Embedding quadratization gadgets on Chimera and Pegasus graphs

Nike Dattani*
Harvard-Smithsonian Center for Astrophysics

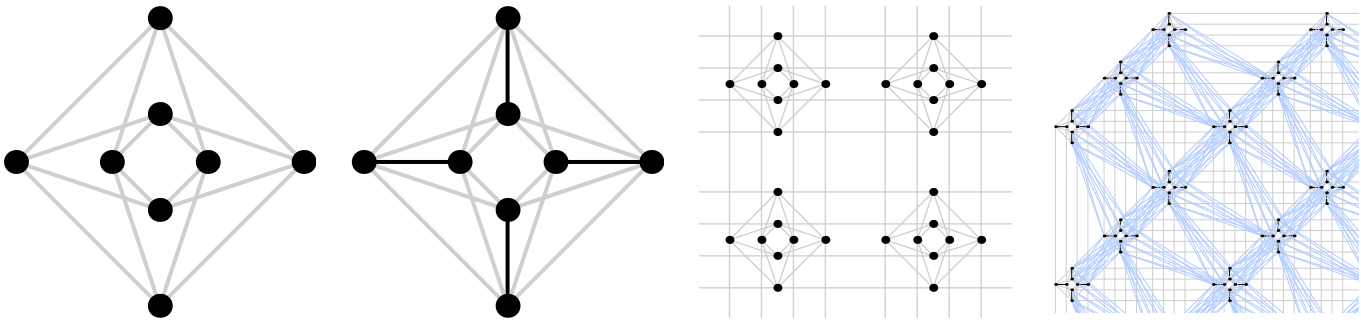
Nicholas Chancellor†
Durham University, Joint Quantum Centre

We group all known quadratizations of cubic and quartic terms in binary optimization problems into five and six unique graphs respectively. We then perform a minor embedding of these graphs onto the well-known Chimera graph, and the brand new *Pegasus* graph. We conclude with recommendations for which gadgets are best to use when aiming to reduce the total number of qubits required to embed a problem.

Discrete optimization problems are often naturally formulated in terms of minimizing some polynomial of degree > 2 [1], which is then 'quadratized' [2] into a quadratic function which can be solved using standard algorithms for universal classical computers [3], using special-purpose classical annealers [4], or using quantum annealers [5]. With more than 40 quadratization methods available for binary optimization problems [2], one should choose the best quadratization for a given problem, and for a given method for solving the quadratized problem.

There are ways to quadratize functions of discrete variables without adding any auxiliary variables [2, 6–9], but when those methods cannot be applied we introduce auxiliary variables. The resulting quadratic functions (called 'gadgets'), that with good accuracy (or sometimes exact accuracy) 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').

Figure 1: Graph connectivities for D-Wave's Chimera and Pegasus graphs.
(a) Chimera (single cell) **(b)** Pegasus (single cell) **(c)** Chimera (repeating) **(d)** Pegasus (repeating)

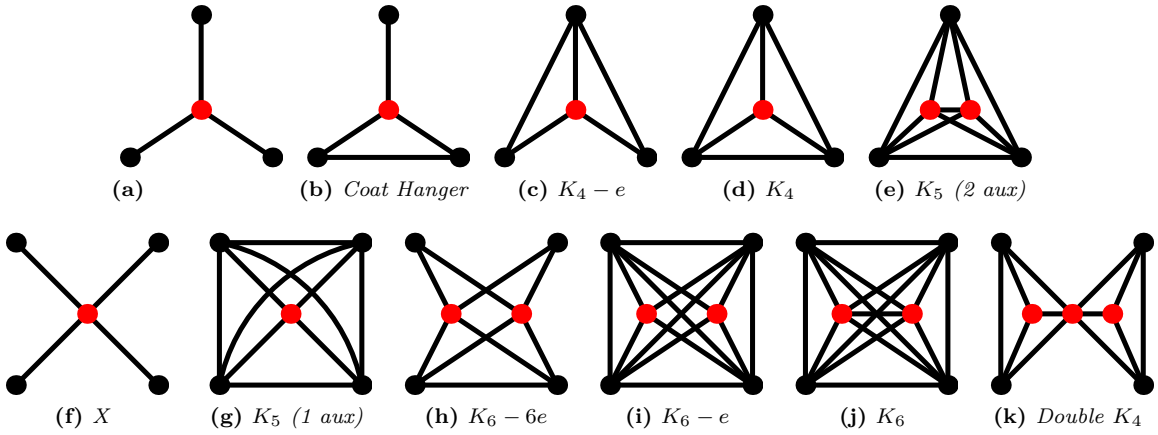


Gadget graphs for all known single cubic terms and for all known single quartic terms are given in Fig. 2. Gadget graphs tell us a lot about how costly the quadratic optimization problem will be, and those with larger connectivity tend to yield more difficult functions to optimize. Furthermore, some 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 [10], and in their very recent *Pegasus* graph, both shown in Fig. 1.

Any graph, can be mapped onto the Chimera or Pegasus graphs by minor-embedding [24, 25], 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 (or more) qubits (called a 'chain') instead of one, making the number of physical qubits needed to solve the original problem larger than before, and sometimes impossibly large to solve. For example if we wanted to factor the smallest RSA number which has not yet been factored on a classical computer (RSA-230) by minimization of a corresponding binary

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

Figure 2: *Gadget graphs.* Graphs showing the connectivity between qubits in quadratization gadgets for cubic to quadratic gadgets (top row), and quartic to quadratic gadgets (bottom row). Red vertices represent auxiliary qubits and black vertices represent logical qubits. Black edges denote the existence of a quadratic term in the gadget, involving the two corresponding qubits represented by vertices connected by the edge. *Linear and constant terms in the gadgets are completely ignored here.*



optimization problem, we would have to minimize a quartic function of 6594 variables which can be turned into a quadratic problem involving 148 776 variables [26]. Say we have a quantum annealer with 149 000 qubits: This means that only 224 out of the 142 182 required gadget graphs can afford an auxiliary qubit for minor-embedding into the hardware's native connectivity graph. We therefore want *efficient gadgets* which can be minor-embedded into our hardware with *zero* or *very few* extra qubits.

In this paper we have provided minor-embeddings for all gadget graphs in Fig. 2, for both Chimera and Pegasus. We note that ***all*** cubic to quadratic gadgets involving 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. Furthermore, *all* cubic and quartic gadgets are embeddable with a single 'cell' of Pegasus and with chains of length at most two, whereas with Chimera cells, three of the quartic gadgets require multiple cells and two require chains of length three.

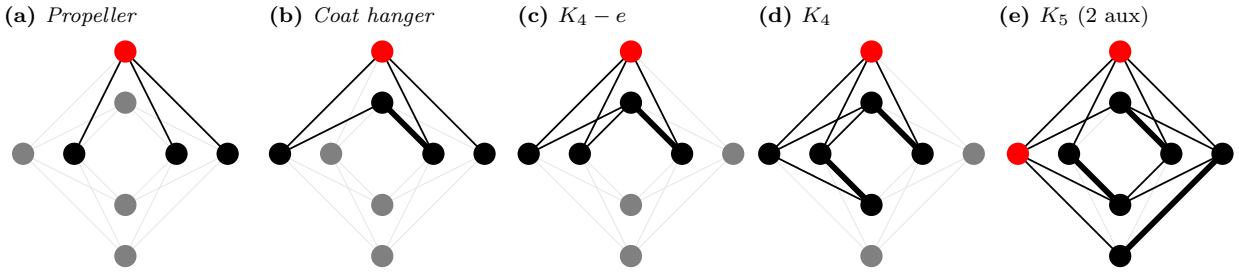
Table I: Grouping of all known quadratization gadgets for single ***cubic*** terms, into categories corresponding to their gadget graphs. For each unique gadget graph, the number of auxiliary qubits required for construction of the gadget is listed, followed by the number of auxiliary qubits required to minor-embed the gadget graph onto Chimera and Pegasus.

Gadget Graph	Example Gadgets (see [2] for definitions)	Reference	N_{aux} **Quadratization**	N_{aux} **Embedding** (Chimera)	N_{aux} **Total** (Chimera)	N_{aux} **Embedding** (Pegasus)	N_{aux} **Total** (Pegasus)
Propeller	NTR-KZFD NTR-ABCG-2	[2, Pg.7][11, 12] [2, Pg.9][13]	1	0	**1**	0	**1**
Coat hanger	PTR-GBP	[2, Pg.23][14]	1	1	**2**	0	**1**
$K_4 - e$	NTR-GBP NTR-ABCG	[2, Pg.10][14] [2, Pg.8][15]	1	1	**2**	0	**1**
K_4	PTR-Ishikawa NTR-RBL-(3→ 2) PTR-BCR-1,2,3,4 PTR-KZ	[2, Pg.14][16] [2, Pg.11][17] [2, Pg.15-18][18, 19] [2, Pg.21][11, 20, 21]	1	2	**3**	0	**1**
K_5 (2 aux)	PTR-RBL	[2, Pg.24][17]	2	3	**5**	1	**2**

I. MINOR EMBEDDINGS FOR CUBIC TO QUADRATIC GADGETS

A. Chimera graph

Figure 3: Minor embeddings of all ***cubic*** to quadratic gadgets for single terms, onto a 'unit cell' of a Chimera graph. Grey vertices and edges are not used. Thick edges denote chains for minor embedding, in which two or more qubits (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).



B. Pegasus graph

Figure 4: Minor embeddings of all ***cubic*** to quadratic gadgets for single terms, onto a Pegasus 'cell'. Grey vertices and edges are not used.

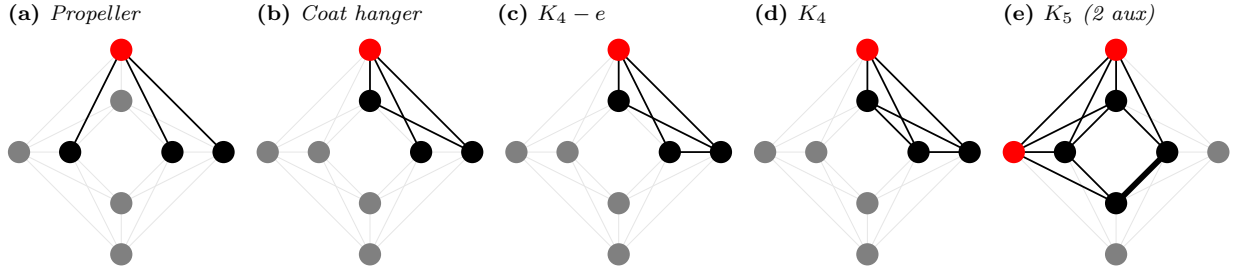


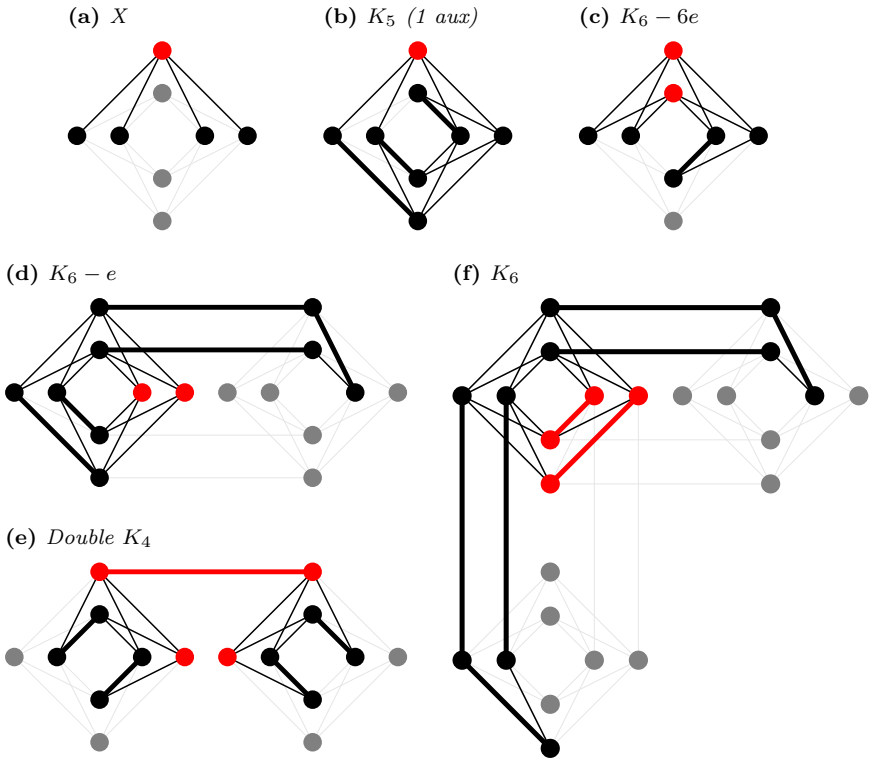
Table II: Grouping of all known quadratization gadgets for single ***quartic*** terms, into categories corresponding to their gadget graphs. For each unique gadget graph, the number of auxiliary qubits required for construction of the gadget is listed, followed by the number of auxiliary qubits required to minor-embed the gadget graph onto Chimera and Pegasus.

Gadget Graph	Example Gadgets (see [2] for definitions)	Reference	N_{aux} **Quadratization**	N_{aux} **Embedding** (Chimera)	N_{aux} **Total** (Chimera)	N_{aux} **Embedding** (Pegasus)	N_{aux} **Total** (Pegasus)
X	NTR-KZFD NTR-ABCG-2	[2, Pg.7][11, 12] [2, Pg.9][13]	1	0	**1**	0	**1**
K_5 (1 aux)	PTR-BCR-2 PTR-BCR-4 NTR-LHZ	[2, Pg.16][18, 19] [2, Pg.18][18, 19] [2, Pg.12][22]	1	3	**4**	1	**2**
$K_6 - 6e$	PTR-BG	[2, Pg.13][23]	2	1	**3**	0	**2**
$K_6 - e$	PTR-Ishikawa	[2, Pg.14][16]	2	5	**7**	2	**4**
K_6	PTR-BCR-3	[2, Pg.17][18, 19]	2	8	**10**	2	**4**
Double K_4	PTR-LZL	[2, Pg.26][20]	3	5	**8**	1	**4**

II. MINOR EMBEDDINGS FOR QUARTIC TO QUADRATIC GADGETS

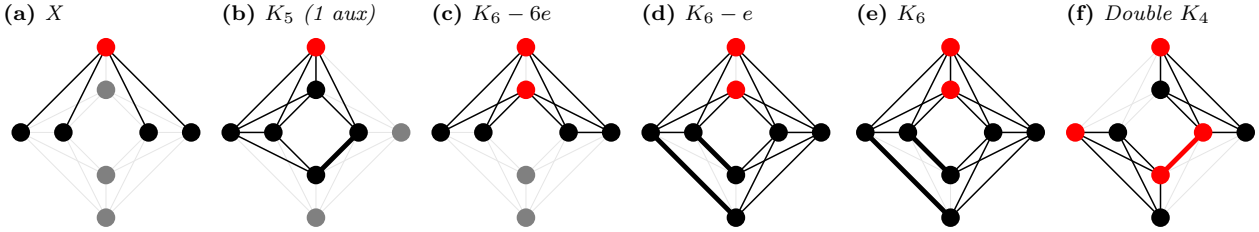
A. Chimera graph

Figure 5: Minor embeddings of all ***quartic*** to quadratic gadgets for single terms, onto a 'unit cell' of a Chimera graph. Grey vertices and edges are not used. Thick edges denote chains for minor embedding, in which two or more qubits (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).



B. Pegasus graph

Figure 6: Minor embeddings of all ***quartic*** to quadratic gadgets for single terms, onto a Pegasus 'cell'. Grey vertices and edges are not used.



III. RECOMMENDED GADGETS

When reducing the number of total qubits is the most important factor in compiling a discrete optimization problem, negative terms (whether cubic or quartic) can be quadratized with only 1 auxiliary qubit, and in such a way not to require any additional qubits for embedding onto either Chimera and Pegasus (so only 1 auxiliary qubit is required in total).

For positive terms the situation is a bit more complicated: Still only 1 auxiliary qubit is required for quadratization

of cubic terms, and for Pegasus the quadratization can be done in such a way as to require no further auxiliary qubits for the minor-embedding, however embedding with Chimera will require at minimum 1 further auxiliary qubit (so 1 auxiliary qubit is required in total for Pegasus, but 2 auxiliary qubits are required in total for Chimera).

For quartic positive terms, we need at least 2 total auxiliary qubits on Pegasus and at least 3 auxiliary qubits for Chimera. It is interesting to note that for positive quartic terms, while the gadgets leading to the K_5 graph require 1 fewer auxiliary qubit than the gadgets leading to the $K_6 - 6e$ graph for quadratization, the graph needs 2 more auxiliary qubits than $K_6 - 6e$ for embedding onto Chimera, and 1 more auxiliary qubit for embedding onto Pegasus (which requires 0 in the former case). Therefore, while gadgets leading to K_5 may appear at first sight to be more efficient than gadgets leading to $K_6 - 6e$ due to requiring fewer qubits for quadratization, they require more *total* auxiliary qubits for embedding onto Chimera and an equivalent number of total qubits for embedding onto Pegasus.

ACKNOWLEDGMENTS

We gratefully thank Nick Wormald of Monash University for helpful comments about our gadget graphs. We gratefully thank Szilard Szalay from the Wigner Research Centre for Physics of the Hungarian Academy of Sciences for assistance with making the graphs in Figures 1 and 2. NC was funded by EPSRC (Project: EP/S00114X).

[1] N. S. Dattani and N. Bryans, (2014), arXiv:1411.6758.
[2] N. Dattani, *Quadratization in discrete optimization and quantum mechanics* (2019) arXiv:1901.04405.
[3] E. Boros and P. L. Hammer, Discrete Applied Mathematics **123**, 155 (2002).
[4] M. Yamaoka, C. Yoshimura, M. Hayashi, T. Okuyama, H. Aoki, and H. Mizuno, IEEE Journal of Solid-State Circuits **51**, 303 (2016).
[5] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose, Nature **473**, 194 (2011).
[6] H. Ishikawa, in *2014 IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2014) pp. 1362–1369.
[7] R. Tanburn, E. Okada, and N. Dattani, *Reducing multi-qubit interactions in adiabatic quantum computation without adding auxiliary qubits. Part 1: The "deduc-reduc" method and its application to quantum factorization of numbers* (2015) arXiv:1508.04816.
[8] E. Okada, R. Tanburn, and N. S. Dattani, *Reducing multi-qubit interactions in adiabatic quantum computation without adding auxiliary qubits. Part 2: The "split-reduc" method and its application to quantum determination of Ramsey numbers* (2015) arXiv:1508.07190.
[9] R. Dridi and H. Alghassi, Scientific Reports **7**, 43048 (2017).
[10] H. Neven, V. S. Denchev, M. Drew-Brook, J. Zhang, W. G. Macready, and G. Rose, *NIPS 2009 Demonstration: Binary Classification using Hardware Implementation of Quantum Annealing*, Tech. Rep. (2009).
[11] V. Kolmogorov and R. Zabih, IEEE Transactions on Pattern Analysis and Machine Intelligence **26**, 147 (2004).
[12] D. Freedman and P. Drineas, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 2 (IEEE, 2005) pp. 939–946.
[13] M. Anthony, E. Boros, Y. Crama, and A. Gruber, Discrete Applied Mathematics **203**, 1 (2016).
[14] A. C. Gallagher, D. Batra, and D. Parikh, in *CVPR 2011* (IEEE, 2011) pp. 1857–1864.
[15] M. Anthony, E. Boros, Y. Crama, and A. Gruber, (2014), arXiv:1404.6535.
[16] H. Ishikawa, IEEE Transactions on Pattern Analysis and Machine Intelligence **33**, 1234 (2011).
[17] A. Rocchetto, S. C. Benjamin, and Y. Li, Science Advances **2** (2016), 10.1126/sciadv.1601246.
[18] E. Boros, Y. Crama, and E. Rodríguez-Heck, *Quadratizations of symmetric pseudo-Boolean functions: sub-linear bounds on the number of auxiliary variables*, Tech. Rep. (2018).
[19] E. Boros, Y. Crama, and E. Rodriguez Heck, *Compact quadratizations for pseudo-Boolean functions* (2018).
[20] N. Chancellor, S. Zohren, and P. A. Warburton, npj Quantum Information **3**, 21 (2017).
[21] M. Leib, P. Zoller, and W. Lechner, Quantum Science and Technology **1**, 15008 (2016).
[22] W. Lechner, P. Hauke, and P. Zoller, Science Advances **1**, e1500838 (2015).
[23] E. Boros and A. Gruber, (2014), arXiv:1404.6538.
[24] V. Choi, Quantum Information Processing **7**, 193 (2008).
[25] V. Choi, Quantum Information Processing **10**, 343 (2011).
[26] Z. Li, N. S. Dattani, X. Chen, X. Liu, H. Wang, R. Tanburn, H. Chen, X. Peng, and J. Du, http://arxiv.org/abs/1706.08061 (2017), arXiv:1706.08061.