



Problem Formulation Guide

WHITEPAPER

2022-01-10

Overview

This guide steps users through the process of developing a model suitable for the D-Wave quantum processing unit (QPU) or hybrid solvers. By reading through this guide and working through the provided exercises, a new user with little to no prior experience with D-Wave technology will be able to model small problems using the binary quadratic model (BQM).

CONTACT

Corporate Headquarters
3033 Beta Ave
Burnaby, BC V5G 4M9
Canada
Tel. 604-630-1428

US Office
2650 E Bayshore Rd
Palo Alto, CA 94303

Email: info@dwavesys.com

www.dwavesys.com



Notice and Disclaimer

D-Wave Systems Inc. (“D-Wave”) reserves its intellectual property rights in and to this document, any documents referenced herein, and its proprietary technology, including copyright, trademark rights, industrial design rights, and patent rights. D-Wave trademarks used herein include D-WAVE®, Leap™ quantum cloud service, Ocean™, Advantage™ quantum system, D-Wave 2000Q™, D-Wave 2X™, and the D-Wave logo (the “D-Wave Marks”). Other marks used in this document are the property of their respective owners. D-Wave does not grant any license, assignment, or other grant of interest in or to the copyright of this document or any referenced documents, the D-Wave Marks, any other marks used in this document, or any other intellectual property rights used or referred to herein, except as D-Wave may expressly provide in a written agreement.

Contents

| | | |
|-------|--|----|
| 1 | Introduction | 1 |
| 1.1 | Example 1: Traveling Salesman Problem | 1 |
| 1.2 | Example 2: Antenna Placement Problem | 1 |
| 1.3 | Exercise | 2 |
| 1.4 | Resources | 2 |
| 2 | Definitions | 2 |
| 2.1 | Binary Quadratic Model | 2 |
| 2.2 | Quadratic Unconstrained Binary Optimization | 3 |
| 2.2.1 | Example | 3 |
| 2.2.2 | Exercise | 3 |
| 2.3 | Ising Model | 4 |
| 2.3.1 | Example | 4 |
| 2.3.2 | Exercise | 4 |
| 2.4 | Equivalence of QUBO and Ising | 4 |
| 2.4.1 | QUBO to Ising | 5 |
| 2.4.2 | Ising to QUBO | 5 |
| 2.4.3 | Exercise | 5 |
| 2.5 | Summary of Key Points | 5 |
| 3 | Small Problems with Two Variables | 6 |
| 3.1 | General Process | 6 |
| 3.2 | Example | 6 |
| 3.2.1 | Step 1: Constraint Satisfaction Table | 6 |
| 3.2.2 | Step 2: Make a System of Equations | 7 |
| 3.2.3 | Step 3: Solve the System of Equations | 8 |
| 3.3 | Exercise | 8 |
| 4 | Problems with Multiple Constraints | 9 |
| 4.1 | Social Networks (Friends and Enemies) | 9 |
| 4.1.1 | Problem Setup | 9 |
| 4.1.2 | Combining Relationships | 10 |
| 4.2 | BQM Formulation | 11 |
| 4.2.1 | Friend Relationship | 11 |
| 4.2.2 | Enemy Relationship | 11 |
| 4.2.3 | Combining All Relationships | 13 |
| 4.3 | BQM Development Process | 14 |
| 4.3.1 | Step 1: Write Objective and Constraints | 14 |
| 4.3.2 | Step 2: Convert Objective and Constraints into Binary Math Expressions | 15 |
| 4.3.3 | Step 3: Transform Math Expressions into a BQM | 15 |
| 4.3.4 | Step 4: Combine Expressions | 16 |
| 4.4 | Example: Choosing Boxes | 17 |
| 4.4.1 | Step 1: Write Objective and Constraints | 17 |
| 4.4.2 | Step 2: Convert Objective and Constraints into Binary Math Expressions | 17 |
| 4.4.3 | Step 3: Transform Math Expressions into a BQM | 18 |
| 4.4.4 | Step 4: Combine Expressions | 19 |
| 4.4.5 | Exercise | 19 |



| | | |
|-------|--|----|
| 5 | Summary of Exercises with Solutions | 19 |
| 5.1 | Introduction | 20 |
| 5.1.1 | Solution | 20 |
| 5.2 | Definitions | 20 |
| 5.2.1 | QUBO | 20 |
| 5.2.2 | Ising | 20 |
| 5.2.3 | Converting Between QUBO and Ising | 21 |
| 5.3 | Small Problems with Two Variables | 21 |
| 5.3.1 | QUBO | 21 |
| 5.3.2 | Ising | 21 |
| 5.4 | Problems with Multiple Constraints | 21 |
| 5.4.1 | QUBO | 21 |
| 5.4.2 | Ising | 22 |
| | References | 22 |

1 Introduction

When working with the D-Wave systems and software, we need to formulate our problem as a binary quadratic model, or BQM. We can formulate many classes of problems as BQMs, such as optimization problems. We can view an optimization problem in terms of objectives and constraints. An objective is something that we want to minimize or maximize, while constraints are rules about which solutions are acceptable. As we explore how to formulate our problems as binary quadratic models, we will use this terminology frequently.

Below are two examples of optimization problems that can be formulated as BQMs and run using the D-Wave QPU or our hybrid solvers. These examples provided demonstrate how real world problems can be broken down into objectives and constraints. While these examples are more complex than those that we will develop in this guide, the same strategy and techniques can help to build a BQM for each of them. For the interested reader we provide links to where more information on the BQM formulation for each example can be found.

1.1 Example 1: Traveling Salesman Problem

A well-known problem in computer science is the traveling salesman problem (TSP). In this problem, we have a list of cities that must be visited (exactly once) on a sales person's route. The sales person would like to take the shortest route possible. We can phrase this problem in terms of objectives and constraints.

Objective:

Minimize the total distance for the route.

Constraint:

Visit each city exactly once.

For more information on how to formulate this problem as a BQM, check out Andrew Lucas's paper on Ising formulations [1].

1.2 Example 2: Antenna Placement Problem

We have a set of 10 antennas that we can place anywhere in a list of 100 locations. We want to cover as much geographic area as possible. However, some locations interfere with each other. Where should we build our antennas?

Objective:

Maximize coverage.

Constraints:

- Build at most 10 antennas.
- Choose locations that do not cause interference.

For more information on how to formulate this problem as a BQM, take a look at the antenna selection example in the Collection of Code Examples [2].

1.3 Exercise

How can we split the following problem up into an objective and constraints?

A business is trying to figure out the best way to pack a box full of items. There are five items, with values 1, 5, 3, 4, and 2 dollars. These items have weights 2, 4, 4, 1, and 3 lbs, respectively. The contents of the box must weigh exactly 6 lbs. What is the best way to pack the box so that the box has the highest total value?

1.4 Resources

Many other resources are available for new users getting with D-Wave. Below is a selection of some that may be especially useful.

Ocean SDK Documentation: <https://docs.ocean.dwavesys.com/>

Collection of Code Examples: <https://github.com/dwave-examples>

D-Wave on YouTube: <https://www.youtube.com/dwavesystems>

2 Definitions

2.1 Binary Quadratic Model

The binary quadratic model (BQM) class of problems consists of Ising models and quadratic unconstrained binary optimization (QUBO) models. These two forms are mathematically equivalent, and our software tools allow you to work in either domain. In this guide, we will provide examples and exercises in both forms. Note that some problems may see better results with Ising over QUBO, and vice versa.

A BQM takes the following form.

$$\min \left(\sum_i a_i v_i + \sum_i \sum_{j>i} b_{i,j} v_i v_j + c \right)$$

The coefficients a_i and $b_{i,j}$ are constant numbers we choose to define our problem, as is the constant term c . The binary variables v_i and v_j are the values that we are looking for to solve our problem. The best solution for these variables is the value for each v_i that produces the smallest value for the overall expression. Searching for the variables that minimize an expression is called an “argmin” in mathematics.

Let’s break this expression down.

Linear Terms. The first summation, $\sum a_i v_i$, contains linear terms, with each having just one binary variable.

Quadratic Terms. The second summation, $\sum \sum b_{i,j} v_i v_j$, contains quadratic terms, with each term in the summation containing a product of two variables.

Constants. In the general BQM form we may or may not include constants. Since we are looking for an argmin, any constant terms will not affect our final answer. However, it may be useful when interpreting the output from D-Wave solvers and samplers.

2.2 Quadratic Unconstrained Binary Optimization

A QUBO model takes the following form.

$$\min \left(\sum_i a_i x_i + \sum_i \sum_{j>i} b_{i,j} x_i x_j + c \right)$$

The binary variables x_i and x_j may take values from $\{0, 1\}$.

When we are ready to run a QUBO with the D-Wave Ocean software tool kit, one method is to store the QUBO in a matrix format for the Ocean SDK. We can map a QUBO to a matrix by placing linear terms along the diagonal and quadratic terms in the off-diagonal positions.

2.2.1 Example

We can map $x_1 + 2x_2 + 3x_3 + 12x_1x_2 + 13x_1x_3 + 14x_2x_4$ to the following matrix.

$$\begin{bmatrix} 1 & 12 & 13 & 0 \\ 0 & 2 & 0 & 14 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In this matrix, we place coefficient a_i in matrix position (i, i) and coefficient $b_{i,j}$ in matrix position (i, j) .

2.2.2 Exercise

Put the following QUBO into matrix form.

$$x_1 + x_1x_2 - 3x_3 + 5$$

2.3 Ising Model

An Ising model takes the following form.

$$\min \left(\sum_i h_i s_i + \sum_i \sum_{j>i} J_{i,j} s_i s_j \right)$$

The binary variables s_i and s_j may take values from $\{-1, +1\}$.

When it comes time to write Ocean programs, we will need to represent our Ising model as a matrix and a vector. We can map the quadratic terms of an Ising model to a matrix by placing them in the off-diagonal positions, and the linear terms to a vector as a list of coefficients as shown below.

2.3.1 Example

We can map $s_1 + 2s_2 + 3s_3 + 12s_1s_2 + 13s_1s_3 + 14s_2s_4$ to the following list and matrix.

Linear.

$$h = [1 \quad 2 \quad 3 \quad 0]$$

Quadratic.

$$J = \begin{bmatrix} 0 & 12 & 13 & 0 \\ 0 & 0 & 0 & 14 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In this matrix, coefficient $J_{i,j}$ in matrix position (i, j) .

2.3.2 Exercise

Put the following Ising model into matrix form.

$$s_1 + s_1s_2 - 3s_3 + 5$$

2.4 Equivalence of QUBO and Ising

The Ising and QUBO models are mathematically equivalent — we can translate from one model to the other using a simple substitution.

2.4.1 QUBO to Ising

To translate a QUBO model to an Ising model, we use the following substitution.

$$x_i \mapsto \frac{s_i + 1}{2}$$

Example.

$$\begin{aligned} f(x) &= -22x_1 - 6x_2 - 14x_3 + 20x_1x_2 + 28x_1x_3 + 9 \\ &= -22\left(\frac{s_1+1}{2}\right) - 6\left(\frac{s_2+1}{2}\right) - 14\left(\frac{s_3+1}{2}\right) \\ &\quad + 20\left(\frac{s_1+1}{2}\right)\left(\frac{s_2+1}{2}\right) + 28\left(\frac{s_1+1}{2}\right)\left(\frac{s_3+1}{2}\right) + 9 \\ &= -11s_1 - 11 - 3s_2 - 3 - 7s_3 - 7 + 5s_1s_2 \\ &\quad + 5s_1 + 5s_2 + 5 + 7s_1s_3 + 7s_1 + 7s_3 + 7 + 9 \\ &= s_1 + 2s_2 + 5s_1s_2 + 7s_1s_3 \end{aligned}$$

2.4.2 Ising to QUBO

To translate an Ising model to a QUBO model, we use the following substitution.

$$s_i \mapsto 2x_i - 1$$

Example.

$$\begin{aligned} g(s) &= s_1 + 2s_2 + 5s_1s_2 + 7s_1s_3 \\ &= (2x_1 - 1) + 2(2x_2 - 1) + 5(2x_1 - 1)(2x_2 - 1) + 7(2x_1 - 1)(2x_3 - 1) \\ &= 2x_1 - 1 + 4x_2 - 2 + 20x_1x_2 - 10x_1 - 10x_2 + 5 + 28x_1x_3 - 14x_1 - 14x_3 + 7 \\ &= -22x_1 - 6x_2 - 14x_3 + 20x_1x_2 + 28x_1x_3 + 9 \end{aligned}$$

2.4.3 Exercise

Convert the following QUBO model to an Ising model.

$$x_1 + x_1x_2 - 3x_3 + 5$$

Convert the following Ising model to a QUBO model.

$$s_1 + s_1s_2 - 3s_3 + 5$$

2.5 Summary of Key Points

- A BQM can contain constant, linear, and quadratic terms, and no higher-order terms.
- We define the coefficients and constant in the expression.
- The solution is the value of each binary variable that minimizes the expression.

3 Small Problems with Two Variables

3.1 General Process

The following steps can be used to create a two-variable BQM.

1. Make a constraint satisfaction table.
2. Create a system of equations.
3. Solve the system of equations.

The constraint satisfaction table is similar to a truth table - it covers all possible values for our two variables and considers whether or not the constraint that we are modeling is true or false for each choice of values.

3.2 Example

We start with a simple problem. We have two variables, say v_1 and v_2 , and we want a BQM that represents the constraint $v_1 = v_2$. For QUBO and Ising we follow the same process.

3.2.1 Step 1: Constraint Satisfaction Table

Set up a table for the two variables that covers all possible values they might take. Add a column in the table to indicate whether or not the constraint is satisfied (true) for each combination of variable values.

| QUBO | | Ising | | |
|-------|-------|-------|-------|-------|
| x_1 | x_2 | s_1 | s_2 | Equal |
| 0 | 0 | -1 | -1 | Yes |
| 0 | 1 | -1 | +1 | No |
| 1 | 0 | +1 | -1 | No |
| 1 | 1 | +1 | +1 | Yes |

Next, replace the final column of yes/no values with small/large values, respectively.

| QUBO | | Ising | | |
|-------|-------|-------|-------|-------|
| x_1 | x_2 | s_1 | s_2 | Value |
| 0 | 0 | -1 | -1 | 0 |
| 0 | 1 | -1 | +1 | 1 |
| 1 | 0 | +1 | -1 | 1 |
| 1 | 1 | +1 | +1 | 0 |

In this step, we choose a small value when the constraint is satisfied because the QPU is designed to minimize a BQM. By choosing a small value, we line up the ground state (lowest energy the QPU can find) with the constraint being satisfied. Note that any choice of small and large value can be used.

3.2.2 Step 2: Make a System of Equations

Recall from the last section the general form for QUBO and Ising.

QUBO: $a_1x_1 + a_2x_2 + b_{1,2}x_1x_2 + c$

Ising: $h_1s_1 + h_2s_2 + J_{1,2}s_1s_2 + c$

Using each row in the constraint satisfaction table, we can write an equation. Each pair of variable values, the general BQM model, and the final column value together determine an equation in which the BQM coefficients are the only unknowns. This results in four different equations (one for each row in the table).

| QUBO |
|---|
| $a_1 \cdot 0 + a_2 \cdot 0 + b_{1,2} \cdot 0 \cdot 0 + c = 0$ |
| $a_1 \cdot 0 + a_2 \cdot 1 + b_{1,2} \cdot 0 \cdot 1 + c = 1$ |
| $a_1 \cdot 1 + a_2 \cdot 0 + b_{1,2} \cdot 1 \cdot 0 + c = 1$ |
| $a_1 \cdot 1 + a_2 \cdot 1 + b_{1,2} \cdot 1 \cdot 1 + c = 0$ |

| Ising |
|---|
| $h_1 \cdot (-1) + h_2 \cdot (-1) + J_{1,2} \cdot (-1) \cdot (-1) + c = 0$ |
| $h_1 \cdot (-1) + h_2 \cdot (+1) + J_{1,2} \cdot (-1) \cdot (+1) + c = 1$ |
| $h_1 \cdot (+1) + h_2 \cdot (-1) + J_{1,2} \cdot (+1) \cdot (-1) + c = 1$ |
| $h_1 \cdot (+1) + h_2 \cdot (+1) + J_{1,2} \cdot (+1) \cdot (+1) + c = 0$ |

3.2.3 Step 3: Solve the System of Equations

Solve the system of equations for your coefficients.

| QUBO | Ising |
|----------------|------------------|
| $a_1 = 1$ | $h_1 = 0$ |
| $a_2 = 1$ | $h_2 = 0$ |
| $b_{1,2} = -2$ | $J_{1,2} = -0.5$ |
| $c = 0$ | $c = 0.5$ |

To help with solving these systems of equations there are many resources online that are available. Some of these are WolframAlpha (wolframalpha.com) or Matrix Calculator (matrixcalc.org).

Our final models are:

QUBO: $x_1 + x_2 - 2x_1x_2$

Ising: $-0.5s_1s_2 + 0.5$

3.3 Exercise

What do the QUBO and Ising models look like for the constraint $v_1 \geq v_2$?

4 Problems with Multiple Constraints

When we work with problems in the real world, we often have an objective and multiple constraints playing a role simultaneously. How do we incorporate all of these components into a single BQM?

4.1 Social Networks (Friends and Enemies)

We work through the “Social Networks” problem, sometimes referred to as “Friends and Enemies”. For more examples that discuss this problem and some applications take a look at our demo (<https://cloud.dwavesys.com/leap/resources#demos>) and Jupyter Notebook (<https://cloud.dwavesys.com/leap/learning/>) available on the Leap platform (<https://cloud.dwavesys.com/leap/>).

4.1.1 Problem Setup

In a group of people, we have some existing relationships. Some pairs of people are friends, others are enemies, and some have no relationship (maybe they’ve never met each other before).

For example, let’s say we have the following friend and enemy relationships.

Friends:

- Katie and Anthony
- Carlos and Betty

Enemies:

- Anthony and Carlos

Suppose that we are asked to split this group of people into two sets. Maybe they’re being divided into two teams, or being split into groups for a project. Our goal is to split the people into two groups so that we have friends in the same group and enemies in different groups. How can we map this to a BQM?

The first step is to think about how we can use binary variables to represent the answer to our problem. When we consider our problem, the answer that we’re looking for is which group each person should be assigned to. If we call these groups “A” and “B”, then an answer should look something like:

- Katie in Group A
- Anthony in Group A
- Carlos in Group B
- Betty in Group B

To do this we can assign a binary variable to each person that tells us which group they belong to. If we're working with a QUBO model we could say $x_K = 0$ means Katie is in Group A, while $x_K = 1$ means Katie is in Group B. Similarly, if we're working with an Ising model we can say $s_K = -1$ means Katie is in Group A, while $s_K = 1$ means Katie is in Group B.

| | Group A | Group B |
|-------|---------|---------|
| QUBO | 0 | 1 |
| Ising | -1 | +1 |

Now that we have figured out how to align our answers with binary valued variables, we need to figure out how to write our minimization function. A good way to tackle this problem is to think about individual relationships instead of the entire set of people as a whole. For two friends with a given relationship, which team assignments are good and which are bad?

| Katie | Anthony | Good/Bad? |
|-------|---------|-----------|
| A | A | Good |
| A | B | Bad |
| B | A | Bad |
| B | B | Good |

This table should look familiar now — it's similar to what we did for our two variable relationships in the beginning of this guide! In fact, this is exactly the same as the QUBO/Ising truth tables we used to create a BQM that represents the constraint $v_1 = v_2$.

4.1.2 Combining Relationships

At this point we are able to write a BQM for each relationship. How can we combine them into one BQM? The answer is surprisingly simple: addition.

Once we have a BQM for every relationship, we can add all of the BQMs together to make a single, comprehensive BQM for the entire population.

4.2 BQM Formulation

4.2.1 Friend Relationship

For our friend relationships, we saw in a previous section (Small Problems with Two Variables) that we can represent this relationship ($x_1 = x_2$) by the following BQMs.

QUBO: $x_1 + x_2 - 2x_1x_2$

Ising: $-0.5s_1s_2 + 0.5$

This gives us the following BQMs for our friend relationships.

| | QUBO | Ising |
|--------------------------|-----------------------|--------------------|
| Katie and Anthony | $x_A + x_K - 2x_Ax_K$ | $-0.5s_As_K + 0.5$ |
| Carlos and Betty | $x_B + x_C - 2x_Bx_C$ | $-0.5s_Bs_C + 0.5$ |

4.2.2 Enemy Relationship

For our enemy relationship, we need to figure out what BQM models the relationship. We do this in the same way we did our two-variable BQMs in the earlier section.

Constraint Satisfaction Table:

| QUBO | | Ising | | |
|-------|-------|-------|-------|------------|
| x_1 | x_2 | s_1 | s_2 | Different? |
| 0 | 0 | -1 | -1 | No |
| 0 | 1 | -1 | +1 | Yes |
| 1 | 0 | +1 | -1 | Yes |
| 1 | 1 | +1 | +1 | No |

Replacing the final column of yes/no values with small/large values.

| QUBO | | Ising | | |
|-------|-------|-------|-------|------------|
| x_1 | x_2 | s_1 | s_2 | Different? |
| 0 | 0 | -1 | -1 | 1 |
| 0 | 1 | -1 | +1 | 0 |
| 1 | 0 | +1 | -1 | 0 |
| 1 | 1 | +1 | +1 | 1 |

Building our system of equations.

| QUBO |
|---|
| $a_1 \cdot 0 + a_2 \cdot 0 + b_{1,2} \cdot 0 \cdot 0 + c = 1$ |
| $a_1 \cdot 0 + a_2 \cdot 1 + b_{1,2} \cdot 0 \cdot 1 + c = 0$ |
| $a_1 \cdot 1 + a_2 \cdot 0 + b_{1,2} \cdot 1 \cdot 0 + c = 0$ |
| $a_1 \cdot 1 + a_2 \cdot 1 + b_{1,2} \cdot 1 \cdot 1 + c = 1$ |

| Ising |
|---|
| $h_1 \cdot (-1) + h_2 \cdot (-1) + J_{1,2} \cdot (-1) \cdot (-1) + c = 1$ |
| $h_1 \cdot (-1) + h_2 \cdot (+1) + J_{1,2} \cdot (-1) \cdot (+1) + c = 0$ |
| $h_1 \cdot (+1) + h_2 \cdot (-1) + J_{1,2} \cdot (+1) \cdot (-1) + c = 0$ |
| $h_1 \cdot (+1) + h_2 \cdot (+1) + J_{1,2} \cdot (+1) \cdot (+1) + c = 1$ |

Solving our system of equations.

| QUBO | Ising |
|---------------|-----------------|
| $a_1 = -1$ | $h_1 = 0$ |
| $a_2 = -1$ | $h_2 = 0$ |
| $b_{1,2} = 2$ | $J_{1,2} = 0.5$ |
| $c = 1$ | $c = 0.5$ |

Our final models are:

QUBO: $-x_1 - x_2 + 2x_1x_2 + 1$

Ising: $0.5s_1s_2 + 0.5$

For the pair of enemies in our set of people, we now have:

| | QUBO | Ising |
|---------------------------|----------------------------|--------------------|
| Anthony and Carlos | $-x_A - x_C + 2x_Ax_C + 1$ | $0.5s_{ASC} + 0.5$ |

4.2.3 Combining All Relationships

Now that we have a BQM for each relationship, we add them together to get our BQM for the entire problem.

QUBO:

$$(x_A + x_K - 2x_Ax_K) + (x_B + x_C - 2x_Bx_C) + (-x_A - x_C + 2x_Ax_C + 1)$$

This simplifies down to the following.

$$x_B + x_K + 2x_Ax_C - 2x_Ax_K - 2x_Bx_C + 1$$

Ising:

$$(-0.5s_{ASK} + 0.5) + (-0.5s_{BSC} + 0.5) + (0.5s_{ASC} + 0.5)$$

This simplifies down to the following.

$$-0.5s_{ASK} - 0.5s_{BSC} + 0.5s_{ASC} + 1.5$$

To double-check our work, we can evaluate our BQM values for all possible assignments for these four people. The rows with the smallest value tell us the best assignment of people to teams.

| Anthony | Betty | Carlos | Katie | Value |
|---------|-------|--------|-------|-------|
| A | A | A | A | 1 |
| A | A | A | B | 2 |
| A | A | B | A | 1 |
| A | A | B | B | 2 |
| A | B | A | A | 2 |
| A | B | A | B | 3 |
| A | B | B | A | 0 |
| A | B | B | B | 1 |
| B | A | A | A | 1 |
| B | A | A | B | 0 |
| B | A | B | A | 3 |
| B | A | B | B | 2 |
| B | B | A | A | 2 |
| B | B | A | B | 1 |
| B | B | B | A | 2 |
| B | B | B | B | 1 |

4.3 BQM Development Process

To successfully map problems to BQMs, we work through a series of steps.

4.3.1 Step 1: Write Objective and Constraints

This should be written out in terms of your problem domain (i.e. not using math!).

The **objective** of a problem is what we are looking to minimize or maximize. Look for those key words in your problem to determine if your problem has one, or if you are just looking to satisfy a set of constraints.

A **constraint** in a problem is a rule that we have to follow. An answer that doesn't satisfy a given constraint is called **infeasible** — it's not a good answer, and generally we won't be able to use it.

For example, the previous problem might be described with the following:

Objective: None

Constraint 1: Friends on the same team

Constraint 2: Enemies on different teams

Note that there may be more than one way to interpret your problem as objectives and constraints.

4.3.2 Step 2: Convert Objective and Constraints into Binary Math Expressions

The first step is to think about binary variables. What answer are you seeking to your problem? What “Yes” or “No” questions can you ask that will give you that answer? Once we think of our problem in these terms, we can assign a binary variable for each question.

Next, we transform our objective and constraints into math expressions using these binary variables. Often this can be done with our truth table approach if can find a way to break our problem down into two- or three-variable relationships.

4.3.3 Step 3: Transform Math Expressions into a BQM

Different types of expressions will require different strategies. If you used the truth table approach in step 2, you may not need to make any adjustments!

Some common transformations are the following.

Squared Terms. Our QUBO and Ising models don’t have room for squared binary variables. It turns out that we don’t actually need them.

QUBO: Our binary variables in a QUBO are 0 and 1. Note that $0^2 = 0$ and $1^2 = 1$, so we can replace any term x_i^2 with x_i .

Ising: Our binary variables in an Ising model are -1 and +1. Note that $(-1)^2 = 1$ and $(+1)^2 = 1$, so we can replace any term s_i^2 with the constant 1.

Maximization to Minimization. If our objective function is a maximization function (for example, you might be maximizing profit), we can convert this to a minimization by multiplying the entire expression by -1.

Example:

$$\arg \max(3v_1 + 2v_1v_2) = \arg \min(-3v_1 - 2v_1v_2)$$

Equality to Minimization. If you have a constraint that is an equality, we can convert it to a minimization expression by moving all arguments and constants to one side of the equality and squaring the non-zero side of the equation. This will leave us with an expression that is satisfied at its smallest value (0) and unsatisfied at any larger value (>0).

Example:

Original constraint:

$$v_1 + v_2 = 1$$

Move all arguments and constants to one side:

$$v_1 + v_2 - 1 = 0$$

Square the expression:

$$(v_1 + v_2 - 1)^2$$

Note that the smallest value a squared expression can equal is zero, so the smallest value $(v_1 + v_2 - 1)^2$ can equal is zero. This expression will equal zero when $v_1 + v_2 - 1 = 0$, which is exactly the constraint that we are looking to satisfy.

4.3.4 Step 4: Combine Expressions

Once we have written all of the components (objective and constraints) as BQM expressions, we make our final BQM by adding all of the components together.

In this step, we have the choice to weight the components if we choose. This weighting is done using a **Lagrange parameter**. Generally, we will assign individual Lagrange parameters to each constraint to prioritize valid solutions (ones in which all of our constraints are satisfied).

$$\text{BQM} = \min(\text{objective} + \gamma(\text{constraint}))$$

Example:

Suppose that we have two companies that can complete the same job, each for a different price. Company 1 can complete the job for \$12, while Company 2 can complete the job for \$8. We want to have the job completed for the smallest cost.

We can model this problem as a QUBO. The objective is to minimize cost and the constraint is that we must choose exactly one company. The binary variables x_1 and x_2 can be used to indicate whether we will hire Company 1 or Company 2 to complete the job. Using these binary variables, we write the objective as $12x_1 + 8x_2$ and the constraint as $(x_1 + x_2 - 1)^2$. These can be combined to make the following QUBO.

$$\min \left((12x_1 + 8x_2) + \gamma(x_1 + x_2 - 1)^2 \right)$$

If we choose $\gamma = 1$, we see the following values for our QUBO.

| x_1 | x_2 | Value |
|-------|-------|-------|
| 0 | 0 | 1 |
| 0 | 1 | 8 |
| 1 | 0 | 12 |
| 1 | 1 | 21 |

Note that the smallest value of 1 corresponds to not choosing either company to complete the job! Clearly this is not the optimal solution to our original problem, so we need to adjust the Lagrange parameter to ensure that our constraint (choose exactly one company) is satisfied.

Setting a larger Lagrange parameter will enforce our constraint. The following table demonstrates the QUBO values for $\gamma = 20$.

| x_1 | x_2 | Value |
|-------|-------|-------|
| 0 | 0 | 20 |
| 0 | 1 | 8 |
| 1 | 0 | 12 |
| 1 | 1 | 40 |

Now we see that the smallest value corresponds to the optimal solution of hiring Company 2 to complete the job.

Note that you may need to try a few different values to identify the best Lagrange parameter value for your specific BQM. A good starting value is to set γ equal to your best estimate of the value of your objective function. If you find that your constraints are not satisfied in the solutions returned, you may need to increase γ . On the other hand, if your constraints are all satisfied but your solutions are not close to optimal, you may need to decrease γ .

4.4 Example: Choosing Boxes

We're given three boxes with different weights. We want to choose the two boxes with the smallest sum.

| | Box 1 | Box 2 | Box 3 |
|--------|-------|-------|-------|
| Weight | 15 | 20 | 25 |

This is a simple problem – we know that the answer is to choose boxes 15 and 20. But how do we phrase this problem as a BQM?

4.4.1 Step 1: Write Objective and Constraints

Objective: We are looking for the smallest sum, so our objective is “minimize the sum of the boxes chosen”.

Constraint: We are allowed to choose two boxes, so our constraint is “choose exactly two boxes”.

4.4.2 Step 2: Convert Objective and Constraints into Binary Math Expressions

Binary Variables. First, we need to define our binary variables. The answer that we are looking for is which boxes we should choose. For each box, we can ask “do we choose this box?”. This points us to how we should define our binary variables.

| | Use Box i | Don't use Box i |
|--------------|-------------|-------------------|
| QUBO | $x_i = 1$ | $x_i = 0$ |
| Ising | $s_i = +1$ | $s_i = -1$ |

Once we have defined our binary variables, we can convert our objective and constraint into math expressions.

Objective. We consider our objective for both QUBO and Ising forms.

QUBO:

To figure out the sum of the boxes that are chosen, we can use a weighted sum: $15x_1 + 20x_2 + 25x_3$. In this sum, the boxes that are chosen will have $x_i = 1$ and the boxes that are not chosen will have $x_i = 0$. In other words, the value of boxes that are not chosen will be multiplied by zero and so we will only be adding up the value of the boxes that are chosen. Our objective function becomes:

$$\min 15x_1 + 20x_2 + 25x_3$$

Ising:

Using our binary variables, we can convert our $+1/-1$ to $1/0$ using the Ising to QUBO translation shown earlier, which maps $+1 \mapsto 1$ and $-1 \mapsto 0$. Our objective function can then be written as:

$$\min \left(15 \left(\frac{s_1 + 1}{2} \right) + 20 \left(\frac{s_2 + 1}{2} \right) + 25 \left(\frac{s_3 + 1}{2} \right) \right)$$

Constraint. We consider our constraint for both QUBO and Ising forms.

QUBO:

Our constraint “choose exactly two boxes” means that we need exactly two of our binary variables to have value 1, and the remaining binary variable will have value 0. In other words, our constraint can be written as:

$$x_1 + x_2 + x_3 = 2$$

Ising:

Our constraint “choose exactly two boxes” means that we need exactly two of our binary variables to have value $+1$, and the remaining binary variable will have value -1 . In other words, our constraint can be written as:

$$s_1 + s_2 + s_3 = 1$$

4.4.3 Step 3: Transform Math Expressions into a BQM

Our objective function is fine as written, so we only need to modify our constraint.

QUBO:

To modify our constraint, we need to use the method for equalities.

Original constraint:

$$x_1 + x_2 + x_3 = 2$$

Move everything to one side:

$$x_1 + x_2 + x_3 - 2 = 0$$

Square the expression:

$$(x_1 + x_2 + x_3 - 2)^2$$

Ising:

Following the same method, we can rewrite our constraint as follows.

$$(s_1 + s_2 + s_3 - 1)^2$$

4.4.4 Step 4: Combine Expressions

Now that we have written our objective and constraint in BQM form, we can combine them together to make our final model using addition and adding in a Lagrange parameter. Both of these can be expanded and simplified to prepare for input to an Ocean python program.

QUBO:

$$\min \left((15x_1 + 20x_2 + 25x_3) + \gamma(x_1 + x_2 + x_3 - 2)^2 \right)$$

Ising:

$$\min \left(15 \left(\frac{s_1 + 1}{2} \right) + 20 \left(\frac{s_2 + 1}{2} \right) + 25 \left(\frac{s_3 + 1}{2} \right) + \gamma(s_1 + s_2 + s_3 - 1)^2 \right)$$

4.4.5 Exercise

Write a BQM for the following problem.

A business is trying to figure out the best way to pack a box full of items. There are five items, with values 1, 5, 3, 4, and 2 dollars. These items have weights 2, 4, 4, 1, and 3 lbs, respectively. The contents of the box must weigh exactly 6 lbs. What is the best way to pack the box so that the box has the highest total value?

5 Summary of Exercises with Solutions

Below you will find solutions to the exercises included in this “Learn to Formulate Problems” guide. For further discussion of these solutions, please visit our community forum in Leap (<https://support.dwavesys.com/hc/en-us/community/topics>).

5.1 Introduction

Split the following problem into an objective and constraints.

A business is trying to figure out the best way to pack a box full of items. There are five items, with values 1, 5, 3, 4, and 2 dollars. These items have weights 2, 4, 4, 1, and 3 lbs, respectively. The contents of the box must weigh exactly 6 lbs. What is the best way to pack the box so that the box has the highest total value?

5.1.1 Solution

This problem can be split into an objective and constraints as follows.

Objective: Maximize total value

Constraint: Total weight of contents is exactly 6 lbs

5.2 Definitions

5.2.1 QUBO

Put the following QUBO into matrix form.

$$x_1 + x_1x_2 - 3x_3 + 5$$

Solution:

$$\begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -3 \end{bmatrix}$$

5.2.2 Ising

Put the following Ising model into matrix form.

$$s_1 + s_1s_2 - 3s_3 + 5$$

Solution:

Linear:

$$h = [1 \quad 0 \quad -3]$$

Quadratic

$$J = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

5.2.3 Converting Between QUBO and Ising

QUBO to Ising:

$$\frac{3}{4}s_1 + \frac{1}{4}s_2 - \frac{3}{2}s_3 + \frac{1}{4}s_1s_2 + \frac{17}{4}$$

Ising to QUBO:

$$-2x_2 - 6x_3 + 4x_1x_2 + 8$$

5.3 Small Problems with Two Variables

5.3.1 QUBO

What is the QUBO model for the constraint $x_1 \geq x_2$?

Solution:

$$x_2 - x_1x_2$$

5.3.2 Ising

What is the Ising model for the constraint $s_1 \geq s_2$?

Solution:

$$-\frac{1}{4}s_1 + \frac{1}{4}s_2 - \frac{1}{4}s_1s_2 + \frac{1}{4}$$

5.4 Problems with Multiple Constraints

Write a BQM for the following problem.

A business is trying to figure out the best way to pack a box full of items. There are five items, with values 1, 5, 3, 4, and 2 dollars. These items have weights 2, 4, 4, 1, and 3 lbs, respectively. The contents of the box must weigh exactly 6 lbs. What is the best way to pack the box so that the box has the highest total value?

5.4.1 QUBO

Solution:

$$\min \left((-x_1 - 5x_2 - 3x_3 - 4x_4 - 2x_5) + \gamma(2x_1 + 4x_2 + 4x_3 + x_4 + 3x_5 - 6)^2 \right)$$

5.4.2 Ising

Solution:

$$\min \left(\left(-\frac{1}{2}s_1 - \frac{5}{2}s_2 - \frac{3}{2}s_3 - 2s_4 + s_5 - \frac{15}{2} \right) + \gamma \left(s_1 + 2s_2 + 2s_3 + \frac{1}{2}s_4 + \frac{3}{2}s_5 + 1 \right)^2 \right)$$

References

- ¹ A. Lucas, "Ising formulations of many np problems," *Frontiers in Physics* **2**, 5 (2014).
- ² *Collection of code examples*, <https://github.com/dwave-examples> (2020).