Chapter 6

New QUBO Formulations

In this chapter, we initially study two problems on a D-Wave Quantum Computer: the Bounded-Depth Steiner Tree problem (a variant of the Steiner Tree Problem) for edge-weighted graphs and the Chromatic Sum problem. Both of them are NP-hard optimization problems. We solve each problem in a procedure similar to the one described in the previous chapter (i.e., reduce each problem into their QUBO form so that it can be solved by quantum annealer). To this end, we propose a QUBO formulation for each problem. Then, the correctness of the formulations is verified by a classical QUBO solver before any instance is solved on a D-Wave computer. An analysis of the computational results is presented at last. We also adapt the proposed method for several closely related problems such as the Bounded-depth Minimum Spanning Tree problem and the Diameter-Constrained Steiner Tree Problem.

6.1 Bounded-Depth Steiner Tree

The minimum spanning tree is a subtree of a weighted graph $G(V, E)$ with a minimum total cost that spans all vertices. Steiner tree problem specifies a set $U \subseteq G$ named terminal vertices that are to be connected by the subtree while the other vertices, called Steiner vertices, may or may not be contained in the subtree. We again add a depth bound to this problem which makes it a slight variant of Steiner Tree problem: the Bounded-Depth Steiner Tree problem.

Definition 21. *Considering an undirected weighted graph $G = (V, E)$ with nonnegative weights $c_{u,v}$ associated with each edge $(u, v) \in E$. Given a set $U \subset V$ of terminal vertices with vertex $v_0 \in U$ specified as the root vertex and an nature number h, the **Bounded-Depth Steiner Trees** problem is to find a minimum cost subtree T of G such that there exists a path in T from v_0 to each vertex in terminal set U and the distance not exceeding h.*

As the Steiner tree problem arises in telecommunications networks design, the study of bounded-depth Steiner tree becomes relevant when a maximum bound on transmission delay in such a case [31]. More relevant real-world optimization problems such as lot-sizing problem have been studied using this model [67, 5].

6.1.1 QUBO formulation

In this subsection, we present our QUBO formulation which requires $\mathcal{O}(h|V|^2)$ binary variables in worst case and produces a QUBO matrix with a density of $\mathcal{O}(h|V|^2)$.
The variables involved in this formulation are:

- $x_{v_0,u,1}$ for each edge $(v_0,u) \in E$.
- $x_{u,v,i}$ and $x_{v,u,i}$ for each edge $(u,v) \in E$ with $u,v \neq v_0$ and $2 \leq i \leq h$

If $x_{u,v,i} = 1$, it means that edge (u,v) is contained in a directed optimal solution tree such that u is closer to the root than v and vertex v is in depth i. As v_0 is specified as root, we assume that the arcs of the arborescence are directed away from the v_0. Therefore, we use a single variable $x_{v_0,u,1}$ to represent each edge $v_0, u \in E$. In total, we need $2(h-1)(|E| - \deg_G(v_0)) + \deg_G(v_0)$ binary variables. In the worst case, $|E|$ is $\mathcal{O}(|V|^2)$, the complexity is then $\mathcal{O}(h|V|^2)$ in terms of variable size.
The objective function $F(\mathbf{x})$ to be minimized is stated as follows.

$$F(\mathbf{x}) = O(\mathbf{x}) + A \cdot P(\mathbf{x}) \tag{6.1}$$

where

$$P(\mathbf{x}) = |V| \cdot (P_1(\mathbf{x}) + P_2(\mathbf{x})) + P_3(\mathbf{x})$$

with

$$P_1(\mathbf{x}) = \sum_{v \in U \setminus \{v_0\}} \left(1 - \sum_{(u,v)\in E} \sum_{i=1}^{h} x_{u,v,i}\right)^2 \tag{6.2}$$

$$P_2(\mathbf{x}) = \sum_{v \in V \setminus U} \sum_{i=1}^{h} \left(\sum_{(u,v),(w,v)\in E} x_{u,v,i} x_{w,v,i} \right) \tag{6.3}$$

$$P_3(\mathbf{x}) = \sum_{v \in V \setminus \{v_0\}} \left(\sum_{(u,v)\in E} \sum_{i=2}^{h} x_{u,v,i} \left(1 - \sum_{(w,u)\in E} x_{w,u,i-1}\right)\right)$$

$$O(\mathbf{x}) = \sum_{(u,v)\in E} \sum_{i=2}^{h} c_{u,v} x_{u,v,i} + \sum_{(v_0,u)\in E} c_{v_0,u} x_{v_0,u,1}$$

and

$$A = E_T \max_{(u,v)\in E} (c_{u,v}) + 1$$

The terms are interpreted as follows.

- $P_1(\mathbf{x}) = 0$ when every terminal vertex excluding the root, that is $v \in U \backslash \{v_0\}$, has exactly one incoming arc and one assigned depth. Otherwise, $P_1(\mathbf{x})$ adds a positive penalty to objective function.
- $P_2(\mathbf{x}) = 0$ when every Steiner vertex $v \in V \backslash U$ has at most one incoming arc and at most one assigned depth. Otherwise, $P_2(\mathbf{x})$ adds a positive penalty to objective function.
- $P_3(\mathbf{x})$, on the other hand, penalize with a positive number if there exists an arc (u, v) in solution tree such that u is not parent of v. We also observe that $\sum_{(w,u)\in E} x_{w,u,i-1}$, which represents $\deg^-(u)$ in T, may exceed 1 and make $P_3(\mathbf{x})$ negative. However, we claim that the combined sum $|V| \cdot (P_1(\mathbf{x}) + P_2(\mathbf{x})) + P_3(\mathbf{x})$ is positive.
- The term $O(\mathbf{x})$ sums up the cost of each edge that is present in an optimal solution.
- A penalty scalar A is finally applied to $P(\mathbf{x})$ that is essential to make this formulation work correctly.

Once the optimal solution of the generated QUBO is obtained, $F(\mathbf{x}^*)$ gives the minimum cost and $\mathbf{x}^*$ encodes a corresponding minimum cost Steiner tree satisfying the depth constraint. Note that optimal solution may not exist for some instances of this problem. The scalar A can be used as a cut-off integer to decide whether a minimized value of the QUBO encodes a optimal solution or not. We will show that in due course. Observe that a variable assignment $\mathbf{x} \in \mathbb{Z}_2^{2(h-1)(|E|-\deg_G(v_0))+\deg_G(v_0)}$ for the QUBO encodes a directed subgraph $S_\mathbf{x} = (V_\mathbf{x}, A_\mathbf{x}) \subseteq G$:

$$V_\mathbf{x} = \bigcup_{i=0}^{h} V_{\mathbf{x},i}$$
$$A_\mathbf{x} = \bigcup_{i=1}^{h} A_{\mathbf{x},i}$$

where

$$V_{\mathbf{x},i} = \{v_0\} \cup \{v \in V \mid \exists u \in V \backslash \{v\} \text{ such that } x_{u,v,i} = 1 \text{ or } x_{v,u,i} = 1\}$$
$$A_{\mathbf{x},i} = \{(u, v) \mid x_{u,v,i} = 1 \text{ or } x_{v,u,i} = 1\}$$

Lemma 2. *Let $\mathbf{x} \in \mathbb{Z}_2^{2(h-1)(|E|-\deg_G(v_0))+\deg_G(v_0)}$. Then $S_\mathbf{x}$ is a rooted Steiner tree for terminal vertices U in G with depth constraint h if and only if $P(\mathbf{x}) = 0$.*

Proof. Given a graph $G = (V, E)$, depth constraint h and v_0 specified as the root, we can represent a bounded-depth Steiner tree T for G as a sequence of vertex sets $\{V_0 = \{v_0\}, V_1, \ldots, V_h\} \subset P(V)$, which represents the vertices in depth i, and arc sets $A_1, A_2, \ldots, A_h$ such that for every $1 \leq i \leq h$:

1. Each arc in A_i is an oriented edge of E.
2. $V_i = \{v \mid (u,v) \in A_i, u \in V_{i-1}, v \in V_i\}$.
3. For any $(u,v) \in A_i, u \in V_{i-1}$ and $v \in V_i$.
4. A terminal vertex $v \in U$ only appears in one of the vertex sets.
5. A Steiner vertex $v \in V \backslash U$ appears in at most one of the vertex sets.
6. $U \subseteq (V_0 \cup V_2 \cup \cdots \cup V_h)$.

With a binary vector $\mathbf{x} \in \mathbb{Z}_2^{2(h-1)(|E|-\deg_G(v_0))+\deg_G(v_0)}$, we have

$$V_{\mathbf{x},i} = \{v \in V \mid x_{u,v,i} = 1\}, \text{ where } 1 \le i \le h$$
$$A_{\mathbf{x},i} = \{(u,v) \in E \mid x_{u,v,i} = 1\}, \text{ where } 1 \le i \le h$$

All the six criteria hold when $P_{\mathbf{x}} = 0$:

1. *Each arc in $A_{\mathbf{x},i}$ is an oriented edge of E.*
   This holds by the definition of variable $x_{u,v,i}$.
2. $V_{\mathbf{x},i} = \{v \mid (u,v) \in A_{\mathbf{x},i}, u \in V_{\mathbf{x},i-1}, v \in V_{\mathbf{x},i}\}$.
   Suppose $v \in V_{\mathbf{x},i}$, which means that $x_{u,v,i} = 1$. If $u = v_0$, then $u \in V_{\mathbf{x},i-1}$ holds by the definition, $v_0 \in V_0$. If $u \neq v_0$, suppose a contradiction $u \notin V_{\mathbf{x},i-1}$. In such case, $1 - \sum_{(u,v)\in E} x_{w,u,i-1} = 1$ which leads to $P_2(\mathbf{x}) > 0$. But $P_2(\mathbf{x}) = 0$, so $u \in V_{\mathbf{x},i-1}$.
3. *For any $(u,v) \in A_{\mathbf{x},i}, u \in V_{\mathbf{x},i-1}$ and $v \in V_{\mathbf{x},i}$.*
   If $(u,v) \in A_{\mathbf{x},i}$, then we have $x_{u,v,i} = 1$. Thus, $u \in V_{\mathbf{x},i-1}, v \in V_{\mathbf{x},i}$ holds as we illustrated in previous criteria.
4. *A terminal vertex $v \in U$ only appears in one of the vertex sets.*
   For a contradiction suppose that we have a terminal vertex $v \in U$ appears in more than one of the vertex sets or appears in none of the vertex sets but $P_1(\mathbf{x}) = 0$. That is, $\sum_i x_{u,v,i} > 2$ or $\sum_i x_{u,v,i} = 0$, which implies $P_1(\mathbf{x}) > 0$, a contradiction.
5. *A Steiner vertex $v \in V \backslash U$ appears in at most one of the vertex sets.*
   For a contradiction suppose that we have a Steiner vertex $v \in V \backslash U$ appears in more than one of the vertex sets but $P_2(\mathbf{x}) = 0$. That is, $\sum_i x_{u,v,i} > 2$, which implies $P_2(\mathbf{x}) > 0$, a contradiction.
6. $U \subseteq (V_0 \cup V_{\mathbf{x},1} \cup \cdots \cup V_{\mathbf{x},h})$.
   Suppose there exists a vertex v such that $v \in U$ and $v \notin (V_0 \cup V_{\mathbf{x},1} \cup \cdots \cup V_{\mathbf{x},h})$. This suggests $\sum_{(u,v)\in E} \sum_i x_{u,v,i} = 0$ which leads to $P_1(\mathbf{x}) > 0$.

As noted in the term $P_3(\mathbf{x})$, if $\deg^-(u) \geq 2$ for some $w \in V$, $P_3(\mathbf{x})$ becomes negative. Suppose for some vertex $v \in V \backslash \{v_0\}$, there exists a vertex u where $\deg^-(u) \geq 1$ and $(u, v) \in A_{\mathbf{x}}$. Let $\deg^-(u)$ incremented by 1. As a consequence,

$$P_3(\mathbf{x}) = \sum_{u \in V \backslash \{v_0\}} \deg^+(u)(1 - \deg^-(u))$$

is decreased by $\deg^+(u) < |V|$. On the other hand, if $v \in U$,

$$|V| P_1(\mathbf{x}) = |V| \sum_{u \in U \backslash \{v_0\}} \left(1 - \deg^-(u)\right)^2$$

will increase by $|V|$ at least. Otherwise,

$$|V| P_2(\mathbf{x}) = |V| \sum_{u \in V \backslash U} \binom{\deg^-(u)}{2}$$

will increase by at least $|V|$. Note that the third dimension variable i representing its depth is ignored in the calculation as this operation has not effect on the final result. Thus, the combined sum $P(\mathbf{x}) = |V|(P_1(\mathbf{x}) + P_2(\mathbf{x})) + P_3(\mathbf{x})$ is positive. □

Corollary 3. *Let $\mathbf{x} \in \mathbb{Z}_2^{2(h-1)(|E| - \deg_G(v_0)) + \deg_G(v_0)}$, then $F(\mathbf{x}) \geq A$ if and only if $S_{\mathbf{x}}$ is not an h-constrained Steiner tree of G.*

Proof. If $S_{\mathbf{x}}$ is not an h-constrained Steiner tree, then by Lemma 2, $P(\mathbf{x}) \neq 1$. This implies $P(\mathbf{x}) \geq 1$ since $P(\mathbf{x})$ is a non-negative integer. Therefore, $F(\mathbf{x}) = O(\mathbf{x}) + A \cdot P(\mathbf{x}) \geq A$. On the other hand, if $S_{\mathbf{x}}$ is a h-constrained Steiner tree of G, we have $P(\mathbf{x}) = 0$ and $F(\mathbf{x}) = O(\mathbf{x})$. Let $T = (V_T, E_T)$ be a Steiner tree, grounded on $c(u, v) \geq 0$ with $(u, v) \in E_T$ and $|V| - 1 = |E_T|$, we have:

$$F(\mathbf{x}) \leq |E_T| \cdot \max_{(u,v) \in E} (c_{u,v}) + 1 < A.$$

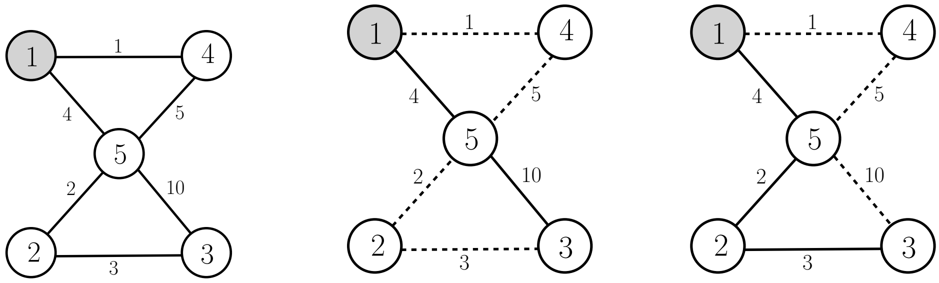
□

Theorem 4. *The QUBO formulation in (6.1) is correct.*

Proof. If a graph $G = (V, E)$ has a h-constrained minimum cost Steiner tree, there exists an assignment $\mathbf{x} \in \mathbb{Z}_2^{2(h-1)(|E| - \deg_G(v_0)) + \deg_G(v_0)} < A$ such that $S_{\mathbf{x}}$ is a Steiner tree. Assume that $\mathbf{x}^*$ is the optimal variable assignment of $F(\mathbf{x})$, then by Lemma 2,

$$F(\mathbf{x}^*) = \min O_{\mathbf{x}} = \min C(S_{\mathbf{x}}).$$

and $S_{\mathbf{x}^*}$ is a minimum cost Steiner tree of graph G. Therefore, (6.1) is a QUBO formulation of bounded-depth Steiner tree problem. □



$v_0 = 1, U = \{1, 3, 5\}$
(a) graph Butterfly

$h = 2$
(b) 2-constrained Steiner tree

$h = 3$
(c) 3-constrained Steiner tree

Figure 6.1: A weighted Butterfly and solutions with different bounded-depth

Table 6.1: QUBO matrix Q_1 for graph Butterfly with $h = 2$

variables	$x_{1,4,1}$	$x_{1,5,1}$	$x_{2,3,2}$	$x_{2,5,2}$	$x_{3,2,2}$	$x_{3,5,2}$	$x_{4,5,2}$	$x_{5,2,2}$	$x_{5,3,2}$	$x_{5,4,2}$
$x_{1,4,1}$	1	0	0	0	0	0	-20.5	0	0	205
$x_{1,5,1}$	0	-201	0	205	0	205	205	-20.5	-20.5	-20.5
$x_{2,3,2}$	0	0	-161	0	0	0	0	0	205	0
$x_{2,5,2}$	0	205	0	-162	0	205	205	0	0	0
$x_{3,2,2}$	0	0	0	0	44	0	0	102.5	0	0
$x_{3,5,2}$	0	205	0	205	0	-154	205	0	0	0
$x_{4,5,2}$	-20.5	205	0	205	0	205	-159	0	0	0
$x_{5,2,2}$	0	-20.5	0	0	102.5	0	0	43	0	0
$x_{5,3,2}$	0	-20.5	205	0	0	0	0	0	-154	0
$x_{5,4,2}$	-102.5	-20.5	0	0	0	0	0	0	0	46

6.1.2 Example: the graph Butterfly

Consider the weighted Butterfly graph shown in Figure 6.1a. The vertex set is $V = \{1, 2, 3, 4, 5\}$, the edge set is $E = \{(1, 4), (1, 5), (2, 3), (2, 5), (3, 5), (4, 5)\}$ and a cost for each edge: $c_{1,4} = 1, c_{1,5} = 4, c_{2,3} = 3, c_{2,5} = 2, c_{3,5} = 10, c_{4,5} = 5$. With $v_0 = 1$ specified as root and a terminal set $U = \{1, 3, 5\}$, we wish to find the minimum spanning tree with different bounded-depth 2 and 3.
Figure 6.1b and Figure 6.1c depict the bounded-depth Steiner tree of Butterfly for $h = 2$ and $h = 3$ respectively. Note that the vertex filled with gray color is root vertex and dashed-lines denote the edges that are not in solution tree.
The number of variables required for the two instances are $2(2-1)(6-2)+2 = 10$ and $2(3-1)(6-2)+2 = 18$. After processing the constants and linear terms, we obtain the diagonal symmetric matrix Q_1 and Q_2 shown in Table 6.1 and Table 6.2.

Table 6.2: QUBO matrix Q_2 for graph Butterfly with $h = 3$

variables	$x_{1,4,1}$	$x_{1,5,1}$	$x_{2,3,2}$	$x_{2,3,3}$	$x_{2,5,2}$	$x_{2,5,3}$	$x_{3,2,2}$	$x_{3,2,3}$	$x_{3,5,2}$	$x_{3,5,3}$	$x_{4,5,2}$	$x_{4,5,3}$	$x_{5,2,2}$	$x_{5,2,3}$	$x_{5,3,2}$	$x_{5,3,3}$	$x_{5,4,2}$	$x_{5,4,3}$
$x_{1,4,1}$	1	0	0	0	0	0	0	0	0	0	-20.5	0	0	0	0	0	102.5	102.5
$x_{1,5,1}$	0	-201	0	0	205	205	0	0	205	205	205	205	-20.5	0	-20.5	0	-20.5	0
$x_{2,3,2}$	0	0	-161	205	0	0	0	-20.5	0	-20.5	0	0	0	0	205	205	0	0
$x_{2,3,3}$	0	0	205	-161	0	0	-20.5	0	0	0	0	0	-20.5	0	205	205	0	0
$x_{2,5,2}$	0	205	0	0	-162	205	0	0	205	205	205	205	0	-20.5	0	-20.5	0	-20.5
$x_{2,5,3}$	0	205	0	0	205	-162	-20.5	0	205	205	205	205	-20.5	0	0	0	0	0
$x_{3,2,2}$	0	0	0	-20.5	0	-20.5	44	0	0	0	0	0	102.5	102.5	0	0	0	0
$x_{3,2,3}$	0	0	-20.5	0	0	0	0	44	0	0	0	0	102.5	102.5	-20.5	0	0	0
$x_{3,5,2}$	0	205	0	0	205	205	0	0	-154	205	205	205	0	-20.5	0	-20.5	0	-20.5
$x_{3,5,3}$	0	205	-20.5	0	205	205	0	0	205	-154	205	205	0	0	-20.5	0	0	0
$x_{4,5,2}$	-20.5	205	0	0	205	205	0	0	205	205	-159	205	0	-20.5	0	-20.5	0	-20.5
$x_{4,5,3}$	0	205	0	0	205	205	0	0	205	205	205	-159	0	0	0	0	-20.5	0
$x_{5,2,2}$	0	-20.5	0	-20.5	0	205	102.5	102.5	0	0	0	0	43	0	0	0	0	0
$x_{5,2,3}$	0	0	0	0	-20.5	0	102.5	102.5	-20.5	0	-20.5	0	0	43	0	0	0	0
$x_{5,3,2}$	0	-20.5	205	205	0	0	0	-20.5	0	-20.5	0	0	0	0	-154	205	0	0
$x_{5,3,3}$	0	0	205	205	-20.5	0	0	0	-20.5	0	-20.5	0	0	0	205	-154	0	0
$x_{5,4,2}$	102.5	-20.5	0	0	0	0	0	0	0	0	0	-20.5	0	0	0	0	46	0
$x_{5,4,3}$	102.5	0	0	0	-20.5	0	0	0	-20.5	0	-20.5	0	0	0	0	0	0	46

The optimal solutions for the two QUBO instances are:

$$\mathbf{x} = [0, 1, 0, 0, 0, 0, 0, 0, 1, 0],$$
$$\mathbf{x} = [0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0].$$

As expected, each of these two assignment gives the optimal solution $S_{\mathbf{x}} = (V_{\mathbf{x}}, E_{\mathbf{x}})$ where $E_{\mathbf{x}} = \{(1, 5), (5, 3)\}$ for $h = 2$ and $E_{\mathbf{x}} = \{(1, 5), (5, 2), (2, 3)\}$ for $h = 3$.

6.2 Bounded-Diameter Steiner Tree

In this section, we show our interest in the Bounded-Diameter Steiner Tree Problem which is closely related to Bounded-Depth Steiner Tree Problem. While a bounded-depth Steiner tree limits the distance between its vertex and a pre-specified root vertex, bounded-diameter requires that for every pair of vertices of the Steiner tree, the number of edges in the path cannot exceed the given diameter bound. Formally, it is defined as follows.

Definition 22. *Given a weighted Graph $G = (V, E)$, where V is a vertex set, E is an edge set, a cost function cost:$E \mapsto \mathbb{N}$ and an diameter bound $2 \leq d \leq n-2$, the **Bounded-Diameter Steiner Tree** problem is to find a spanning tree T such that it has the minimal total cost of edges and satisfies that for each pair of vertices $(u, v) \in T$, the number of edges in the path in T from u to v is not more than D.*

The problem is proven to be NP-hard if $d \geq 4$ [53]. Our QUBO formulation for the bounded-depth Steiner tree problem does not work directly for an instance of bounded-diameter Steiner tree problem. However, by iterating the vertex set V in a graph $G = (V, E)$ with each vertex assigned as the root, we can obtain a set $\Gamma(\mathbf{x}) = \{F_{v_0=1}(\mathbf{x}), F_{v_0=2}(\mathbf{x}), \ldots, F_{v_0=n}(\mathbf{x})\}$. Then, the minimum cost bounded-diameter Steiner tree of G with $d = 2$ is $\min \Gamma(\mathbf{x})$. We take the graph Butterfly as an example (which is described in Section 6.1.2). Without a specified root and the bounded-depth $h = 2$ replaced with a bounded-diameter $D = 2$, we wish to find the minimum cost bounded-diameter Steiner tree. The solution can be clearly observed which is shown in Figure 6.2. The root vertex becomes $v_0 = 5$ this time and is highlighted with a gray-filled color.

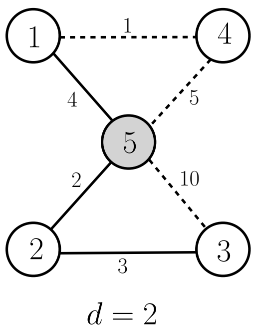


Figure 6.2: A minimum bounded-diameter Steiner tree of Butterfly graph

6.3 Bounded-Depth Minimum Spanning Tree

The minimum spanning tree is in graph theory. Classic algorithms exist for minimum spanning tree problems such as well-known Prims algorithm and Kruskals algorithm [40, 58]. However, with a depth constraint added, the problem becomes NP-hard [30, 22, 51]. Alfandari and Paschos (1999) in their work [4] also show that the problem cannot be solved within polynomial time with case bounded-depth equals 2 unless $P = NP$.
The Bounded-Depth MST problem (also known as hop-constrained MST problem or bounded-hop MST problem) arises in the design of centralized telecommunication networks with quality of service constraints [69]. Various applications can be found in [3].

Definition 23. *Given an weighted graph $G = (V, E)$ with a cost function $c\colon E \mapsto \mathbb{Z}$ and a natural number h, the* ***Bounded-Depth Minimum Spanning Tree*** *problem is to find a minimum cost spanning tree T of G, such that there exists a path in T from a specified root vertex v_0 to any other vertex with at most h edges.*

From the Definition 21, it is quite clear that the bounded-depth minimum spanning tree problem is the special case of the bounded-depth Steiner tree problem where the terminal set of Steiner tree is the set of all vertices (i.e., $U = V$). Based on this, we can adapt the formulation (6.1) for Bounded-Depth MST problem.
First, the variables involved in this QUBO formulation are the same as these in 6.1.1. Then, we replace set U in Equation (6.2) by set V and remove the second penalty term (6.3) since $V \backslash U = \emptyset$. With other terms remained, we obtain the following:

$$F(\mathbf{x}) = O(\mathbf{x}) + A \cdot P(\mathbf{x}) \tag{6.4}$$

where

$$P = |V| \cdot P_1(\mathbf{x}) + P_2(\mathbf{x})$$

with

$$P_1(\mathbf{x}) = \sum_{v \in V \backslash \{v_0\}} \left(1 - \sum_{(u,v)\in E} \sum_{i=1}^{h} x_{u,v,i}\right)^2$$

$$P_2(\mathbf{x}) = \sum_{v \in V \backslash \{v_0\}} \left(\sum_{(u,v)\in E} \sum_{i=2}^{h} x_{u,v,i} \left(1 - \sum_{(w,u)\in E} x_{w,u,i-1}\right)\right)$$

$$O(\mathbf{x}) = \sum_{(u,v)\in E} \sum_{i=2}^{h} c_{u,v} x_{u,v,i} + \sum_{(v_0,u)\in E} c_{v_0,u} x_{v_0,u,1}$$

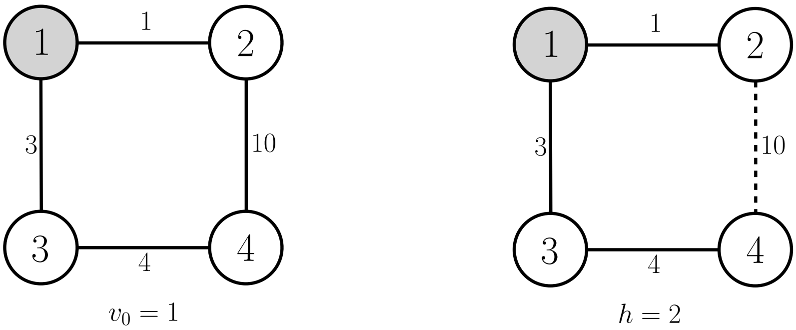
and

$$A = (|V| - 1) \cdot \max_{(u,v)\in E} (c_{u,v}) + 1$$

6.3.1 Example: a weighted graph C_4

Consider the graph C_4 in Figure 6.3a where vertex set $V = \{1, 2, 3, 4\}$, edge set $E = \{(1, 2), (1, 3), (2, 4), (3, 4)\}$ and a cost for each edge: $c_{1,2} = 1, c_{1,3} = 3, c_{2,4} = 10, c_{3,4} = 4$. Given a depth bound $h = 2$ and a dedicated root $v_0 = 1$, we wish to find the bounded-depth minimum spanning tree. A solution tree $T = (V, E_T)$ with $E_T = \{(1, 2), (1, 3), (3, 4)\}$, as shown in Figure 6.3b, can be observed clearly. Note that we use dashed line to mark these edges that are not in the solution.

The formulation requires $2 \cdot (2 - 1) \cdot (4 - 2) + 2 = 6$ variables:



(a) graph C_4

(b) a bounded-depth MST of C_4

Figure 6.3: A weighted C_4 and a solution

$$\mathbf{x} = \{x_{1,2,1}, x_{1,3,1}, x_{2,4,2}, x_{3,4,2}, x_{4,2,2}, x_{4,3,2}\}$$

Then, we compute each term:

$$\begin{aligned}
O(\mathbf{x}) &= \sum_{(u,v)\in E} \sum_{i=2}^{h} c_{u,v} x_{u,v,i} + \sum_{(v_0,u)\in E} c_{v_0,u} x_{v_0,u,1} \\
&= x_{1,2,1} + 3x_{1,3,1} + 10(x_{2,4,2} + x_{4,2,2}) + 4(x_{3,4,2} + x_{4,3,2}) \\
P_1(\mathbf{x}) &= \sum_{v\in V\setminus\{v_0\}} \left(1 - \sum_{(u,v)\in E} \sum_{i=1}^{h} x_{u,v,i}\right)^2 \\
&= \begin{array}{l} (1 - (x_{1,2,1} + x_{4,2,2}))^2 + (1 - (x_{1,3,1} + x_{4,3,2}))^2 \\ + (1 - (x_{2,4,2} + x_{3,4,2}))^2 \end{array}
\end{aligned}$$

$$
\begin{aligned}
& 1-2(x_{1,2,1}+x_{4,2,2})+x_{1,2,1}^2+x_{4,2,2}^2+2x_{1,2,1}x_{4,2,2} \\
= \quad & +1-2(x_{1,3,1}+x_{4,3,2})+x_{1,3,1}^2+x_{4,3,2}^2+2x_{1,3,1}x_{4,3,2} \\
& +1-2(x_{2,4,2}+x_{3,4,2})+x_{2,4,2}^2+x_{3,4,2}^2+2x_{2,4,2}x_{3,4,2} \\
P_2(\mathbf{x}) = & \sum_{v\in V\setminus\{v_0\}}\left(\sum_{(u,v)\in E}\sum_{i=2}^{h}x_{u,v,i}\left(1-\sum_{(w,u)\in E}x_{w,u,i-1}\right)\right) \\
= & \; x_{2,4,2}(1-x_{1,2,1})+x_{3,4,2}(1-x_{1,3,1})
\end{aligned}
$$

$$A = (4-1)\cdot 10 + 1 = 31$$

There are some linear terms as well as some constants that we need to process before we can encode them in a QUBO instance. As the technique mentioned in Chapter 5, all constants can be ignored and all linear terms $x_{u,v,i}$ can be replaced by $x_{u,v,i}^2$ since for all binary variables $x \in \{0,1\}$ we have $x = x^2$. After the processing we obtain the following:

$$
\begin{aligned}
F(\mathbf{x}) = & -123x_{1,2,1}^2-121x_{1,3,1}^2-83x_{2,4,2}^2-89x_{3,4,2}^2-114x_{4,2,2}^2-120x_{4,3,2}^2 \\
& -31x_{1,2,1}x_{2,4,2}+248x_{1,2,1}x_{4,2,2}-31x_{1,3,1}x_{3,4,2}+248x_{1,3,1}x_{4,3,2} \\
& +248x_{2,4,2}x_{3,4,2}
\end{aligned}
$$

With the coefficients of each quadratic term in $F(\mathbf{x})$ set to the corresponding entry, the complete QUBO matrix is obtained - Table 6.3. An optimal solution $\mathbf{x}^*$ for this QUBO

Table 6.3: QUBO matrix for graph C_4 with $h = 2$

variables	$x_{1,2,1}$	$x_{1,3,1}$	$x_{2,4,2}$	$x_{3,4,2}$	$x_{4,2,2}$	$x_{4,3,2}$
$x_{1,2,1}$	-123	0	-31	0	248	0
$x_{1,3,1}$		-121	0	-31	0	248
$x_{2,4,2}$			-83	248	0	0
$x_{3,4,2}$				-89	0	0
$x_{4,2,2}$					-114	0
$x_{4,3,2}$						-120

instance generated by a QUBO solver is

$$\mathbf{x}^* = [1, 1, 0, 1, 0, 0].$$

This encodes the optimal solution tree $S_{\mathbf{x}} = (V, E_{\mathbf{x}^*})$ where $V = \{1, 2, 3, 4\}$ and $E_{\mathbf{x}^*} = \{(1, 2), (1, 3), (3, 4)\}$.

6.3.2 Experimental results and discussion

In the initial experiment, we run two groups of test cases on D-Wave 2X for the Bounded-Depth Steiner Tree Problem, which consist of test cases for several small weighted graphs (see in Appendix A.3) with two different sets of depth constraints. Each weighted graph, together with a nominated root (red vertex), a terminal set (green vertices) and a depth constraint, comprise an instance of Bounded-Depth Steiner Tree Problem. We sum up the experimental results in Table 6.4 and Table 6.5. Column h represents the depth constraints assigned to each case. The rest columns have the same meaning as these columns interpreted in Section 5.4.

From the tables, we can see that the *success probability* decreases as the number of *physical qubits* required increases. In Table 6.4, the selections of depth constraint are relatively smaller comparing to the depth constraints in Table 6.5. Recall that the QUBO objective function (6.1) requires more variables when the depth constraint h becomes larger. In other words, more physical qubits are needed after embedding as h increases resulting in the lower success rates. For these QUBO instances whose embedding chain size is under three, we observe a success rate at 100 percent. This observation applies in the experimental results of Degree-constraint MST and Steiner tree as well, which again suggests a better accuracy of the D-Wave computer when the topology of a problem is closer to the host graph. The *density*, on the other hand, has no direct impact on the success probability, which makes a difference between empirical and theoretical result.

The second group of test cases is created for Bounded-Depth Minimum Spanning Tree Problem. They consist of ten K_6 graphs with their edges labeled with different weights ranging from 1 to 10. For each weighted K_6 graph, the $v_0 = 0$ is specified as root (see in Appendix A.4) and given depth constraints $h = 2$ and $h = 3$. A total of 15000 trials are done for each test case, and the results are shown in Table 6.6. The best valid solutions D-Wave can find are listed in column *Minimum D-Wave*.

We first notice that the *physical qubits* required after embedding differs from one to another even for the cases featuring the same size of logical qubits. The differences are due to the minor embedding algorithm we used which is heuristic. The algorithm uses a randomized method during its initial stage and results in the slight fluctuation of physical qubits as well as the embedding chain size [14]. Then, we again observe a significant decline in the *success probability* as problem size becomes large. When depth bound $h = 3$, the *physical qubits* required exceed 400, and the numbers of chain size are over 10. Only one of ten test cases succeeds in finding a ground state energy among its 15000 trials.

Table 6.4: Results for the Bounded-Depth Steiner Tree 1

Graph	Order	Size	h	Logical Qubits	Physical Qubits	Embedding Max Chain	Density	Success Probability	Optimal Answer
Bull	5	5	2	9	14	3	40	15000/15000	22
Butterfly	5	6	2	8	8	1	44.44	15000/15000	2
C_4	4	4	2	6	6	1	52.38	15000/15000	5
C_5	5	5	2	8	8	1	38.89	15000/15000	20
C_6	6	6	3	18	41	3	26.9	15000/15000	21
C_7	7	7	3	22	58	5	22.92	15000/15000	23
C_8	8	8	4	38	137	5	18.35	40/15000	41
C_9	9	9	4	44	141	5	15.25	254/15000	34
C_{10}	10	10	5	66	320	6	12.71	1/15000	33
C_{11}	11	11	5	74	390	7	11.6	0/15000	49
C_{12}	12	12	6	102	621	12	8.72	0/15000	12
Diamond	4	5	2	8	12	2	41.67	15000/15000	10
Grid2x3	6	7	3	22	65	5	27.67	3212/15000	7
Grid3x3	9	12	4	57	341	9	16.33	0/15000	16
Hexahedral	8	12	4	57	422	13	17.48	0/15000	17
House	5	6	2	10	12	2	32.73	15000/15000	8
$K_{2,3}$	5	6	2	9	10	2	33.33	15000/15000	12
$K_{3,3}$	6	9	3	27	120	6	26.46	18/15000	18
K_3	3	3	2	4	4	1	80	15000/15000	10
K_4	4	6	2	9	16	3	48.89	15000/15000	9
K_5	5	10	2	16	36	3	38.24	11136/15000	12
Q_3	8	12	4	57	404	11	17.42	0/15000	20
Wagner	8	12	4	57	400	9	17.12	0/15000	9

Table 6.5: Results for the Bounded-Depth Steiner Tree 2

Graph	Order	Size	h	Logical Qubits	Physical Qubits	Embedding Max Chain	Density	Success Probability	Optimal Answer
Bull	5	5	5	33	176	11	15.18	7/15000	15
Butterfly	5	6	5	20	43	3	4.72	10019/15000	2
C_4	4	4	4	14	39	4	4.42	11979/15000	5
C_5	5	5	5	26	93	6	8.11	1642/15000	18
C_6	6	6	6	42	229	8	15.12	13/15000	19
C_7	7	7	6	52	310	11	13.85	0/15000	20
C_8	8	8	5	50	245	7	13.79	5/15000	41
C_9	9	9	5	58	277	7	10.04	3/15000	34
C_{10}	10	10	6	82	687	19	9.2	0/15000	33
C_{11}	11	11	6	92	570	15	11.31	0/15000	49
C_{12}	12	12	6	102	753	18	6.59	0/15000	12
Diamond	4	5	4	20	102	7	7.8	88/15000	8
Grid2x3	6	7	6	52	409	14	16.7	1/15000	7
Grid3x3	9	12	5	75	612	16	12.46	0/15000	16
Hexahedral	8	12	5	75	712	15	14.7	0/15000	17
House	5	6	5	34	199	8	14.7	23/15000	8
$K_{2,3}$	5	6	5	27	132	8	12.34	729/15000	12
$K_{3,3}$	6	9	6	63	817	25	16.58	0/15000	18
K_3	3	3	3	6	9	2	1.09	15000/15000	10
K_4	4	6	4	21	116	7	8.35	734/15000	9
K_5	5	10	5	52	811	23	22.99	1/15000	12
Q_3	8	12	5	75	786	22	15.97	0/15000	20
Wagner	8	12	5	75	746	17	14.82	0/15000	9

Table 6.6: Results for Bounded-Depth MST

Graph	Order	Size	h	Logical Qubits	Physical Qubits	Embedding Max Chain	Density	Success Probability	Minimum D-Wave	Optimal Answer
K_{6-1}	6	15	2	25	90	4	23.08	145/15000	9	9
K_{6-2}	6	15	2	25	85	5	23.08	198/15000	25	25
K_{6-3}	6	15	2	25	84	5	23.08	799/15000	13	13
K_{6-4}	6	15	2	25	92	5	23.08	565/15000	17	17
K_{6-5}	6	15	2	25	71	5	23.08	288/15000	21	21
K_{6-6}	6	15	2	25	96	7	23.08	131/15000	23	23
K_{6-7}	6	15	2	25	89	9	23.08	672/15000	23	23
K_{6-8}	6	15	2	25	89	7	23.08	729/15000	18	18
K_{6-9}	6	15	2	25	96	5	23.08	333/15000	17	17
K_{6-10}	6	15	2	25	80	4	23.08	216/15000	18	18
K_{6-1}	6	15	3	45	413	14	25.6	0/15000	10	7
K_{6-2}	6	15	3	45	468	16	25.6	0/15000	27	24
K_{6-3}	6	15	3	45	438	14	25.6	0/15000	16	12
K_{6-4}	6	15	3	45	435	15	25.6	0/15000	14	13
K_{6-5}	6	15	3	45	414	12	25.6	1/15000	18	18
K_{6-6}	6	15	3	45	487	16	25.6	0/15000	24	21
K_{6-7}	6	15	3	45	479	16	25.6	0/15000	21	19
K_{6-8}	6	15	3	45	424	18	25.6	0/15000	21	14
K_{6-9}	6	15	3	45	459	15	25.6	0/15000	18	16
K_{6-10}	6	15	3	45	443	14	25.6	0/15000	20	14