Solving quadratic optimization problems with bosonic quantum computer simulator

Report I

Bálint Hantos
supervisor: Péter Rakyta

Motivation

The field of quantum computing is a promising subject because of the powerful applications in solving hard optimization problems.

Quadratic uncostrained binary optimization (QUBO)[4]

In many applications we face optimization problems, where one has to fit parameters of a given model. Mostly, those problems are about finding continous parameters, for which there are powerful tools eg. gradient descent algorithm. However, this requires convexity which means the
$f : \mathbb{R}^m \to \mathbb{R}$ objective function to be twice continously differentiable.

In our case, what we're trying to do is to find an optimal set of parameters from a finite set of discrete values. This implies that the domain of the objective function is not continous, therefore convexity is not satisfied. In most of the problems brute-force search is not feasible due to the large number of potential variations. Instead, there is a large amount of literature on polynomial-time algorithms for certain classes of such problems, which either approximate the optimal solution or rule out a part of the search space.

However there are still a number of combinatorial optimization problems, which have no determenistic polinomial-time algorithm yet. One of these problems is *quadratic unconstrained binary optimization* (QUBO).

Finding the optimal solution to a QUBO is equivalent to minimizing the H Hamilton operator of the Ising-model. It has a wide range of applications apart from physics, for example in binary portfolio optimization, or various other problems in finance and mathematics.

We can formulate it as finding the minimum of a quadratic binary mapping $f_Q(x)$ of a fixed $m \times m$ symmetric Q matrix $f_Q(x) : \{0,1\}^M \to \mathbb{R}$ in the form of

$$f_Q(x) = \frac{1}{2}\langle\, x|\, Q\, |x\,\rangle = \frac{1}{2}\sum_{i,j=1}^{M} Q_{ij}x_i x_j. \tag{1}$$

The goal is to find $x^* = \text{argmin}(f(x))$ for the given $Q\,\epsilon\,\mathbb{R}^{m\times m}$ symmetric matrix.

If there is a linear term it can be incorporated into the matrix $\tilde{Q}$:

$$\begin{aligned} f_Q(x) &= \sum_{i,j=1}^{M} Q_{ij}x_i x_j + \sum_{i=1}^{M} b_i x_i = \\ &= \sum_{i,j=1}^{M} Q_{ij}x_i x_j + \sum_{i=1}^{M} b_i x_i x_i = \\ &= \sum_{i=1}^{M}(Q_{ii} + b_i)x_i x_i + \sum_{\substack{i,j=1 \\ i\neq j}}^{M} Q_{i,j}x_i x_j = \\ &= \sum_{i=1}^{M} \tilde{Q}_{ii}x_i x_i + \sum_{\substack{i,j=1 \\ i\neq j}}^{M} Q_{i,j}x_i x_j = \\ &= \sum_{i,j=1}^{M} \tilde{Q}_{ij}x_i x_j, \end{aligned} \tag{2}$$

where Q_{ii} is the m-dimensional vector made up from the main diagonal of Q. This result can be reached because it makes no difference if b_i is multiplied by x_i or $x_i x_i$, since $x_i\,\epsilon\,\{0,1\}$. So, the Q matrix is transformed in the way

$$\tilde{Q}_{ij} = \begin{cases} Q_{ii} + b_i & \text{if} \quad i = j \\ Q_{ij} & \text{if} \quad i \neq j. \end{cases} \tag{3}$$

In the worst case we can find the optimal $|x\rangle$ of a QUBO by brute-force search. For a $m \times m$ Q matrix, the size of the binary vector $|x\rangle$ is m. So the number of potential variations for $|x\rangle$ is 2^m.

Introduction and theoretical background

Boson sampling[1]

Boson sampling shows an example of a classically computationally expensive mathematical problem, which can be efficiently solved by a realizable physical experimental setup. The boson sampler consists of a linear optics network and a coincidence photodetector. The linear optic part consists of beam splitters and phase shifter, characterized by angles ϑ_i and ψ_i for each piece of equipment accordingly.

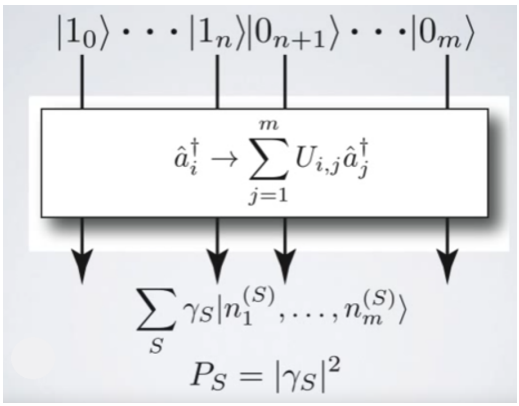

Figure 1: Schema of a boson sampler.
On top, the single photon states and vacuum states, propagated through a linear optics network U. At the bottom the superpositions of the multiphoton optical states. γ_S is the amplitude of the configuration S (instead of S we use $|\mathbf{n}\rangle$ in this document), P_S is the probability of measuring the configuration S, which we try to reconstruct with repeated photon coincidence measurements. Image credits to [1].

The boson sampler is an M-mode linear interferometer, where the inputs are single photons, and the outputs are multi photon states. The input photons are indistinguishable from each other, so we are not concerned about which input photon end up in which output state, we consider only the output configuration

$$|\xi(\theta,\psi)\rangle = \sum_{|\mathbf{n}\rangle} \alpha_{|\mathbf{n}\rangle} |n_1, n_2, ...n_M\rangle. \tag{4}$$

This output superposition of multiple states is measured so $|\xi\rangle$ collapses to a single $|\mathbf{n}\rangle = |n_1, n_2, ..., n_M\rangle$ multiphoton Fock-vector with probability $\alpha^2_{|\mathbf{n}\rangle}$, where n_i are the number of photons in the i state.

Simulating such an equipment on a classical computer, that samples bosons in this way is pretty hard because the probabilities (and amplitudes) associated with the optical states involves calculating the permanent of a complex-valued matrix, which is #P-complete [2]:

$$P(|\mathbf{n}\rangle) = \frac{|\mathrm{Per}(A_{|\mathbf{n}\rangle})|^2}{n_1! \cdot n_2! \cdot ... \cdot n_M!}, \tag{5}$$

where $A_{|\mathbf{n}\rangle}$ is a unitary matrix describing both the unitary matrices of the linear optics network configuration and the $|\mathbf{n}\rangle$ output configuration. The n_i in the denominator are the number of photons in state i for the given configuration $|\mathbf{n}\rangle$.

Variational bosonic solver[3]

Most problems in physics and other fields of study have no exact solutions. The variational approach is a way of obtaining a numerical solution to such problems. It approaches the exact solution (if there is any) as close as it is needed. We generally think of energy as an objective function, and the goal is to minimize it by finding the optimal values for the free parameters.

In order to interpret the measurement in a qubit basis the Fock-vector is mapped onto a M component $|\mathbf{b^{(j)}}\rangle$ binary vector with the $\wp_j$ parity function:

$$\wp_j : b_i^{(j)} = (n_i \bmod 2) \oplus j. \tag{6}$$

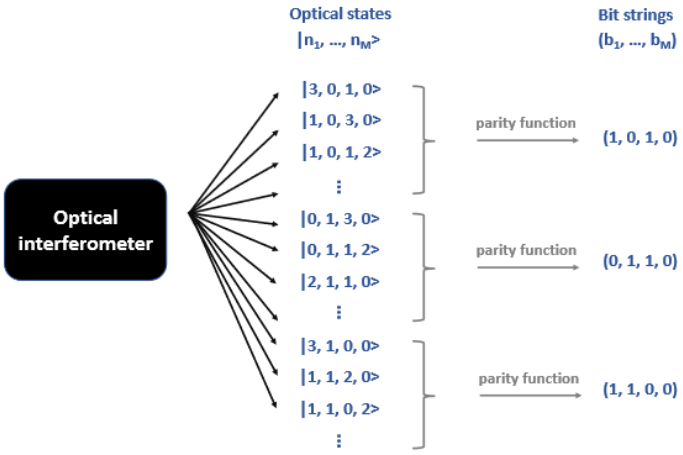

Figure 2: Examples of mapping multiple optical states the same bitstring. Image credits to [3].

Note that the different optical states can be mapped to the same bit strings, in that case their probabilities are summed. So if a $|\mathbf{n}\rangle$ and $|\mathbf{n}'\rangle$ with amplitudes α and α' are mapped to the same binary vector $|\mathbf{b^{(j)}}\rangle$ then the associated probability is $\beta = \alpha^2 + \alpha'^2$.

The $|\mathbf{b^{(j)}}\rangle$ bitstring has a corresponding amplitude $\beta_{|\mathbf{b^{(j)}}\rangle}$. Then the expected value of the E energy is calculated in the following way:

$$E_{|\mathbf{b^{(j)}}\rangle}(j; \vartheta, \psi) = \sum_{|\mathbf{b^{(j)}}\rangle} \beta_{|\mathbf{b^{(j)}}\rangle} \langle \mathbf{b^{(j)}}| \, H \, |\mathbf{b^{(j)}}\rangle. \tag{7}$$

The variational bosonic solver algorithm tries to find the E_{min} minimal energy and the corresponding configuration $|\mathbf{b}_{min}\rangle$. In each epoch the gradient descent step is made in the direction to the minimal energy value by adjusting ϑ_i and ψ_i angles of the beam splitters and phase-shifters.

Setup

Technical details

We're going to use the Picasso and Picasso Boost libraries as a bosonic quantum simulator, which are written in python and C++. In order to install them we needed to install a couple of dependencies on a Linux machine. For this task I decided to use a Docker Container built from an Ubuntu image, upon which I installed the Miniconda distribution of Anaconda to spare some disk space. With the help of my supervisor we installed the necessary packages and built the Piquasso Boost library. Then we launched a prewritten test of the package which turned out well, thus the installation was succesful.

Examining the code

After that I checked how the variational QUBO-solver worked with some examples. I used a general 3-by-3 symmetric matrix to calculate its expected values with all possible binary binary vectors

$$\mathbf{Q} = \begin{pmatrix} A & B & C \\ B & D & E \\ C & E & F \end{pmatrix}. \tag{8}$$

(1,0,0)	A
(1,1,0)	A + 2B + D
(1,1,1)	A + D + F + 2(B+ C + E)
(1,0,1)	A+2C+F
(0,1,1)	D+2E+F
(0,0,1)	F
(0,1,0)	D
(0,0,0)	0

Table 1: Expected value of **Q** with the possible binary vectors.

I found that the implementation of the variational algorithm works as intended. Although I found some corner cases, for example when the minimal expected values are degenerate then the optimal binary vector is ambiguous. In this case, the solver returns only one answer after multiple runs.

QUBO application: Break minimization in Mirrored Double Round-robin Tournament

The ultimate value of sports lies within fairness and sportsmanship. It is encouraged not only individually but also on a team level. It is more favorable to play on home ground eg. the home team travels less before the game and possibly they don't have to get used to a different climate. Also there is the psychological effect of fans on players.

There are measures that are put in place to balance the advantages or disadvantages of playing home or away. For instance the away team gets to hit first or in some sports the away team's score is doubled given the two teams play both at home and away. It is also possible, that they play in a neutral venue.

Another measure to ensure fairness among teams is *break minimization*. We call a *break* when a team plays two consecutive games home or away.

Mirrored Double Round-robin Tournament (MDRRT)

Firstly, we're going to discuss the details of the MDRRT competition as it is described in [5].

1. Each team meets every other team twice.
2. Each team has its own venue in its home town.
3. Each game is played at the home of either team.
4. Given a pair of teams, if the first game takes place in one team's venue, the second game is played on the other team's venue.

5. The first half of the tournament is in the same order as the second half, but the games are played in the opposite venue.

Symbols

Firstly, we define the symbols, as used in [5].

- $2n$: number of teams. $n \geq 2$ is an integer.
- $T \, \epsilon \, \{1, 2, ..., 2n\}$: a set of teams.
- $S \, \epsilon \, \{1, 2, ..., 2(2n-1)\}$: a set of slots.
- $\tau(t, s) \, \epsilon \, T \times S$: the opponent that plays against team t at slot s.
- $a(t, s)$: $a(t, s) = 1$ if a team plays at home and $a(t, s) = 0$ if a team plays away.
- $\mathscr{T}$: timetable with entries $\tau(t, s)$.
- $\mathscr{A}$: home-away assignment table with entries $a(t, s)$.

Secondly, we show an example on the tables defining one tournament. The timetable shows the order in which teams are going to meet each other. The home-away assignment table shows whether a team plays the game home or away.

Table 2: Timetable and home-away assignment table

Slot	**1**	**2**	**3**	**4**	**5**	**6**
team 1	2	3	4	2	3	4
team 2	1	4	3	1	4	3
team 3	4	1	2	4	1	2
team 4	3	2	1	3	2	1

Slot	**1**	**2**	**3**	**4**	**5**	**6**
team 1	1	0	1	0	1	0
team 2	0	0	1	1	1	0
team 3	1	1	0	0	0	1
team 4	0	1	0	1	0	1

To create a tournament first the timetable is created without deciding the venue. Then each game is assigned to either team's venue. We are dealing with this second step on how to optimize this assignment in a way that the number of breaks is reduced.

Formulating break minimization

Formulating the break minimization we introduce $\mathscr{K}(k)$ representing the kth combination of pair of teams and their slots, $k \,\epsilon\, \{1, 2, ..., n(2n-1)\}$, given $2n$ teams.

$$\mathscr{K}(k) \;=\; \{(t_k, s_k),\, (t_k, s'_k),\, (t'_k, s_k),\, (t'_k, s'_k)\} \tag{9}$$

[5] describes two ways of formulating this problem, one with constraints and another without constraints. We describe only the latter with some comment on the first, because we aim to solve using tools special for the unconstrained case.

The unconstrained formulation uses $\{y_{t_k s_k}, y_{t_k s'_k}, y_{t'_k s_k}, y_{t'_k s'_k}\}$ sets of binary variables for a given pair of teams. $y_{ts} = 1$ if t team plays at home and in slot s and $y_{ts} = 0$ if at home. There are three constraints for y_{ts}:

- Each team plays at its home or at its opponent's home: $y_{t_k s_k} + y_{t'_k s_k} = 1$.
- A team plays at home (or away) against its opponent, then the second game is played at away (or home): $y_{t_k s_k} + y_{t_k s'_k} = 1$.
- The first game is played at home (or away) and the second game is played at home (or away) of the opponent: $-y_{t_k s_k} + y_{t'_k s'_k} = 0$.

We can formalize the problem in a way that the constraints are inherently satisfied by introducing $z_k := y_{t_k s_k}$:

$$y_{t_k s_k} = z_k, \quad y_{t_k s'_k} = 1 - z_k, \quad y_{t'_k s_k} = 1 - z_k, \quad y_{t'_k s'_k} = z_k. \tag{10}$$

Important to mention that $z_k \,\epsilon \{0, 1\}^k$

The objective function $f(\mathbf{z})$ of four terms, within those two parts, one representing two consecutive home games and two consecutive away games:

$$\begin{aligned} f(\mathbf{z}) = \sum_{t_k \epsilon T} \sum_{s_k \epsilon S \setminus \{4n-2\}} & a(t_k, s_k)(z_k z'_k + (1-z_k)(1-z'_k)) \\ & + b(t_k, s_k)((1-z_k)z_{k'} + z_k(1-z_{k'})) \\ & + c(t_k, s_k)(z_k(1-z_{k'}) + (1-z_k)z_{k'}) \\ & + d(t_k, s_k)(z_k z_{k'} + (1-z_k)(1-z_{k'})). \end{aligned} \tag{11}$$

In equation 11 k' satisfies $(t_k, s_{k+1}) \epsilon \mathscr{K}(k)$ for any k. This is the QUBO formulation, which enables us to use quantum annealing to solve this problem.

Example of break minimization

We walk through an example of creating a QUBO matrix from a time table and finding the minimal break home-away assignment z_k vectors. This time we put less emphasis on using the boson sampler simulator and more on understanding what we deal with.

Table 3: Home away assignment table schema according to the unconstrained formalism

Slot	**1**	**2**	**3**	**4**	**5**	**6**
team 1	z_1	z_2	z_3	$1-z_1$	$1-z_2$	$1-z_3$
team 2	$1-z_1$	z_4	z_5	z_1	$1-z_4$	$1-z_5$
team 3	z_6	$1-z_2$	$1-z_5$	$1-z_6$	z_2	z_5
team 4	$1-z_6$	$1-z_4$	$1-z_3$	z_6	z_4	z_3

Table 4: Terms of the objective function. Columns: terms grouped by team

$4z_1z_2 - 2z_1 - 2z_2$	$-4z_1z_4 + 2z_1 + z_4$	$-4z_2z_6 + 2z_2 + 2z_6$	$4z_4z_6 - 2z_4 - 2z_6$
$4z_2z_3 - 2z_2 - 2z_3$	$4z_4z_5 - 2z_4 - 2z_5$	$4z_2z_5 - 2z_2 - 2z_5$	$4z_3z_4 - 2z_3 - 2z_4$
$-2z_1z_3 + z_1 + z_3$	$2z_1z_5 - z_1 - z_5$	$2z_5z_6 - z_5 - z_6$	$-2z_3z_6 + z_3 + z_6$

If we group the coefficients for the terms, we get the QUBO matrix $\mathbf{Q}$:

$$\mathbf{Q} = \begin{bmatrix} 0 & 4 & -2 & -4 & 2 & 0 \\ 4 & -4 & 4 & 0 & 4 & -4 \\ -2 & 4 & -2 & 4 & 0 & -2 \\ -4 & 0 & 4 & -4 & 4 & 4 \\ 2 & 4 & 0 & 4 & -6 & 2 \\ 0 & -4 & -2 & 4 & 2 & 0 \end{bmatrix} \tag{12}$$

We calculated the expected value $\langle \mathbf{z}|\mathbf{Q}|\mathbf{z}\rangle$ for this matrix for every $z_k \epsilon \{0,1\}^k$ and counted the breaks associated with them. This is the point where we emphasize our contribution for writing a break calculator, which returns the number of breaks given a timetable and a home-away assignment vector.

Unfortunately we found no clear correspondence between the expected value of the QUBO matrix and the number of breaks as it can be seen on Figure 3. This leads to the conclusion that perhaps the $\mathbf{Q}$ matrix is calculated in a wrong way. In the future we might try using different timetables with the same calculations, however we presume that it is not very likely to generate better results.

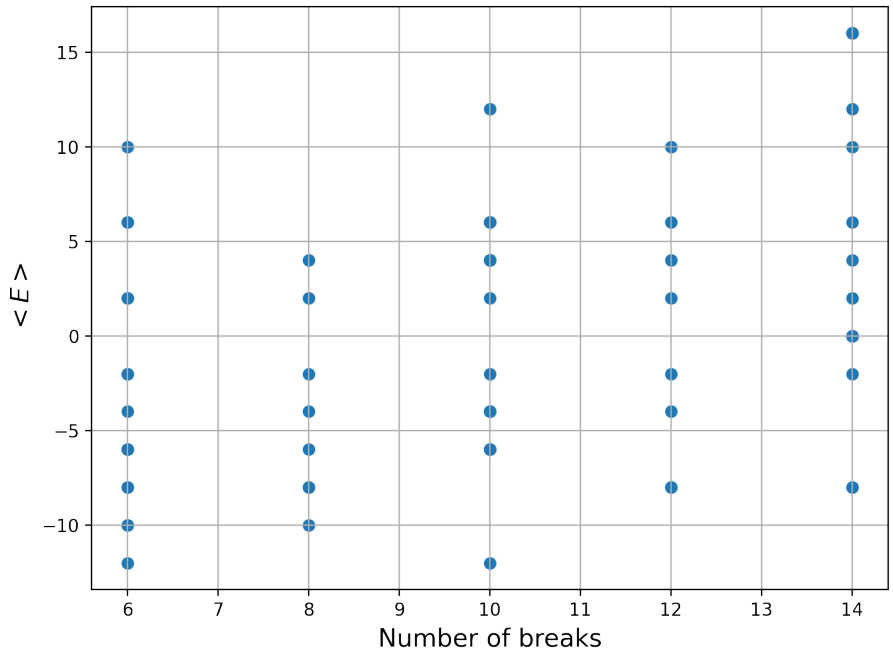

Figure 3: Number of breaks plotted against the expected value of the QUBO matrix. We can see that there is little to no correspondence between energy and the number of breaks. It is important to note that there are multiple overlapping points in the graph.

Corrections

After thorough investigation we found that although the algorithms written for the HA-assignment and break counting are correct, the z_i components were mismatched during the calculations of the **Q** QUBO-matrix.
The previous tables and quantities after correction are the following:

$$\mathbf{Q} = \begin{bmatrix} 0 & 4 & -2 & 2 & -4 & 0 \\ 4 & -4 & 4 & 4 & 0 & -4 \\ -2 & 4 & -2 & 0 & 4 & -2 \\ 2 & 4 & 0 & -6 & 4 & 2 \\ -4 & 0 & 4 & 4 & -4 & 4 \\ 0 & -4 & -2 & 2 & 4 & 0 \end{bmatrix} \tag{13}$$

Table 5: Home away assignment table schema according to the unconstrained formalism

Slot	**1**	**2**	**3**	**4**	**5**	**6**
team 1	z_1	z_2	z_3	$1-z_1$	$1-z_2$	$1-z_3$
team 2	$1-z_1$	z_5	z_4	z_1	$1-z_5$	$1-z_4$
team 3	z_6	$1-z_2$	$1-z_4$	$1-z_6$	z_2	z_4
team 4	$1-z_6$	$1-z_5$	$1-z_3$	z_6	z_5	z_3

Table 6: Corrected terms of the objective function.
Columns: terms grouped by team

$4z_1z_2 - 2z_1 - 2z_2$	$-4z_1z_5 + 2z_1 + z_5$	$-4z_2z_6 + 2z_2 + 2z_6$	$4z_5z_6 - 2z_5 - 2z_6$
$4z_2z_3 - 2z_2 - 2z_3$	$4z_4z_5 - 2z_4 - 2z_5$	$4z_2z_4 - 2z_2 - 2z_4$	$4z_3z_5 - 2z_3 - 2z_5$
$-2z_1z_3 + z_1 + z_3$	$2z_1z_4 - z_1 - z_4$	$2z_4z_6 - z_4 - z_6$	$-2z_3z_6 + z_3 + z_6$

There is a slighter correction that needs to be added due to overlapping data points on Figure 4. Important to note that the lowest energy data point is associated with a minimum-break HA-assignment, thus quantum annealing can be a viable option for finding it.

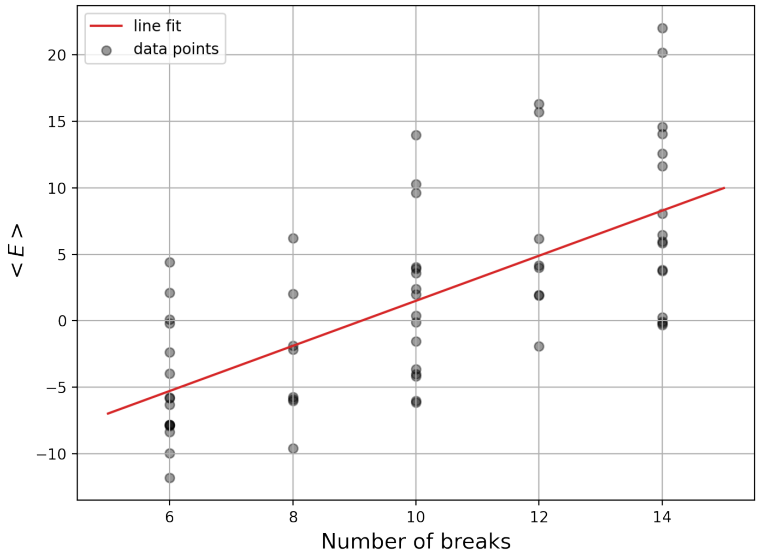

Figure 4: Number of breaks plotted against the expected value of the QUBO matrix. The correspondence between lower energies and breaks can be confirmed visually, hence the fitted line. Overlapping data points are darker.

References

[1] Gard, Bryan T., et al. "An introduction to boson-sampling." *From atomic to mesoscale: The role of quantum coherence in systems of various complexities.* 2015

[2] Aaronson, Scott, and Alex Arkhipov. "The computational complexity of linear optics." Proceedings of the forty-third annual ACM symposium on Theory of computing. 2011. https://arxiv.org/abs/1011.3245

[3] Bradler, Kamil, and Hugo Wallner. "Certain properties and applications of shallow bosonic circuits." arXiv preprint arXiv:2112.09766 (2021). https://arxiv.org/abs/2112.09766

[4] Quadratic unconstrained binary optimization - Wikipedia, 2022 [Accessed 18 February 2022]

[5] Kuramata, Michiya, Ryota Katsuki, and Kazuhide Nakata. "Solving Large Break Minimization Problems in a Mirrored Double Round-robin Tournament Using Quantum Annealing." arXiv preprint arXiv:2110.07239 (2021). https://arxiv.org/pdf/2110.07239.pdf