An Optimal Constraint for QUBO Models

Clark Alexander*
email: the author
January 11, 2021

Abstract
A quadratic binary unconstrained optimization model, hereafter QUBO, by definition is unconstrained. This, however, is not ideal if one needs to select a model containing only a fixed size binary vector. In this work we show how to add a constraint to a QUBO to force a particular size solution.

1 Introduction
As the industry of quantum computing has been making inroads in applications, the first set of problems which are in-line for fast, efficient “solutions” are combinatorial optimization problems \cite{CQ1, DwaveI, STB} As of this writing, the technique which appears to be furthest along is quantum annealing. This is an appealing technique as it allows one with a reasonably powerful laptop and a high level language compiler to benchmark the quantum annealing process. Generally speaking, the most-straightforward way to apply quantum annealing is to pose one’s problem in terms of a quadratic form. In particular, if one can pose a problem in terms of a solution which marks a set of elements as “in” or “out” then quantum annealing works well. Additionally, there are many classical, heuristic, and probabilistic techniques which one can use to benchmark a quantum annealer; for example, simulated annealing, genetic algorithms, simulated bifurcation machines, branch-and-bound, etc. \cite{DA, GKD, SBM, EKU}.

However, quadratic forms with “in”/“out” solutions require binary vectors and depending on the particular type of quadratic form given, one can expect a very specific set of solutions to show up. For example, if one provides a positive definite matrix as a quadratic form then the known minimum is the zero vector. There is no need to apply probabilistic techniques, all non zero vectors will give positive solutions and in searching for a minimum, we cannot go lower than zero. Additionally if the quadratic form is a well known random matrix, a Gaussian ensemble, for instance, then the best solution will contain roughly half “in” and half “out.” These solutions are expected, but maybe not completely useful.

*Chicago Quantum
Recently, there has been a push for using quadratic forms in financial optimization [CQ3, CQ2, MLO] One common strategy is to maximize expected return while minimizing volatility. Volatility is, however, covariance, which is positive semi-definite. In reality covariance is positive definite with real data. Thus trying to minimize a positive definite matrix reveals the “mathematically correct” solution of not investing as that reduces volatility to zero. However, for an investor, this is not a useful solution. Perhaps an investor wants to invest in an “optimal” set of 15 assets. An unconstrained problem will never produce such an answer, thus one wishes to add a constraint in a somewhat natural way so that a set of 15 assets becomes visible.

2 Elementary Properties of a Random QUBO

A quadratic unconstrained binary optimization (hereafter QUBO) can be thought of as a square matrix. While this is a slight abuse of notation, there should not be too much confusion as the actual optimization involves finding a vector for the particular matrix. So where it is clear we will refer to the matrix and the optimization problem both as QUBO.

Definition 1. Given a real symmetric matrix $A \in \mathbb{R}^{N \times N}$ and a real vector $B \in \mathbb{R}^N$, a quadratic unconstrained binary optimization is a problem in which one seeks the vector $x \in \{0, 1\}^N$ so that

$$Q(A, B) = x^t Ax + B \cdot x$$

(1)

achieves a minimum value. That is

$$\min_x x^t Ax + B \cdot x$$

(2)

It is also common to expect $A$ to be upper or lower triangular and traceless. However by considering $(A + A^t)/2$ one can trivially move back and forth between triangular and symmetric matrices. We prefer the symmetric matrix as it guarantees real eigenvalues and is more in-line with the principles of quantum mechanics and therefore allows one to pose a matrix $A$ (or a small variant thereof) as a Hamiltonian in an Ising model which is the current technology used in quantum annealing and simulated bifurcation machines. This also allows one the ability to easily translate between quantum, digital, and simulated annealing for benchmarking purposes.

Lemma 2. While considering a binary vector $x \in \{0, 1\}^N$ one can reduce a QUBO into a single matrix (which need not be traceless). That is one can reduce

$$x^t Ax + B \cdot x \rightarrow x^t \tilde{A}x$$

where

$$\tilde{A} = A + \text{diag}(B)$$

(3)

with $\text{diag}(B)$ being a diagonal matrix with

$$\text{diag}(B)_{ij} = B_i \delta_{ij}$$
Proof. Since each $x_i$ is 0 or 1 we trivially have $x_i = x_i^2$. This reveals

$$B \cdot x = x^t \text{diag}(B)x$$

Which allows us to factor $x^t$ on the left and $x$ on the right

$$x^tAx + B \cdot x = x^tAx + x^t \text{diag}(B)x = x^t(A + \text{diag}(B))x = x^t\tilde{A}x \quad (4)$$

From here forward we shall simply refer to $\tilde{A}$ as $A$.

In quantum annealing, the vector $x_i \in \{0, 1\}$ is exchanged for a vector $z_i \in \{-1, 1\}$ with the simple transformation

$$z_i = 2x_i - 1$$

This affects our matrix $A$ and also gives one an offset vector where

$$x^tAx = z^tJz + C \cdot z + \text{const} \quad (5)$$

The matrix $J = A/4$ and offset vector $C$ are calculated by a simple change of variables. For the purposes of this article we will remain in the space $x_i \in \{0, 1\}$ and only mention that can transform when necessary.

3 The Constraint Matrix

Given our model

$$\min_{x} x^tAx$$

with $A$ a real symmetric matrix of size $N \times N$ we wish to add a constraint matrix $C$ to $A$ so that our vector $x$ has norm $\|x\|^2 = m$; or more simply the $L^1$ norm is $M$, $\|x\|_1 = M$ for some $1 \leq M \leq N$.

Thus our new model becomes

$$\min_{x} x^t(A + C)x$$

If we pass this quadratic form to a solver (whether a simulated annealer or quantum annealing computer or a simulated bifurcation machine) we can expect that a “good” solver will produce a vector $x$ of the required size.

**Theorem 3.** Let $A$ be a real symmetric matrix of size $N \times N$. Then the addition of matrix

$$C = \alpha J_N + \beta I_N \quad (6)$$

to $A$ will produce a QUBO with optimal vector $x$ of size $\|x\|_1 = M$ when

$$M = \frac{-\beta}{2\alpha} \quad (7)$$

In particular there is a line of solutions. The larger $\alpha$ the stronger the pull toward $\|x\|_1 = M$.  

3
Proof. Since we wish to minimize $x^t Ax$ where $|x|_1 = M$ we consider the addition of the matrix $C = \alpha J_N + \beta I_N$ where $J_N$ is the $N \times N$ matrix of all ones and $I_N$ is the $N \times N$ identity matrix.

This reduces our calculation to

$$\min_x x^t (A + C)x \implies \min_{|x|_1 = M} x^t Cx$$ (8)

Since $J_N$ and $I_N$ yield straightforward multiplications we have

$$x^t (\alpha J_N + \beta I_N)x = \alpha M^2 + \beta M$$ (9)

Now we see the results directly. We choose $\alpha$ and $\beta$ to minimize $x^t Cx$ by

$$M = \frac{-\beta}{2\alpha}$$ (10)

Noting that $\alpha > 0$ gives a minimum and $\alpha < 0$ gives a maximum.

Further more we can reparameterize this to be a single parameter constraint as

$$C(\alpha) = \alpha(J_N - 2MI_N)$$ (11)

We can also see this as

$$C(\alpha) = \alpha \begin{bmatrix} 1 - 2M & 1 & 1 & \ldots & 1 \\ 1 & 1 - 2M & 1 & \ldots & 1 \\ 1 & 1 & 1 - 2M & \ldots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \ldots & 1 - 2M \end{bmatrix}$$

\[\square\]

4 Two Examples

Example 4. Let’s take a quick look at how to simulate this in a modern computing language. We’ll produce a random symmetric matrix of size $N \times N$ and require a solution of size $M$ where $M$ is significantly different from $N/2$.

In the code on the author’s local computer, we will run a simulated annealer which produces two outputs, (a) the cost $x^t Ax$, (b) the solution vector.

Algorithm 1 Picking M asets; Julia/Octave style

```plaintext
A = randn(N, N)
A = (A + A')/2
C(\alpha) = \alpha * (ones(N, N) - 2M * eye(N))
cost, solution = anneal(A + C(0.5))
sum(solution)
```

4
Using five hundred runs with parameters $N = 30, M = 8$ we have a histograms for $\alpha \in \{0, 0.1, 0.2, 0.5, 1, 2, 10\}$. When $\alpha = 0$ we have the unconstrained model, where one expects a solution of size $15 \pm \sqrt{30}$ so we expect between 10 and 20 as the global best solution. We are running our simulated annealer cooling very quickly and only a few trials per degree so as to show the efficacy of the constraint. We can see the pull toward 8 assets as $\alpha$ increases.

![Histograms with increasing constraints](image)

Figure 1: Histograms with increasing constraints

The final frame in 2 is a stacked histogram showing all 7 histograms together. We see the density of selections increasing as we move toward 8. Looking all the way back to the first figure in 1 we see that the global minimum is likely larger than 15, most likely at 17. If we were to have a larger Gaussian ensemble or one in which the global minimum is at or below $N/2$ then the shift toward a smaller number of selected assets is even clearer. In this case, if we had selected the number of assets to be 20, we would see the rightward shift in the histograms more strongly.

Example 5. For our second example, we’ll use a positive semi-definite matrix of size $30 \times 30$ and again apply constraints to select 8 assets.

It is important to note that the simulated annealer on the author’s computer tends to avoid picking exactly zero assets. Thus in our first figure in 2, we see a split between zero and one, even though we know the mathematically sound answer is exactly zero.
We see the efficacy of the constraint here much more prominently. Simulated annealers are not necessary with such small scale QUBO as $30 \times 30$ as a brute force solution can be obtained in a matter of minutes. Additionally, small shifts in small QUBO models don’t have such a large effect. If one were to repeat these examples with $N = 5000$ and $M = 50$ the effects of $\alpha = 0.1$ would appear much more clearly. Nonetheless even with $\alpha$ as small as 0.2 we begin seeing solutions with 8 assets. At $\alpha = 0.5$ the lion’s share are solutions with 8 assets (442 out of 500 in this particular numerical experiment). And at $\alpha = 2$ and above we select 8 assets without fail. Thus is the last figure in [4] we see an intense density at 8 assets. With all other assets smaller. This is consistent with the first example in which 8 assets becomes the minimum number of assets chosen as the unconstrained solution has greater than 8 assets, in this case the unconstrained solution has fewer.

References

[CQ1] Cohen, J., Alexander, C., Picking Efficient Portfolios from 3,171 US Common Stocks with New Quantum and Classical Solvers https://arxiv.org/abs/2011.01308
Figure 3: Histograms with increasing constraints

[CQ2] Cohen, J., Khan, A., Alexander, C., Portfolio Optimization of 60 Stocks Using Classical and Quantum Algorithms https://arxiv.org/abs/2008.08669

[CQ3] Cohen, J., Khan, A., Alexander, C., Portfolio Optimization of 40 Stocks Using the DWave Quantum Annealer https://arxiv.org/abs/2007.01430

[DA] Fujitsu Research Lab, Quantum Future - Quantum Present https://sp.ts.fujitsu.com/dmsp/Publications/public/wp-da-overview-ww-en.pdf

[DwaveI] https://docs.dwavesys.com/docs/latest/c_gs_3.html

[EKU] Eren, Y., Küçükdemiral, İ., Üstoğlul, İ, Optimization in Renewable Energy Systems, Chapter 2 - Introduction to Optimization, "https://doi.org/10.1016/B978-0-08-101041-9.00002-8" "http://www.sciencedirect.com/science/article/pii/B9780081010419000028"

[GKD] Glover, F., Kochenberger, G., Du, Y., A Tutorial on Formulating and Using QUBO models, 2019 https://arxiv.org/pdf/1811.11538.pdf

[MLO] Miguel, S., Lizaso, E., Orus, R., Use Cases of Quantum Optimization for Finance https://arxiv.org/abs/2010.01312
Figure 4: Histograms with increasing constraints

[SBM] Goto, H., Tatsumura, K., & Dixon, A.R. (2019) Combinatorial optimization by simulating adiabatic bifurcations in nonlinear Hamiltonian systems, Science Advances, 5(4), DOI:10.1126/sciadv.aav2372

[STB] Şeker O., Tanoumand N., Bodur M., Digital Annealer for quadratic unconstrained binary optimization: a comparative performance analysis. https://arxiv.org/abs/2012.12264

[Wiki] https://en.wikipedia.org/wiki/Quadratic_unconstrained_binary_optimization