Embedding quadratization gadgets on Chimera and Pegasus graphs

Nike Dattani*
Harvard-Smithsonian Center for Astrophysics

Nicholas Chancellor†
Durham University

We group all known quadratizations of cubic and quartic binary optimization problems into five and seven unique graphs respectively. We then perform a minor embedding of these graphs onto the well-known Chimera graph, and the brand new *Pegasus* graph. In cases where two or more graphs have a minor embedding with the same overhead in terms of auxiliary variables, we make recommendations for which gadgets are best to use for certain problems.

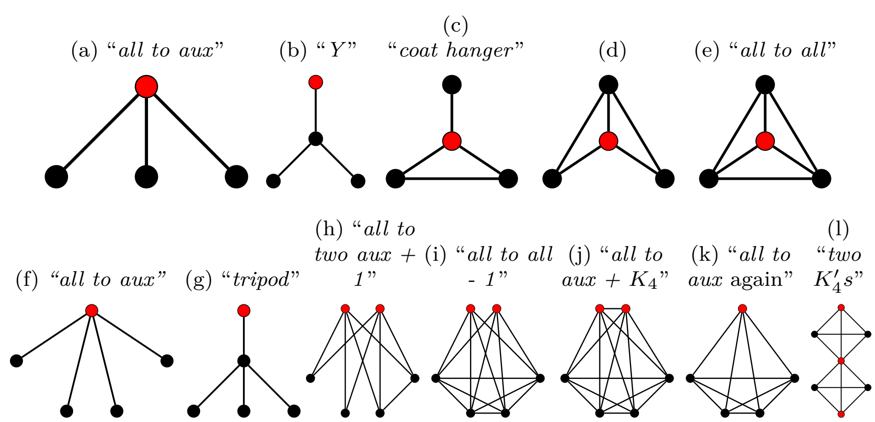
Discrete optimization problems are often naturally formulated in terms of minimizing some polynomial of degree > 2, which is then 'quadratized' into a quadratic function which can be solved using standard algorithms for universal classical computers [1], using special-purpose classical annealers [2], or using quantum annealers [3]. With dozens of quadratization methods available [4], one should choose the best quadratization for a given problem, and for a given method for solving the problem.

There are ways to quadratize functions of discrete variables without adding any auxiliary variables [5], but when those methods cannot be applied we introduce auxiliary variables. The resulting quadratic functions (called 'gadgets') that accurately or exactly simulate the original high-degree functions, will have some connectivity between the binary variables (or bits, or qubits, herein referred to for convenience only as qubits) which can be represented by a graph in which vertices represent qubits and edges indicate when two different qubits appear together in a quadratic term. Since this graph incorporates no information about the linear terms, constant term, or the coefficients of the quadratic terms, many different gadgets have the same graph, therefore in this paper we will classify all known quadratization gadgets into categories according to their corresponding graph (herein called their 'gadget graph').

Gadget graphs for all known single cubic terms and for all known single quartic terms are given in Figure 1. Gadget graphs tell us a lot about how costly the quadratic optimization problem will be, and those with larger connectivity tend to yield more difficult functions to optimize. Furthermore, some optimization methods only work if their corresponding graph has a certain connectivity, two examples of such connectivities being the ones in D-Wave's well-known Chimera graph [5], and in their very recently presented *Pegasus* graph, both shown in Fig. 2.

Any graph, can be mapped onto the Chimera or Pegasus graphs by minor-embedding [6], where the Chimera graph or the Pegasus graph is a graph minor of the graph representing the problem that needs to be optimized. This often means that one binary variable in the quadratic optimization problem needs to be represented by two qubits instead of one, making the number of qubits needed to solve the original problem much larger than before, and sometimes

Figure 1



* n.dattani@cfa.harvard.edu
† nicholas.chancellor@durham.ac.uk

Figure 2

completely impossible. For example a quartic function with 1000 binary variables has $\binom{1000}{3} > 166$ million possible cubic terms and $\binom{1000}{4} > 40$ billion possible quartic terms which have to be quadratized, and then minor-embedded. If our minimization method can only be applied for up to 50 billion qubits, we cannot afford for each quartic-to-quadratic gadget to require its own auxiliary qubit for minor-embedding.

We have provided minor-embeddings for all gadget graphs in Figure 1, for both Chimera and Pegasus. We note that ***all*** cubic to quadratic gadgets involvnig one auxiliary qubit can be embedded onto Pegasus without any further auxiliary qubits for the embedding, because Pegasus contains the K_4 graph, which means any possible connections between the three logical qubits and the one auxiliary qubit are already contained in Pegasus. Since Chimera does not contain K_4, only negative cubic terms are so far known to be quadratizable with gadgets that embed directly onto Chimera without any extra qubits for the embedding.

I. MINOR EMBEDDINGS FOR CUBIC TO QUADRATIC GADGETS

A. Chimera graph

B. Pegasus graph

Table I

Gadget Graph	Example Gadgets	N_{aux} Quadratization	N_{aux} Chimera	N_{aux} Pegasus
		Cubic $\to$ Quadratic		
All to Aux	NTR-KZFD NTR-ABCG	1	0	0
Y	NTR-ABCB	1	0	0
Coat Hanger	AR	1	1	0
(d)	ACR	1		0
All to All	PTR-Ishikawa PTR-BCR(1-4) PTR-KZ	1		0
		Quartic $\to$ Quadratic		
All2Aux	NTR-KZFD	1	0	0
Tripod	NTR-ABCB	1	0	0
All22Aux+1	PTR	2	1	0
All2All - 1	PTR-Ishikawa	2	?	2
All2Aux+K_4	PTR-BCR-2 PTR-BCR-4	1	3	2
All2Aux	PTR-BCR-3	2	?	2
Two $K_4's$	PTR-KZ	3	?	1

Figure 3: Minor embeddings of all cubic to quadratic gadgets onto a 'unit cell' of a Chimera graph. Transparent vertices and edges are not used. Thick black edges denote graph minors, in which two physical qubits (two vertices) represent one logical qubit (this is done when logical qubits need to be connected to more qubits than the Chimera unit cell otherwise allows).

(a) "*all to aux*" (b) "*Y*" (c) "*coat hanger*" (d) "" (e) "*all to all*"

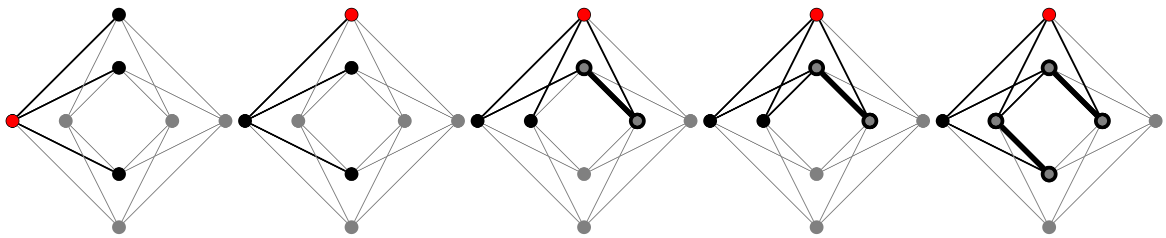


Figure 4: Minor embeddings of all cubic to quadratic gadgets onto a 'unit cell' of a Pegasus graph. Transparent vertices and edges are not used. Figs 3a and 3b do not need to be altered since the Chimera graph is a sub-graph of Pegasus, so only the gadgets that required auxiliary qubits for minor embedding onto Chimera are embedded for Pegasus here to show that with Pegasus no auxiliary qubits are needed for cubic to quadratic gadgets.

(a) "*coat hanger*" (b) "" (c) "*all to all*"