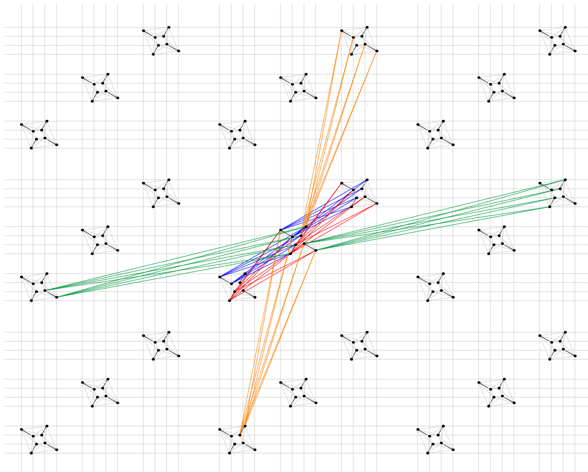
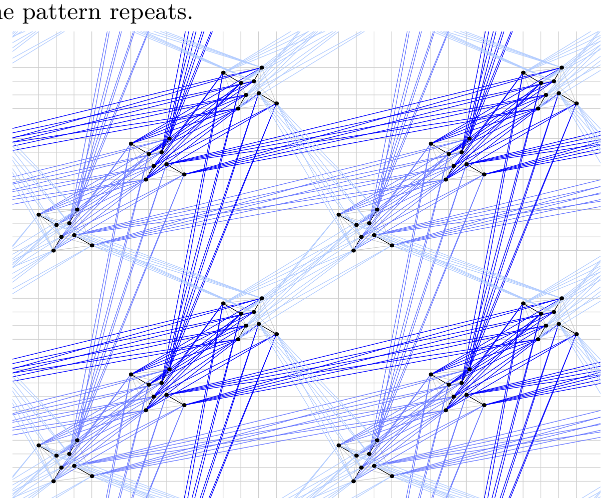


Figure 3. The Pegasus graph, with open edges to show that the pattern repeats.



15, which is 2.5 times larger than the maimum degree achieved in Chimera.

B. Non-planarity

We note that certain binary optimization problems forming planar graphs can be solved on a classical computer with a number of operations that scales polynomially with the number of binary variables, with the blossom algorithm [1, 2]. Therefore it is important that the qubits of a quantum annealer are connected by a non-planar graph. The $K_{4,4}$ cells of Chimera are already sufficient to make all commercial D-Wave annealers non-planar. However, if each $K_{4,4}$ cell of Chimera physical qubits were to encode one logical qubit (in for example, an extreme case of minor embedding), then Chimera would be planar. While all red, blue, and green lines added in Pegasus are of the form $K_{2,4}$, which itself is planar; these $K_{2,4}$ lines connect cells of different planes of chimeras in a non-planar way, such that even if each cell were to represent one logical qubit, *these logical qubits would still form a non-planar graph in Pegasus.* This should expand the number of binary optimization problems that can be embedded onto a D-Wave annealer, and cannot be solved on a classical computer with a number of operations that scales polynomially with the number of binary variables.

C. Embedding of quadratization gadgets

We have written an entire paper on the embedding of quadratization gadgets onto Chimera and Pegasus. One highlight of that work is the fact that *all* quadratization gadgets for single cubic terms which require one auxiliary qubit, can be embedded onto Pegasus with no further auxiliary qubits because Pegasus contains K_4,which means that all three logical qubits and the auxiliary qubit can be connected in any way, without any minor-embedding.

D. Application: Nike will embed two problems on Chimera and Pegasus to show reduction in # of auxiliary qubits.

[1] J. Edmonds, Canadian Journal of Mathematics **17**, 449 (1965).

[2] J. Edmonds, *JOURNAL OF RESEARCH of the Na ti onal Bureau of Standards-B. Mathematics and Mathematical Physics*, Tech. Rep. 2 (1965).

Figure 4.

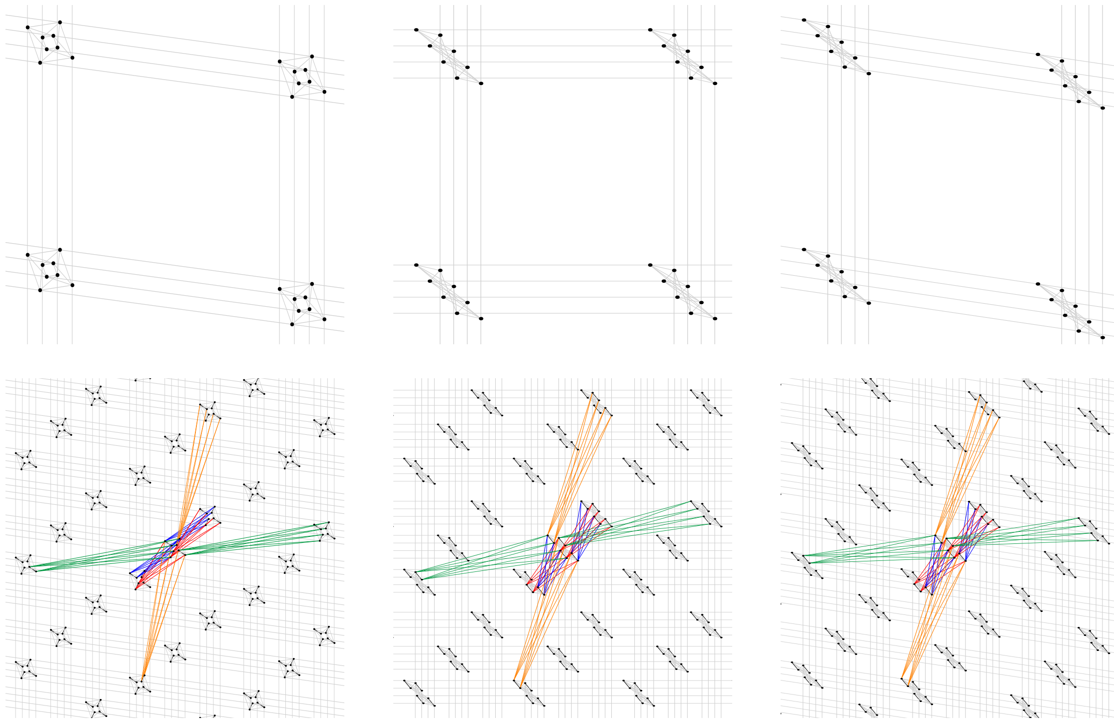


Figure 5.