Benchmarking Leviosa

Jesse Berwald

August 2023

1 Introduction

This document provides a foundational outlines of test problems, discussions of the importance of each problems (or type of problem) within the benchmark suite, and expected results when available.

2 Key definitions

Many of the problems we're interested in are optimization problems that are formulated as QUBO's. These are easily converted to Hamiltonians for ingestion by Leviosa's hardware using a simple linear transformation outlined below.

For posterity, I'll outline the general problem defintion for Qubos. A more complete summary and tutorial is available from Fred Glover and collaborators [**?**]. A QUBO is a quadratic unconstrained binary optimization and is used to represent combinatorial optimization problems. The Qubo models the problem as a symmetric matrix and the problems is typically expressed as a minimization:

Given a Qubo, Q

$$x^* = \min_x x^t Q x, \qquad (1)$$

where $x^* \in \{0,1\}^n$ represents the minimum solution vector that minimizes the above model. What makes this optimization problem special is the space over which is it solved. We can rewrite $x^t Q x$ as a real-valued polynomial, $Q(x) = \sum_{i=0}^{n} h_i x_i + \sum_{i=0}^{n} \sum_{j=1}^{n} J_i j x_i x_j$, where h_i and $J_{i,j}$ are both in $\mathbb{R}$, but $x_i \in \{0,1\}$. This yields a very restrictive solution space for which traditional continuous solvers often do not work, even with relaxation methods [**?**]. Due to the lack of continuity in the space, traditional methods used to find and prove minima are ineffective.

Nevertheless, many effective techniques have been developed to find local minima of a Qubo over the hypercube $\{0,1\}^n$. One such method is the Tabu search, developed by Fred Glover. The `QBSolv` package that we use to solve Qubos is one such implementation of a Tabu solver.

2.1 Technical notes

The conversion from a symmetric matrix to a polynomial is straightforward, but a couple subtleties should be noted. First, for every $q_{i,j} \in Q$, where $i \neq j$, $q_{i,j} = q_{j,i} = J_{i,j}/2$. Also, note that $x_i = x_i^2$.

Due to it's square symmetry, every Qubo can be view as a weighted graph adjacency matrix. The diagonal terms need to be dealt with, either as node weights or even self loops, depending on the situation.

There is a simple way to convert *classical* Ising Hamiltonians to/from Qubos with a change of variables. If a Hamiltonian has variables $s_i \in \{-1, 1\}$, then an equivalent Qubo with variables $x_i \in \{0, 1\}$ can be represented by the substituting $x_i \frac{1+s_i}{2}$.

3 Parameters

In general, benchmarking and device characterization studies need to define the parameters over which they will test a device (or devices). We will delineate our studies over a few parameter spaces, and list them all here. In each case study below we will define the parameters used and set appropriate bounds for each one

4 Random Qubos

Often the most basic type of problem to test is the random Qubo. The parameter space to explore here is

- number of variables
- dynamic range of the coefficients in the Qubo
- density of the Qubo matrix
- pertubation of coefficients and sensitivity to solutions

4.1 Perturbations and degenerate solutions

One way to test the accuracy of a device is to slowly perturb a parameter to determine where the device can "see" the changes. This is often termed "sensitivity". Take a Hamiltonian (which we'll convert to a Qubo below),

$$H(\alpha) = s_1^2 - 6z1 + s_2^2 - 6s_2 + (6 - \alpha)s_1 s_2. \tag{2}$$

This problem is a good example to show sensitivity to a parameter. By varying α we can see the precision of a device. Using QBsolv's `sample_ising` we can solve the problem directly. First, setting $\alpha = 0$, we see that the problem has three degenerate solutions:

$$[1, 1], [-1, 1], [1, -1], \tag{3}$$

α	Num QBsolv solutions	QBSolv energy	Num Ham solver solutions	Ham solver energy
0	3	-8.0	na	na
0.01	1	-8.01	na	na

Probing a device around α is an easy way to define a *sensitivity metric*. For $\alpha > 0$, (2) has a single solution. The key is to understand and record the smallest value that a device will switch over from three to one solution. In this case, the expected energy will be $H(0) - \alpha$.