UNIVERSITY OF CALIFORNIA

Los Angeles

Towards Quantum Computing:
Solving Satisfiability Problem by Quantum Annealing

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Electrical Engineering

by

Juexiao Su

2018

ABSTRACT OF THE DISSERTATION

Towards Quantum Computing:
Solving Satisfiability Problem by Quantum Annealing

by

Juexiao Su
Doctor of Philosophy in Electrical Engineering
University of California, Los Angeles, 2018
Professor Lei He, Chair

To date, conventional computers have never been able to efficiently handle certain tasks, where the number of computation steps is likely to blow up as the problem size increases. As an emerging technology and new computing paradigm, quantum computing has a great potential to tackle those hard tasks efficiently. Among all the existing quantum computation models, quantum annealing has drawn significant attention in recent years due to the realization of the commercialized quantum annealer, sparking research interests in developing applications to solve problems that are intractable for classical computers.

However, designing and implementing algorithms that manage to harness the enormous computation power from quantum annealer remains a challenging task. Generally, it requires mapping of the given optimization problem into quadratic unconstrained binary optimization(QUBO) problem and embedding the subsequent QUBO onto the physical architecture of quantum annealer. Additionally, practical quantum annealers are susceptible to errors leading to low probability of the correct solution.

In this study, we focus on solving Boolean satisfiability (SAT) problem using quantum annealer while addressing practical limitations. We have proposed a mapping technique that maps SAT problem to QUBO, and we have further devised a tool flow that embeds the QUBO onto the architecture of a quantum annealing device. Additionally, We have optimized the proposed embedding flow to reduce run-time in addition to shortening the qubit chain length, leading to robust quantum annealing. To further improve the reliability of quantum annealing, we have also developed a post processing embedding technique that enlarges the energy gap between ground state and the first excited state. To demonstrate the effectiveness of proposed methods, we have conducted experiments on real quantum annealing devices manufactured by D-Wave Systems, showing compelling result of using quantum annealer to solve SAT problem.

The dissertation of Juexiao Su is approved.

Todd D. Millstein

Sudhakar Pamarti

Puneet Gupta

Lei He, Committee Chair

University of California, Los Angeles

2018

I dedicate my PhD dissertation to my wife, my son and my parents.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Prof. Lei He, for his guidance and support in my research and career development.

I would like to thank Prof. Puneet Gupta, Prof. Sudhakar Pamarti, and Prof. Todd D. Millstein for their support, guidance, patience, and many valuable comments that improved the contents of this dissertation. My thanks also go to Dr. Aidan Roy, for his insightful comments and encouragement, as well as for the great help which enables me to conduct experiments on a real quantum annealer.

I am also grateful to my fellow PhD students, Ju-Yueh Lee, Tianheng Tu, Min Gao, Yunxuan Yu, Wei Wu, Xiao Shi for their help, comments, and stimulating discussions we have had in the last five years.

Last but not the least, I would like to thank my family: my parents, my mother-in-law, and my father-in-law for their support and encouragement. I would also like to thank my son James Su, who has inspired me with his love. At the end, I would like to express appreciation to my beloved wife Dr. Xiaowen Xu, for her determination and constant encouragement.

VITA

2007-2011 Bachelor of Engineering, School of Astronautics, Beihang University, Beijing, China.

2011–2013 Master of Science, Mechanical and Aerospace Engineering Department, University of California, Los Angeles, California

PUBLICATIONS

J. Su and L. He, Fast Embedding of Constrained Satisfaction Problem to Quantum Annealer with Minimizing Chain Length, in Proceedings of the 54th Annual Design Automation Conference 2017, New York, NY, USA, 2017, p. 77:177:6.

J. Su, J. Y. Lee, C. Wu, and L. He, In-place LUT polarity inVersion to mitigate soft errors for FPGAs, in 2016 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2016, pp. 8186.

J. Su, T. Tu, and L. He, A Quantum Annealing Approach for Boolean Satisfiability Problem, in Proceedings of the 53rd Annual Design Automation Conference, New York, NY, USA, 2016, p. 148:1148:6.

J. Y. Lee et al., Heterogeneous configuration memory scrubbing for soft error mitigation in FPGAs, in 2012 International Conference on Field-Programmable Technology (FPT), 2012, pp. 2328.

F. Xu, X. Song, X. Wang, and J. Su, Neural Network model for earthquake prediction using

DMETER Data and Seismic Belt information, in Intelligent Systems (GCIS), 2010 Second WRI Global Congress on, 2010, vol. 3, pp. 180183.

CHAPTER 1

Introduction

Quantum computing has been proven in theory to be more efficient in solving certain classes of optimization problems [CD10]. In addition, experiments[BRI14] showed that quantum computer has a much smaller scaling factor when dealing with the problem that has the same complexity on classical computer. Among all the existing quantum heuristics, quantum annealing (QA) has drawn significant attention in recent years due to the realization of the first commercialized QA device.

1.1 Quantum Annealing

QA was first proposed as an extension of simulated annealing that is able to avoid local minimal [FGS94]. Later, practical computation framework of QA has been proposed and developed by [FGG01]. It works by starting a physical system in a ground state and gradually varying the system until a solution of the target optimization problem is obtained.

Quantum annealing offers potential computational power by taking advantage of quantum mechanical effects. The evolution of quantum annealing is governed by the time dependent Schrödinger equation: it starts from the initial system whose ground state can be easily constructed; then, it adiabatically evolves to the final system where the optimization problem is elaborately encoded, as shown in Eq. (1.1) (1.2). The final state, which is also the ground state of the final system, immediately encodes the solution of the optimization problem.

$$\boldsymbol{H}(t)\boldsymbol{\psi}(t) = i\hbar\tfrac{\partial}{\partial t}\boldsymbol{\psi}(t) \tag{1.1}$$

$$H(\tau) = A(\tau)H_b + B(\tau)H_p \qquad \tau = t/t_a \tag{1.2}$$

1.2 D-Wave Quantum Annealer

Recently, D-Wave Systems has successfully built such quantum annealing device whose system Hamiltonian can be expressed by Ising model, as shown in Eq. (1.3) (1.4).

$$E(s) = \sum_{1 \le i \le N} h_i s_i + \sum_{1 \le i \le j \le N} J_{ij} s_i s_j \tag{1.3}$$

$$\mid h_i \mid \le 2, \mid J_{ij} \mid \le 1, and\, s_i \in \{+1, -1\} \tag{1.4}$$

where s is the qubit that indicates the system state by the direction of the circulating current. h and J are programmable coefficients, specifying the energy bias and coupling strength, so that different optimization problems can be encoded. A plethora of problems can be reduced to Ising energy minimization problem[PDD12][RVO15][NDR12], which is the native problem solved by D-Wave annealer. It is immediately apparent that the quantum annealer is capable of solving any problems as long as the problem can be represented in the form of Ising energy minimization problem. However, in practice, not every Ising energy problem can be directly solved by D-Wave annealer due to the limited connectivity in the physical architecture. Therefore, embedding the mapped Ising problem to the annealer's architecture becomes a necessary step in using D-Wave quantum annealer.

Several researches [PFN15] [BCI14] [PDD12] have proposed approaches to utilize the D-Wave annealer; nevertheless, they are far from touching the real world problems. We summarized two major difficulties in programming the QA device to solve practical optimization problems.

- Mapping: the optimization problem has to be formulated in the form of finding the ground state in a energy function described by the Ising model.
- Embedding: the mapped problem has to fit into the underlying topology of the actual device, where each qubit has limited connectivity to interact with other qubits.

Additionally, quantum annealing is also susceptible to errors. Theoretically, it guarantees the finding of ground state in the final Hamiltonian if the system evolves sufficiently slowly; yet errors are still observed when experimenting with the practical quantum annealer [DJA13][LAB14], where the returned solution is not correct. As suggested by the research[CFP01][NTK15][JAG11][KM14], the robustness of quantum annealing is related to the energy gap Δ between the ground state and the first excited state. Moreover, Ising chain, which has been extensively used in many embedding techniques, impedes the progress of quantum annealing if their size becomes large, leading to wrong solution[VMK15]. Therefore, finding the valid embedding is not enough, the quality of embedding which leads to robust annealing should also be considered in designing embedding algorithms.

1.3 Thesis Outline

In this research, we focus on solving Boolean satisfiability(SAT) problem by D-Wave quantum annealer while overcoming practical difficulties. Specifically, the contribution of this research are summarized as follows:

1. Formulate SAT problem into Ising energy minimization problem
2. Develop algorithms to embed the formulated problem onto the architecture of the actual device
3. Improve embedding quality with Ising chain minimization
4. Enlarge energy gap for existing embedding result
5. Evaluate the behavior of solving SAT problem on the current generation of D-Wave machine

The rest of the thesis is organized as follows. Chapter 2 introduces the QUBO formulation of SAT problem. Chapter 3 presents the embedding technique. Chapter 4 discusses the optimization of chain length. Chapter 5 presents the post-embedding optimization to enlarge energy gap. Chapter 6 summarizes this work.

CHAPTER 2

SAT Problem and Quantum Annealing

2.1 Chapter Introduction

SAT problem is stated as: given a Boolean function defined over n variables, find an assignment of variables such that the Boolean function is true. We propose a method to transform the Boolean SAT problem to Ising energy minimization problem. In General, our approach can be divided into 3 steps:

1. Convert the Boolean function in SAT into an acyclic Boolean network with only basic logic operations.
2. Convert the Boolean network to an Ising model by introducing small gate Ising model and Ising chains.
3. Set constraints so that Ising model output is logic one. Then, find the corresponding assignments.

Since the Boolean function is directly related to Boolean network, it can be easily converted using a naive algorithm. Hence our focus is on the last two steps. In this chapter, we will also use a toy example, as shown in Eq. (2.1), to illustrate how to find the Ising model which can be used to determine satisfiability.

$$f = (x_1 \vee \neg x_3) \wedge (x_2 \vee \neg x_3) \tag{2.1}$$

2.2 Ising Model and QUBO

Compared with Ising model, quadratic unconstrained binary optimization (QUBO) problem has a slightly different representation. In the Ising model, the variable s_i is defined over $\{-1, +1\}$, whereas in QUBO, the variable x_i is defined over $\{0, 1\}$. The two representations can be interchanged by replacing the variable in the Ising model with with $s_i = 1 - 2x_i$ and altering the coefficient accordingly. The energy function in Ising model has the same form as the cost function in QUBO. Therefore, Ising model and QUBO are equivalent. In this thesis, we do not distinguish the two representations in the following discussion.

2.3 Ising Model for Boolean Function

[BCM10] has proposed expressions of the Ising model for three logic operators. In this work, we extend the discussion to variant of these three operators. Subsequently, we propose the gate Ising models, which are the basic elements that can be used to represent any given Boolean function using Ising model.

In logic, an arbitrary Boolean formula can be represented by a logic gate network, which consists of gates and wires. Figure 2.1a shows the logic network that represents the Boolean formula in Eq. (2.1). We then write the system Hamiltonian as

$$H_{sys} = \sum H_{G_i} + \sum H_{C_j} \tag{2.2}$$

where each H_{O_i} depends only on the logic gates and H_{C_j} depends only on the wires, resembling the components in the original Boolean network. Eq. (2.2) is an ideal form, as it can be conveniently represents any Boolean network. The philosophy behind Eq. (2.2) is that finding small piece of Ising model to represent logic gate and wires , and then putting them together to formulate large Ising model for entire logic network. In the remaining of this chapter, we will discuss the steps to achieve Eq. (2.2).

2.3.1 Gate Ising Model

To represent logic gate in Ising model, we construct an Ising model penalty function, as shown in Eq. (2.5).

$$Penalty_{C_j}(\boldsymbol{s}) = \begin{cases} k & \text{if } \boldsymbol{s} \in C_i \\ \geq k+g & \text{if } \boldsymbol{s} \notin C_i \text{ and } g > 0 \end{cases} \tag{2.3}$$

where $\boldsymbol{s}$ is an assignment, C_i is the constrain which encodes the logic function.

Here, we take NAND gate as an example to illustrate how to construct gate Ising model. We assume that a 3-qubit quantum annealer with full connectivity is used to construct the Ising model for a NAND gate. The 3-qubit system Hamiltonian can be written as Eq. (2.4).

$$H(s_1, s_2, s_3) = h_1 s_1 + h_2 s_2 + h_3 s_3 + J_{12} s_1 s_2 + J_{13} s_1 s_3 + J_{23} s_2 s_3 \tag{2.4}$$

The remaining task is to decide the energy bias $\boldsymbol{h}$ and coupling strength $\boldsymbol{J}$ such that the ground state encodes the NAND truth table. We then write the Ising model as follow

$$H_{NAND}(\boldsymbol{s}, \boldsymbol{h}, \boldsymbol{J}) = \begin{cases} k & \text{if } \boldsymbol{s} \in C \\ \geq k+g & \text{if } \boldsymbol{s} \notin C \text{ and } g > 0 \end{cases} \tag{2.5}$$

where $\boldsymbol{h}$, $\boldsymbol{J}$ are the coefficients of the hardware and $C = \{(-1,-1,1)\ ,(-1,1,1),(1,-1,1),(1,1,-1)\}$. The purpose of Eq.(2.5) is to penalize the states that are not in the NAND truth table. Subsequently, we cast the problem into a linear programming(LP) problem, as shown in Eq. (2.6), where $\boldsymbol{h}$, $\boldsymbol{J}$, g and k are variables.

$$
\begin{array}{llcl}
\max & g & & \\
\text{s.t.} & H(-1,-1,1) & = & k \\
& H(-1,1,1) & = & k \\
& H(1,-1,1) & = & k \\
& H(1,1,-1) & = & k \\
& H(-1,-1,-1) & = & k+g \\
& H(-1,1,-1) & = & k+g \\
& H(1,-1,-1) & = & k+g \\
& H(1,1,1) & = & k+g \\
& -1 \leq & J_{ij} & \leq 1 \\
& -2 \leq & h_i & \leq 2 \\
& g & > & 0
\end{array}
\tag{2.6}
$$

Table 2.1: Energy Landscape for NAND Ising Model

S1	S2	S3	Energy
-1	-1	-1	4.5
-1	-1	1	-1.5
-1	1	-1	0.5
-1	1	1	-1.5
1	-1	-1	0.5
1	-1	1	-1.5
1	1	-1	-1.5
1	1	1	0.5

The feasibility of the above LP problem dictates if the embedding solution exists for a 3-qubit system. The objective is designed to maximize the energy gap between C and $\overline{C}$, as large energy gap increases the ground state probability and reduces the required annealing time[FGG00][VMK15], we will discuss more in Chapter 5. By finding the feasible solution for Eq.(2.6), the Ising model for NAND gate can be found, as shown in Figure 2.1. Based on the solution, we examine the result by listing all possible energies of this 3-qubit system, as shown in Table 2.1. Obviously, the system energy is minimized if and only if $\boldsymbol{s} \in C$.

We further conclude the Ising models for 16 basic 2-input logic operations using 3 qubits in Table 2.2. For each Ising model, the system energy is minimized when the state of three variables satisfies the corresponding logic relation.

Table 2.2: Basic Ising Primitives

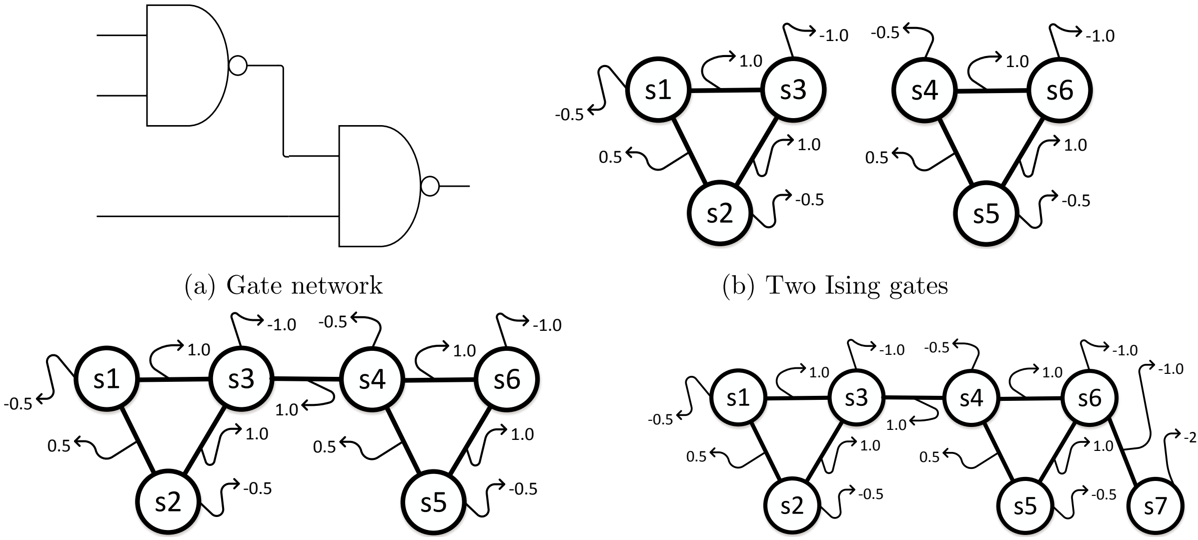
Num	Logic function	Ising function
1	$z = 0$	$2S_z$
2	$z = x \wedge y$	$-S_x - S_y + 2S_z - 2S_xS_z - 2S_yS_z + S_xS_y$
3	$z = x \wedge \neg y$	$-S_x + S_y + 2S_z - 2S_xS_z + 2S_yS_z - S_xS_y$
4	$z = x$	$-2S_xS_z$
5	$z = \neg x \wedge y$	$+S_x - S_y + 2S_z + 2S_xS_z - 2S_yS_z - S_xS_y$
6	$z = y$	$-2S_yS_z$
7	$z = x \oplus y$	None
8	$z = x \vee y$	$S_x + S_y - 2S_z - 2S_xS_z - 2S_yS_z + S_xS_y$
9	$z = \neg x \wedge \neg y$	$S_x + S_y + 2S_z + 2S_xS_z + 2S_yS_z + S_xS_y$
10	$z = x \leftrightarrow y$	None
11	$z = \neg y$	$2S_yS_z$
12	$z = x \vee \neg y$	$S_x - S_y - 2S_z - 2S_xS_z + 2S_yS_z - S_xS_y$
13	$z = \neg x$	$2S_xS_z$
14	$z = \neg x \vee y$	$-S_x + S_y - 2S_z + 2S_xS_z - 2S_yS_z - S_xS_y$
15	$z = \neg x \vee \neg y$	$-S_x - S_y - 2S_z + 2S_xS_z + 2S_yS_z + S_xS_y$
16	$z = 1$	$-2S_z$

2.3.2 Chain Ising Model

The gates in the logic network are connected with wires. In Ising model, wire is also needed, ensuring the qubit belong to different gate Ising model take the same value. Wiring two qubits can simply be achieved by applying negative coupling coefficient, which is also known as chain. Figure 2.1c shows the chain connecting qubit S_3 and S_4. By combining the discussion of gate Ising model and chain Ising model, we can write the Ising model for Figure 2.1a as

$$\begin{aligned} H_{NAND} = \; & -0.5s_1 - 0.5s_2 - s_3 + s_1s_3 + s_2s_3 + 0.5s_1s_2 \\ & -0.5s_4 - 0.5s_5 - s_6 + s_4s_6 + s_5s_6 + 0.5s_4s_5 \\ & -s_3s_4 \end{aligned} \tag{2.7}$$

where the first 12 terms describe the two NAND Ising gates and the last term describes the chain. Figure 2.1c shows the graph representation of the same Ising model.



(a) Gate network

(b) Two Ising gates

(c) Connected Ising gates

(d) Ising model with forced output

Figure 2.1: This is an example showing the conversion of gate network to Ising model.

2.4 Boolean Satisfiability

The ground state of the system with Hamiltonian described by Eq. (2.7) encodes all possible states of the example Boolean network. That is, if the quantum annealer is programmed as Eq. (2.7), the result returned by the quantum annealer will be one of the logic circuit state.

Theorem 1. *The total energy of the converted Ising model is minimized if and only if the* *qubit state of the Ising model is one of the states that satisfy all the logic operations in the acyclic Boolean network.*

Proof. Taking one possible state from Boolean network and assigning it to the Ising model, the total energy is minimized because the energies of all gate Ising models and chains are minimized.

In other direction, it can be proved by contradiction. We assume that the total energy is minimized and one primitive or chain violates the logic relation. Due to the acyclicity, there are always possible states of the Boolean network satisfying all logic operations. By comparing the violated and satisfied scenarios, one can conclude that the violated gate or chain will have the same energy with the counterpart in the all satisfied Ising model. At this point, we arrived at contradiction which there exist an Ising gate model whose energy is also minimized when the logic relation is not satisfied. Then, the energy of Boolean network Ising model is minimized if and only if the states satisfy the all logic relations in Boolean network. □

In fact, the qubits on quantum annealer is constructed in superposition states, representing all possible state of the given Boolean network. However, for SAT problem, screening for all possible states of a Boolean network is not enough. We need to further narrow it down to the states that make the Boolean function evaluate to being true. Although it is impossible to preset a logic circuit output to one, to manipulate the output of the Ising model is relatively easy. An intuitive way is to connect the output qubit with an Ising model that always produces one, as shown in Figure 2.1d. The negative weight is assigned to a single qubit S7, therefore the system energy for this single qubit system is minimized if the qubit takes one, making it becomes a constant one Ising model.

When chaining the output of a Boolean network Ising model with constant one Ising model, as shown in 2.1d, there will be two scenarios: If the given Boolean function is satisfiable, the returned state minimizes energy of all Ising gates and chains, encoding the assignment that makes output to one; on the other hand, if the given Boolean function is unsatisfiable, the returned states still minimizes the energy of the entire Ising model, however there exists at least one Ising gate or chain whose energy is not minimized.

Conveniently, the lower bound of the converted Ising model can be used to determine the satisfiability. With given Boolean network Ising model with chained constant one Ising model, the lower bound can be calculated by adding up the minimal energy of every Ising gate, Ising chain and constant one Ising model. Therefore, whether the lower bound can be achieved determines the satisifiability of the original Boolean function.

The greatest advantages of the proposed Ising model formulation is that we are constructing an Ising model using small Ising gates and connecting them by chains. Compared to other Ising model formulations that also employ chains [BCM10][BCI14], our method is simplified because the energies in gates and chains are separated, relaxing the the requirement of strong negative coupling strength in chains. Besides, we formulate a gate-level netlist, where each gate is represented by an Ising gate. Therefore, when embedding it to a physical quantum annealer, we are able to leverage integrated circuit design automation techniques, such as logic optimization, placement and routing.

CHAPTER 3

Embedding

3.1 Chapter Introduction

Mapping the original problem to QUBO does not guarantee that practical quantum annealer, such as D-Wave, will solve it. One obvious constraint is the limited connectivity on the physical architecture. Therefore, embedding the mapped Ising energy minimization problem to the annealer's architecture becomes a necessary step in using D-Wave quantum annealer.

Many researches [AH15][PDD12][RHG16] have studied a wide range of applications that can be mapped to QUBO including machine learning, protein folding, satisfiability problem and trading strategy. However, only a few researches discussed embedding techniques and those techniques are not even scalable to the current generation of D-Wave annealer when embedding certain types of QUBO problems [Cho08][ADF11][CMR14]. It is desired to have an efficient embedding technique that not only manages to handle the problem with the size of the current architecture but is also scalable to the future devices with much larger capacity. [BCI16] proposed scalable techniques that models the embedding as a place-and-route problem in electronic design automation. Here, we propose an embedding technique for the Ising model described in Chapter 2. The major steps can be summarized as follows: (1) local embedding, (2) Ising gate placement, (3) chain routing.

Beginning with local embedding, gate Ising model is embedded into a sub-graph of the annealer's architecture topology. Then, during placement, the embedded gate Ising models are spread out on the architecture. Lastly, the gate Ising model's inputs and outputs are connected by chains. In this chapter, we discuss the embedding flow.

3.2 D-Wave Architecture

The qubits in the D-Wave annealer is interconnected to form a Chimera graph, which consists of $M \times N$ cells[HJL10][DJA13]. Each cell consists 8 qubits with $K_{4,4}$ bipartite topology. These 8 qubits are also interconnected with adjacent cells vertically and horizontally. Figure 3.1 shows the D-Wave architecture with 2×4 cells.

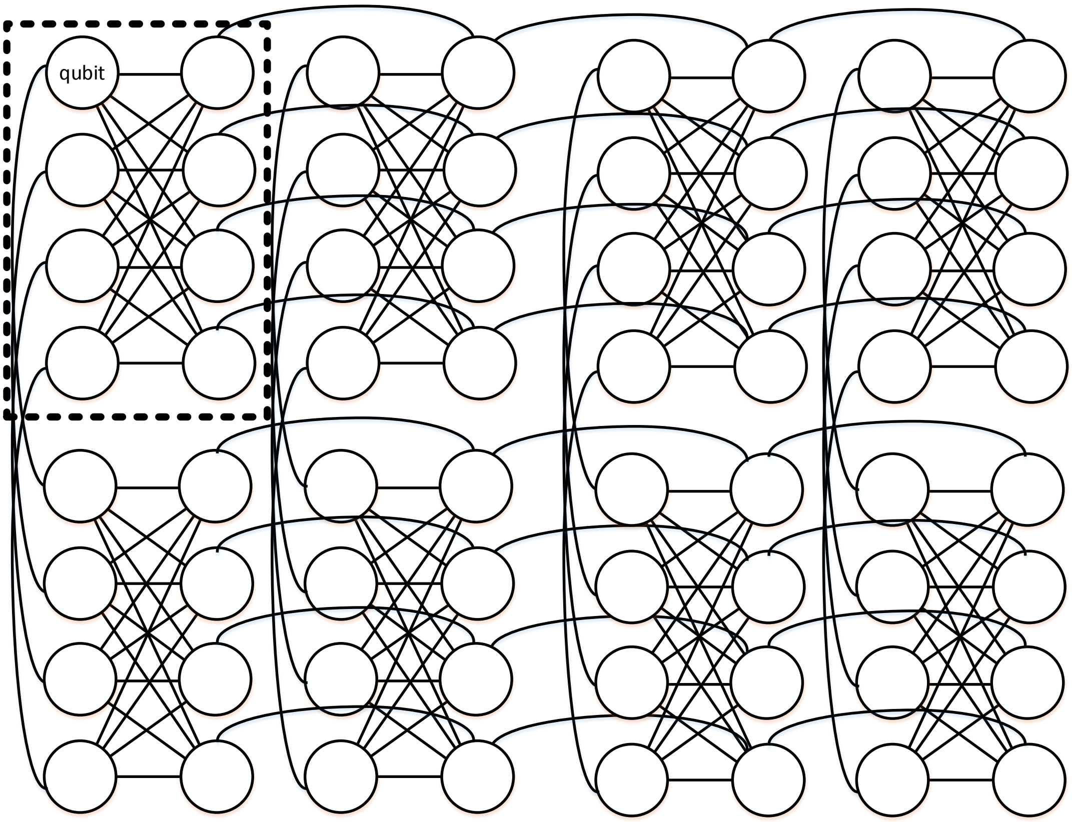


Figure 3.1: D-Wave Chimera Graph with 2×4 cells

The task of embedding is to find an equivalent Ising model whose pattern can be found in the D-Wave architecture. For example, an Ising problem with two qubits can be directly mapped to a D-Wave with any two qubits and a coupler in between. However, an Ising problem with three qubits cannot be directly mapped to the hardware as there is no K_3 complete graph pattern in the architecture.

As discussed in Chapter 2, Boolean network can be represented by Ising model which comprises gate Ising model and chain Ising model. If the embedding of gate Ising model can be found, the embedding of the entire Ising model can be achieved by spreading out gate Ising model and connecting them by chains, which is essentially a place-and-route problem.

3.3 Local Embedding

As shown in Figure 3.1, the cell structure is organized as a $K_{4,4}$ bipartite graph. Qubits on the left column is able to interact with any qubits on the right column, while qubits on the same column is unable to interact with each other. As suggested by [Hea13], one can easily embed a K_4 full graph onto the cell pattern by chaining qubits on the same row, as shown in Figure 3.2. Therefore, the connectivity of cell can be equivalently transformed from $K_{4,4}$ bipartite graph to K_4 full graph. When converting logic Ising model to physical Ising model in a cell, weights on qubits and couplers are equally distributed to the underlying physical qubits.

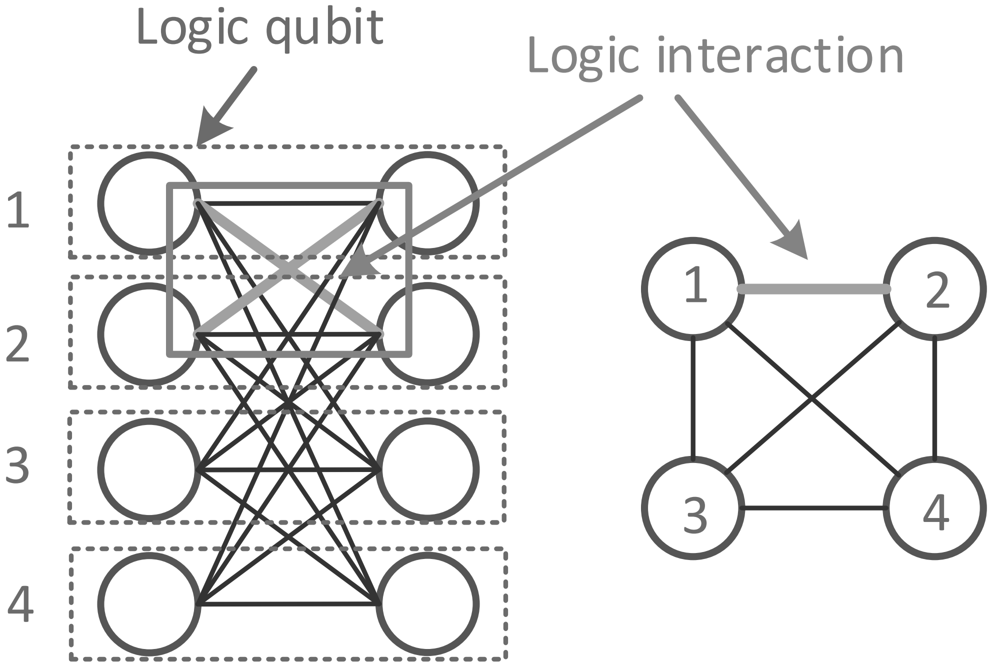


Figure 3.2: Cell with Logic Qubits

Alternatively, we can write the mathematical formulation, which leads to the embedding solution. We present an example of embedding a NAND gate using $K_{3,3}$ pattern.

Eq. (3.1) shows the Ising model for $K_{3,3}$ pattern.

$$\begin{gathered} H_{3,3}(\boldsymbol{s}) = h_1 s_1 + h_2 s_2 + h_3 s_3 + h_4 s_4 + h_5 s_5 + h_6 s_6 \\ J_{14} s_1 s_4 + J_{15} s_1 s_5 + J_{16} s_1 s_6 + J_{24} s_2 s_4 + J_{25} s_2 s_5 + J_{25} s_2 s_5 \\ J_{34} s_3 s_4 + J_{35} s_3 s_5 + J_{36} s_3 s_6 \end{gathered} \tag{3.1}$$

Similar to the discussion in Chapter 2, finding the local embedding for single NAND gate can be formulated into the below constraint programming problem.

$$\begin{array}{lrcll} \text{max} & g & & & \\ \text{s.t.} & H_{3,3}(\boldsymbol{s}) & = & k & if s \in C \\ & H_{3,3}(\boldsymbol{s}) & = & k+g & if s \notin C \\ & -1 \leq & J_{ij} & \leq 1 & \\ & -2 \leq & h_i & \leq 2 & \\ & g & > & 0 & \end{array} \tag{3.2}$$

where $C = \{(-1,-1,1,-1,-1,1), (-1,1,1,-1,1,1), (1,-1,1,1,-1,1), (1,1,-1,1,1,-1)\}$. Note that C is set of interest, which in-explicitly includes the logic qubit constraint that ensures the qubits in the same row will take the same state. Eq. (3.2) can be extended to find the embedding for other 2-input gates with different C.

3.4 Placement

Now that, each Ising gate can be embedded to a single cell, the remaining task is to spread out the gate Ising models and connect them by chains. A good placement substantially decides the quality and efficiency of in routing stage. The objectives of placement are placing the connected gate Ising model close to each other and leaving enough space for routing.

We adopted the simulated annealing framework to place Ising gate models, as shown in Algorithm 1. According to [LD88], simulated annealing makes the largest placement improvement when succeed swap rate α is around 0.44. In our placement, the update temperature function is designed so that temperature drops more slowly when α is around

Algorithm 1 Simulated Annealing Placement

```
initializePlacement();
initializeTemperature();
while (!shouldExit()) {
    while(!isMoveLimit()) {
        trySwapPlacedCell();
        $\Delta$Cost = cost($P_{new}$) - cost($P_{old}$);
        r = random(0, 1);
        if ( r < $e^{-\Delta Cost/T}$ ) {
            acceptMove();
        } else {
            revertMove();
        }
    }
    T = updateT($\alpha$);
}
```

0.44.

The cost function used in our placement is illustrated by Eq. (3.3). Similar to the placement algorithm for FPGA [MBR00], we use the summation of half perimeter wire length (HPWL) as our cost function, where the compensation coefficient $q(i)$ for multi-fanout net is based on the research [Che94]. To estimate the local congestion within bounding box of each wire, we consider the routing supply by counting the available unmapped cells. For each mapped cell, only one qubit is available to connect with other qubits by constructing chains, leaving each mapped cell only have one vertical and one horizontal routing resource supply. As for an unmapped cell, each has four vertical and four horizontal routing resource supplies. By introducing routing supply, the cost function will have smaller value if gates are placed in the area that have more routing resource supply.

$$Cost = \sum_{i \in Nets} q(i)[\frac{bb_x(i)}{S(i)} + \frac{bb_y(i)}{S(i)}] \tag{3.3}$$

3.5 Routing

After placement, the remaining task is to connect gate Ising models by chains. However, for a large Ising model that consists of many small Ising gates, it is a non-trivial problem that routing all the wire to the designated location: a previously routed wire may block other wire later. Therefore an efficient routing algorithm is the key to successfully embedding the Ising model. We extend the pathfinder algorithm[ME95] to solve the chain routing problem. One of the key philosophy in the pathfinder algorithm is using criticality to decide when to take the direct route and when to detour.

3.5.1 Routing Graph

Routing graph facilitates the routing algorithms. Figure 3.3 shows the local routing graphs used and unused cells. To solve the qubit assignment problem, we employ virtual MUX to separate the gate Ising model and cell pattern. Therefore, the routing result can be used to decide the usage of actual physical qubits. For example, if top most node on the left of virtual mux and out node are used in routing, then when generating the actual configuration, the out qubit will be implemented by 2 physical qubits in the first row of a cell.

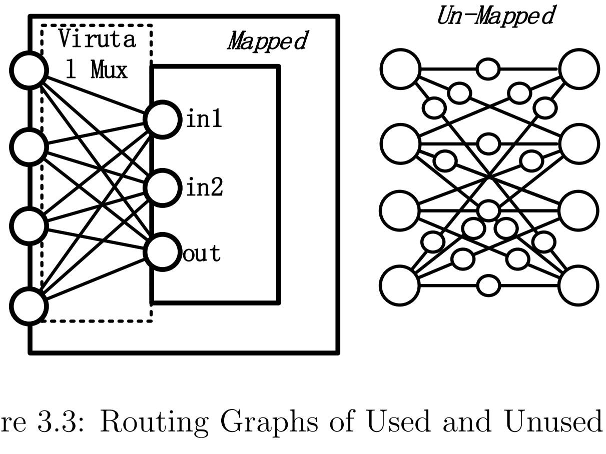


Figure 3.3: Routing Graphs of Used and Unused Cell

A routing resource graph (RR graph) is built based on the post-placement solution, as shown in Figure 3.4. All qubits and their interactions are represented by nodes in the RR graph and connectivity by edges. The RR graph is an undirected graph, as qubits and their interactions can be used in both directions.

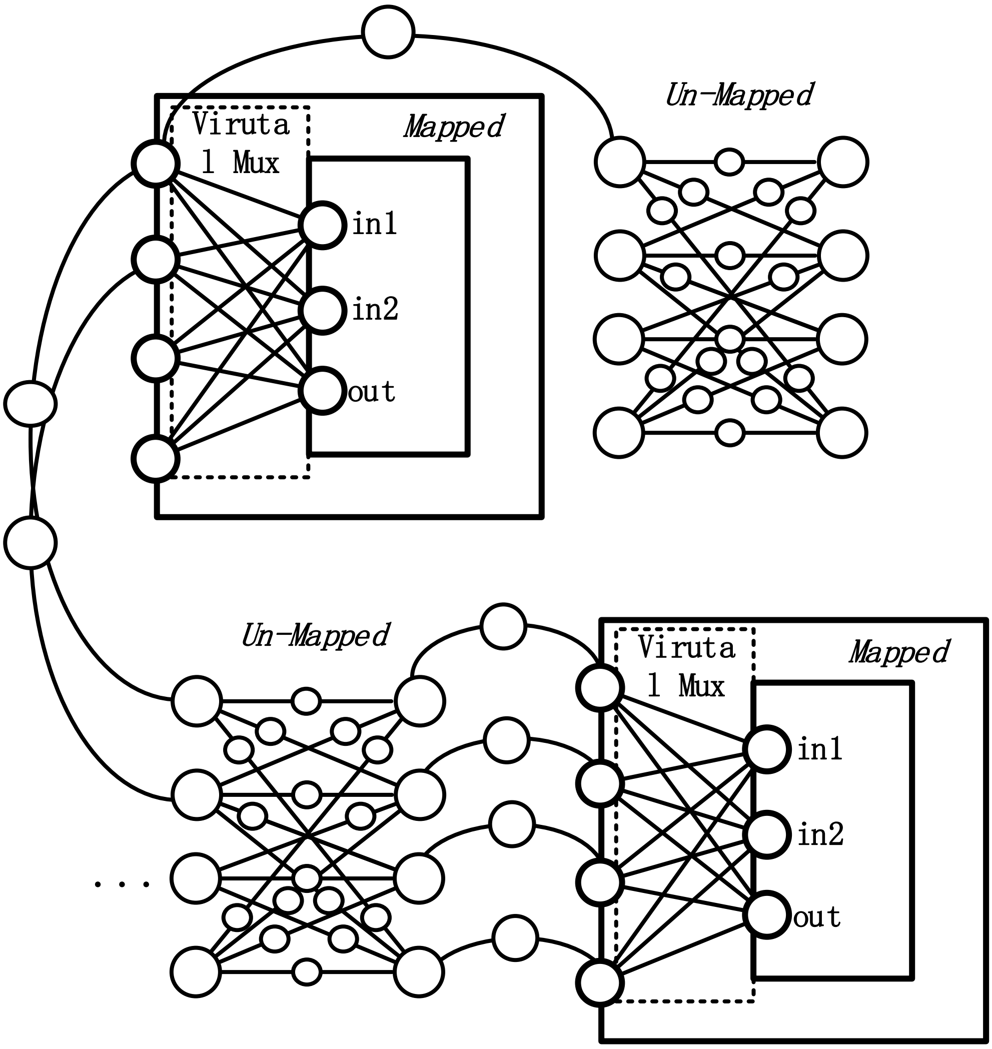


Figure 3.4: Partial Routing Graph of 4 Cells

3.5.2 Routing Algorithm

The routing algorithm scheme is described in Algorithm 2. Similar to other negotiation-based routing algorithm like pathfinder [ME95], the cost function of each node is calculated by Eq. (3.4), where C_b is the base cost, C_h is the historical cost, C_c is the current congestion

Algorithm 2 Routing Algorithm

```
while ( overused resources exist) { /* illegal routing*/
    annotateAllSlackRatio();
    sortNetsSlackRatio();
    foreach (net i) {
        ripUp(net i);
        foreach (sink j in net i) {
            pushRoutedResources();
            findShortestPath(sink j);
        }
        updateCongestionCost();
    }
    updateHistoricalCost();
}
```

cost and *sr* is the slack ratio of a wire. The base cost is a constant. Current congestion cost is calculated based on the current usage of the routing resource. The calculation of historical cost takes into account the historical congestion, in order to avoid swing between routing iterations. The slack ratio indicates the importance of the wire. The longer the wire, the more important it is and the less slack ratio it has. In calculating the cost, slack ratio plays trade-off between the base cost and other costs. The rationale behind is to route the important wires with shorter path while detour those less important ones.

$$Cost_n = (1 - sr) \cdot C_b + sr \cdot (C_b + C_h + C_c) \tag{3.4}$$

3.5.3 Routing Tune-up

To save routing runtime, we limit our routing exploration to bounding box and its vicinity. By limiting the exploration space, the router maintains a good quality routing result and save unnecessary exploration runtime.

Furthermore, we encourage reuse the partial tree when routing multi-fanout wire, because it saves routing resources. This is achieved by adding a push routed resources before the shortest path algorithm, ensuring the costs of routed resources are set to zero.

3.6 Configuration

Based on the placement and routing result, the final step is to generate the configuration to program the weight of qubits and interactions. This is achieved by first assigning Ising primitive configurations to each cell with logic qubits and interactions. Then, configuration of physical Ising model is generated by converting the logic cells to physical cells and combining the routing result. Figure 3.5 shows a configuration map of a majority voter.

3.7 Experiments

In this section, we report results of our experiments using the proposed QUBO mapping approach and tool flow. We chose the SAT problem testcases from SAT competition and logic circuit benchmark, which contain sophisticated Boolean network used in real applications.

Our flow starts with converting SAT testcases in CNF and circuit benchmarks to BLIF format, which is a Boolean network format used in academic world. We used ABC synthesis tool to optimize the converted BLIF ensuring that every logic operator is binary. Then we applied the proposed flow to map and embed the optimized Boolean network.

Our experiments were ran on a server with Xeon E3-1245L CPU and 8-Gigabyte memory. Figure 3.6 and 3.7 show the succeeded testcases embedded to the 12 x 12 cells architecture same to the capacity of the latest D-Wave 2X system. Some of these testcases are in the large circuit class in the MCNC benchmark suite showing that D-Wave has the potential in solving real world problems. The proposed technique is extremely efficient for the current generation device, as all testcases are finished in less than a second. Still, the capacity of qubits and their interactions limit the capability of D-Wave to solve larger problems.

To evaluate the scalability of the proposed approach, we used a hypothetical D-Wave

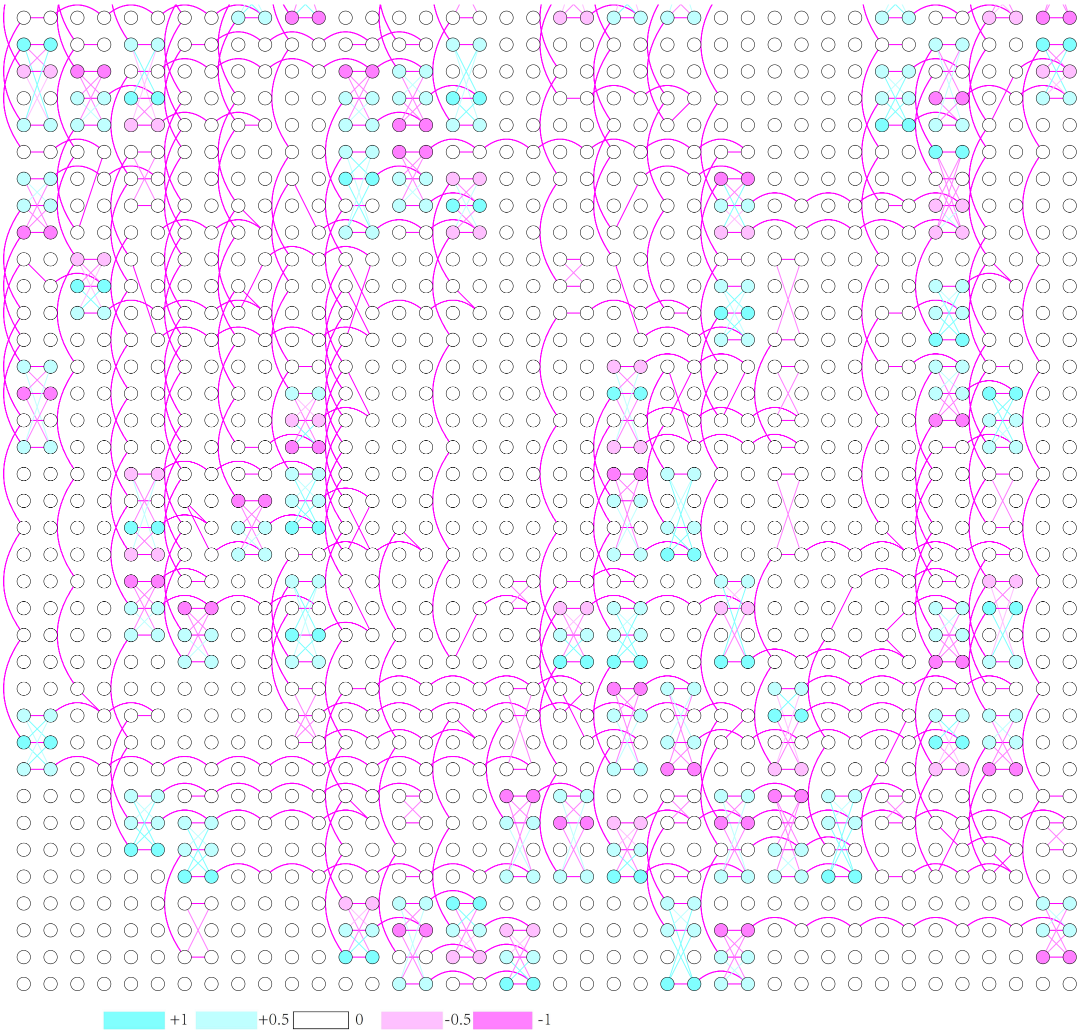


Figure 3.5: The Configuration of a Majority Voter

architecture annealer device with 100 x 100 cells, which is almost hundred times of the latest D-Wave system. We successfully managed to embed some hard SAT problems and complicated logic circuits into this device. The results presented in Table 3.1 are the 20 largest ones among all succeeded test cases. Run-time indicates the proposed technique scales well for future devices.

We separately analyzed the runtime increase caused by placement and routing. For placement, one necessary step is to update the cost, which is calculated based on the HPWL and the number of used cells within the bounding boxes, requiring information update from all the nets. Therefore, increase either in device size or the update frequency would increase

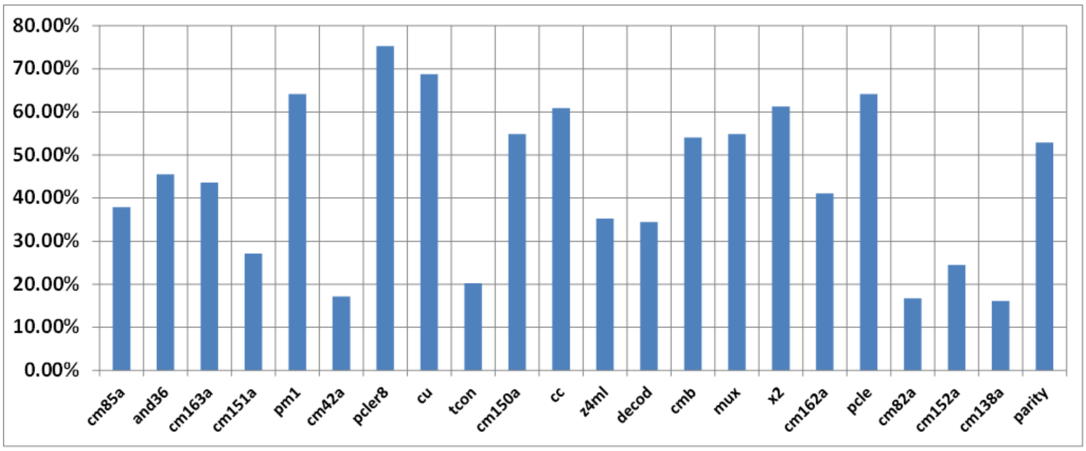


Figure 3.6: Utilization Rate on 12 x 12 Cells

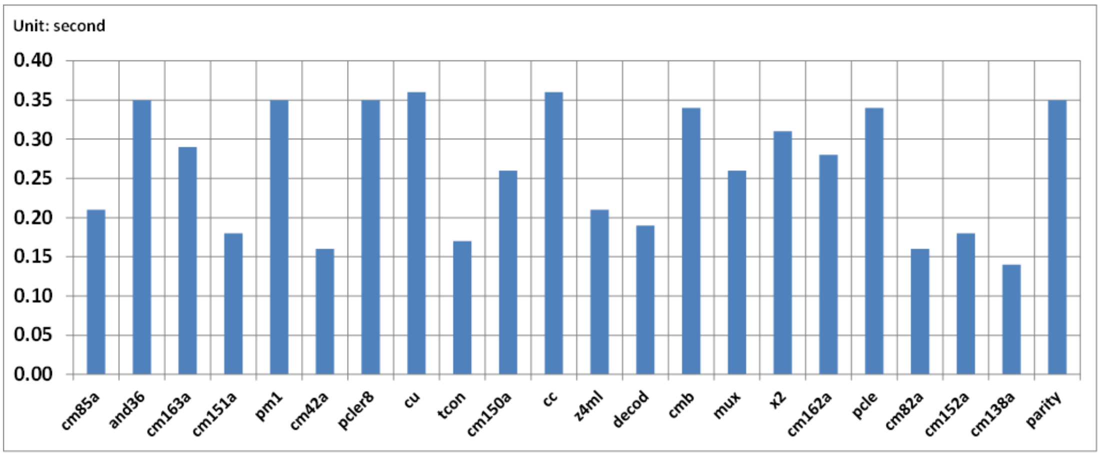


Figure 3.7: Runtime for 12 x 12 Cells

placement runtime. For routing procedure, as the device grows larger, the router tends to explore more nodes in a large routing graph. Moreover, for more complicated designs, the router needs more iterations to find valid path leading to cost inflation, further slowing down the router. Our solution to decrease the placement runtime is reduce the update frequency, while for routing, we can adjust exploration range and historical cost coefficient.

The success test cases also give us a glimpse of the problems can be solved by future devices. *sgen6* series testcases are originated from SAT Competition 2014 [BDH14], which are not solved in the SAT competition. Other than that, many applications such as equivalent

Table 3.1: Experiments on Architecture with 100 x 100 Cells

Name	Primitive#	Chain#	WireLength	CellUsage	InterUsage	TotalUsage	RunTime (s)
C6288	2370	2432	68366	11.88%	54.26%	66.14%	704.31
C5315	1775	2141	42438	8.89%	32.92%	41.82%	376.51
pair	1574	1918	37268	7.89%	28.91%	36.80%	344.62
dalu	1361	1509	41730	6.82%	33.29%	40.11%	309.39
frg2	1131	1421	28422	5.66%	22.09%	27.75%	202.16
C3540	1031	1129	31944	5.16%	25.51%	30.67%	222.75
i7	865	1261	13032	4.33%	9.40%	13.73%	119.66
sgen6-960-5-1	889	1207	37200	4.44%	29.69%	34.13%	2391.32
edges-072-3-7923777-13	782	1206	21366	3.75%	15.74%	19.49%	1077.08
x3	857	1125	18098	4.29%	13.77%	18.06%	106.61
edges-070-3-1250111-33	760	1172	20036	3.64%	14.71%	18.36%	1613.07
apex6	786	1054	15678	3.93%	11.79%	15.72%	98.5
sgen6-840-5-1	775	1053	28074	3.87%	22.24%	26.11%	1672.47
i9	736	910	17166	3.68%	13.24%	16.92%	92.56
i6	683	957	10744	3.42%	7.84%	11.26%	64.43
rot	661	931	11894	3.31%	8.69%	11.99%	77.79
too_large	726	800	22766	3.63%	18.21%	21.84%	99.86
alu4	717	743	26076	3.59%	21.07%	24.65%	186.44
edges-025-4-10062999-1-00	519	719	12106	2.48%	8.89%	11.37%	835.36
i2	418	818	4778	2.09%	2.97%	5.05%	36.23

checking and test generation can be derived from logic test cases. For these applications, people have been long searching for efficient methods providing high quality results. The current generation of D-Wave is able to return a valid result in millisecond level, [BRI14] also showed that quantum computer is much more scalable compare to classical computer, our experimental results further validating the great potential of quantum annealer to become an efficient tool in solving SAT related problems.

Overall, we have successfully embedded over 100 testcases, showing that our approach scales properly for the current generation and future devices.

3.8 Conclusions

We have presented a complete tool flow that first maps the Boolean SAT problem to QUBO problem and then embeds it into the practical quantum annealer. By leveraging and adapting the integrated circuit design automation techniques, we have successfully converted the embedding problem into a place-and-route problem. We have tested over 100 test cases that can be successfully implemented onto a D-Wave architecture quantum annealer in reasonable amount of time. Our approach also enables us to implement some hard SAT problems that there is no existing solver can solve. In conclusion, we greatly enhanced the SAT problem solving ability of quantum annealer.

CHAPTER 4

Fast Embedding with Chain Length Optimization

4.1 Chapter Introduction

The efficiency of embedding techniques is of great importance, as the capacity of the D-Wave quantum annealer exponentially increases. As discussed previously, chain has been introduced to solve embedding problem [PDD12][Cho08], which ensures the multiple qubits act as a single one during quantum annealing. It helps to reduce the degree of the vertex in the original QUBO, so that the QUBO can be embedded into the architecture with low connectivity. Minimizing chain length is also an important objective in developing embedding techniques, as the size of a chain significantly impacts the performance of the quantum annealer.

In this chapter, we discuss embedding optimization. The distinguished features in this work are summarized as follows:

- An incremental placement technique that employs winner tree and look up table data structure to speed up the cost calculation.
- An A* routing engine to accelerate the routing speed, based on the repeated pattern in the D-Wave topology.
- A new placement cost function that shortens the longest chain.
- A novel dynamic criticality technique to optimize the chain length and improve the routing convergence.
- A rip-up and reroute stage to further optimize the chain length.

To validate the effectiveness, we conducted two experiments. First, we embedded CSPs onto an hypothetical C100 D-Wave architecture which is over a hundred times larger than the latest available D-Wave annealer, experimental result shows that our approach achieves 3.4X speed-up with reduced longest chain size. In the second experiment, we embedded CSPs onto real D-Wave 2X quantum annealer, result suggests that our approach increases the ground state probability by 29% on average.

4.2 Challenges in Embedding

D-Wave annealer has a low connectivity. Its architecture is realized by arranging qubits according to the Chimera graph topology, as shown in Figure 3.1. Moreover, the performance of quantum annealer is affected by the embedding result. D-Wave 2X annealer performs computation by quantum annealing [DJA13], which is a non-deterministic procedure. Since it is impossible to guarantee that the annealer would find the ground state in each run, a straight forward strategy is to fire multiple runs and select the best from a set of candidate solution. Therefore, to assess the performance of quantum annealer, one of the most important standard is the probability to observe the ground state. The ground state probability depends on many factors, among which the longest chain size in the embedding result is a major one. It is very hard to formulate the kinks in long chains, as their dynamics may slow down and impede the development of alignment during the transition between the initial state and the final state[VMK15].

4.3 Embedding Flow

In the previous chapter, we discuss using place-and-route algorithm to solve embedding problem. [BCI16] proposed an iterative combined place-and-route flow. As compared to iterative flow, we decide to stick to the waterfall flow for the following reasons.

- The waterfall flow clearly segregates the duties of placement and routing, hence the quality of placement and routing can be accurately measured. Furthermore, the result of each step also helps to fine tune the algorithm.

- The combined flow is likely to introduce instability that leads to unpredictable result. This is because the placement affects the routing, and in turns, the routing also affects the quality of placement, leading to a chicken-and-the-egg problem.
- The waterfall flow supports the integration of the path finder [ME95], which is a very efficient algorithm in solving the field programmable gate array (FPGA) routing problem. It is reasonable to believe that it is also very efficient in routing the chains, given that the routing graph in D-Wave is much smaller than that of FPGA.

In the remaining of this chapter, we discuss details of the proposed optimization techniques used in placement and routing.

4.3.1 Placement Improvements

The quality of the placement greatly affects the subsequent routing quality. Based on our analysis in the previous sections, there are three objectives in the placement. (1) minimizing the total wire length, (2) minimizing those long chains that affects the ground state probability, (3) leaving enough space for routing. In comparison with previous chapters, the improvements in placement include a comprehensive placement cost function to achieve three optimization objectives, along with, dedicated date structures to improve the computational efficiency.

4.3.1.1 Placement Cost Function

The quality of placement result depends heavily on the cost function. We propose the following cost function to factor in the chain length minimization, as shown in Eq. (4.1).

$$Cost = \sum_{i \in Chains} q(i) \cdot [\frac{bb_x(i)}{(S(i))^m} + \frac{bb_y(i)}{(S(i))^m}] \cdot Crit(i) \tag{4.1}$$

where $Crit(i) = \sqrt{Chain_i/Chain_{longest}}$, indicating the criticality of each chain. The longer the chain, the higher its criticality. We take the square root of the length ratio to emphasize the importance of those chains that close to the longest chain, as the ratio is always smaller than one. $S(i)$ is the routing supply in vertical(horizontal) direction, and $q(i)$ is the coefficient for multi-fanout chains.

In D-Wave embedding flow, the routability is the key to the success of the subsequent routing stages. Unlike placement in circuit design, the routing resources per gate Ising model are very limited, and the situation could be even worse if multiple gate Ising models are placed in the same area as the Ising gate itself will consume some of routing resource. In the proposed cost function, we model the routing resource as the number of vertexes and edges that are available within the chain bounding box. m is a positive number to adjust the influence of the routing supply in the cost function and it can be fine tuned by experiment.

4.3.1.2 Incremental Cost Update

Apparently, an efficient way to compute the cost of the placement should considerably reduce the run-time in the placement. Based on the incremental bounding box update proposed by[LKJ11], we extend the incremental computation to routing supply and chain criticality.

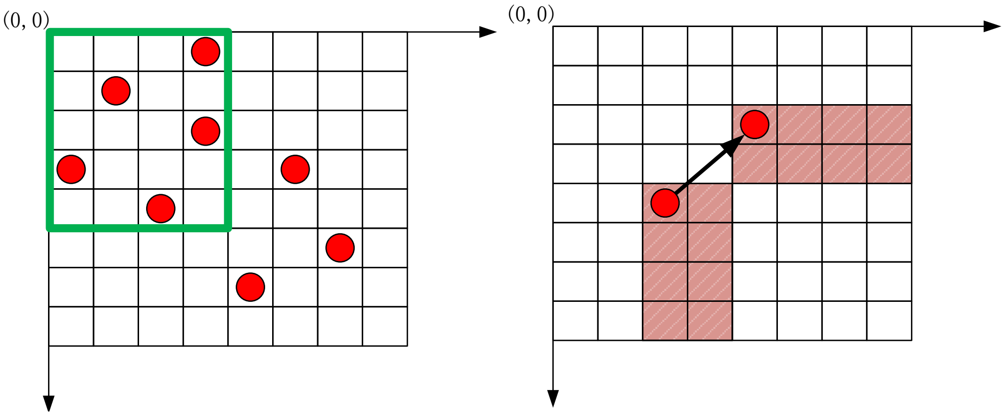


Figure 4.1: Routing Supply Update

We use a two-dimensional array to store the available routing resource with given placement. It is straight forward to use cells in the chimera graph to formulate the bins in the grid graph. We store the number of available routing resources in the top left area of $\{(0, 0), (i, j)\}$ in $S(i, j)$, so that we can calculate the available routing resources in any bounding box that expands its area from (x_a, y_a) to (x_b, y_b) using the Eq (4.2). To incrementally update the available routing resource, we only need to update the $S(i, j)$ in the shading area, after the placement move is generated, as shown in Figure 4.1.

$$S = S(x_b, y_b) - S(x_a, y_b) - S(x_b, y_a) + S(x_a, y_b) \tag{4.2}$$

For the chain criticality, winner tree is used to support incremental update as shown in Figure 4.2. The bottom layer stores the length of each chain, while each upper layer stores the winners from the lower layer. By doing this, we can update the winner caused by a single-chain-length change with complexity of $logN$, where N is the number of chains.

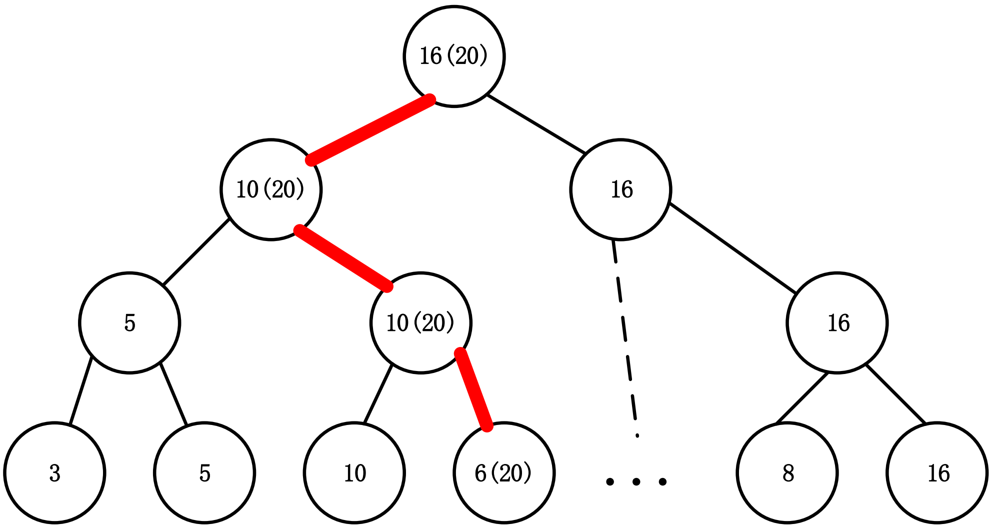


Figure 4.2: Winner Tree Chain Size Update

4.3.2 Chain Routing

Chapter 2 describes a chain routing algorithm based on pathfinder. As an improvement of previous work, we further propose innovative methods that improve the run-time and minimize the chain length. In the remaining of this section, we first discuss the run-time speed-up techniques, and then present a rip-up and re-route algorithm that aggressively minimizes the longest chain length.

4.3.2.1 Routing Speed-up

We use A* to accelerate the shortest path algorithm, which employs a heuristic cost to guide the order of vertex wave expansion in Dijkstra shortest path algorithm. Here we use the shortest path length as the A* heuristic cost, as it can be conveniently calculated based on the periodic pattern in the D-Wave Chimera graph. Eq. (4.3) shows the shortest path length calculation of any given vertex pair. Figure 4.3 shows an example of the shortest path with given source and destination vertex pair.

$$ShortestPath(v_1, v_2) = |v_{1x} - v_{2x}| + |v_{1y} - v_{2y}| + k \tag{4.3}$$

where x,y indicate the cell coordinates and k is used to add adjustment length if the v_1, v_2 are not sitting in the counterpart location of two different cells.

4.3.2.2 Dynamic Criticality

We propose the dynamic criticality D_{cr}-based negotiation routing. Eq. (4.4)-(4.6) show the cost functions for each vertex. The terminology is similar to the pathfinder: C_b is the base cost, C_s is the congestion cost or share cost, and C_h is the history cost and P_i is the penalty factor that increases with the routing iterations. Unlike static criticality in pathfinder, we propose dynamic criticality to reduce the chain length and to accelerate routing convergence.

$$C_v = Dcr * C_b + (1 - Dcr) * (C_b + C_s * P_i + C_h) \tag{4.4}$$

$$L_{pred} = (L_c + L_d) \tag{4.5}$$

$$Dcr = 1 - \frac{L_{pred}}{L_{longest}} \tag{4.6}$$

where L_c is the chain length from the source vertex to the current wave front vertex, L_d is the shortest length from the current wave front vertex to the destination vertex, as shown in Figure 4.4. The $L_{longest}$ can be seen as the length budget to route the current chain, since the routing result will not get worse if the L_c is shorter than the previous longest.

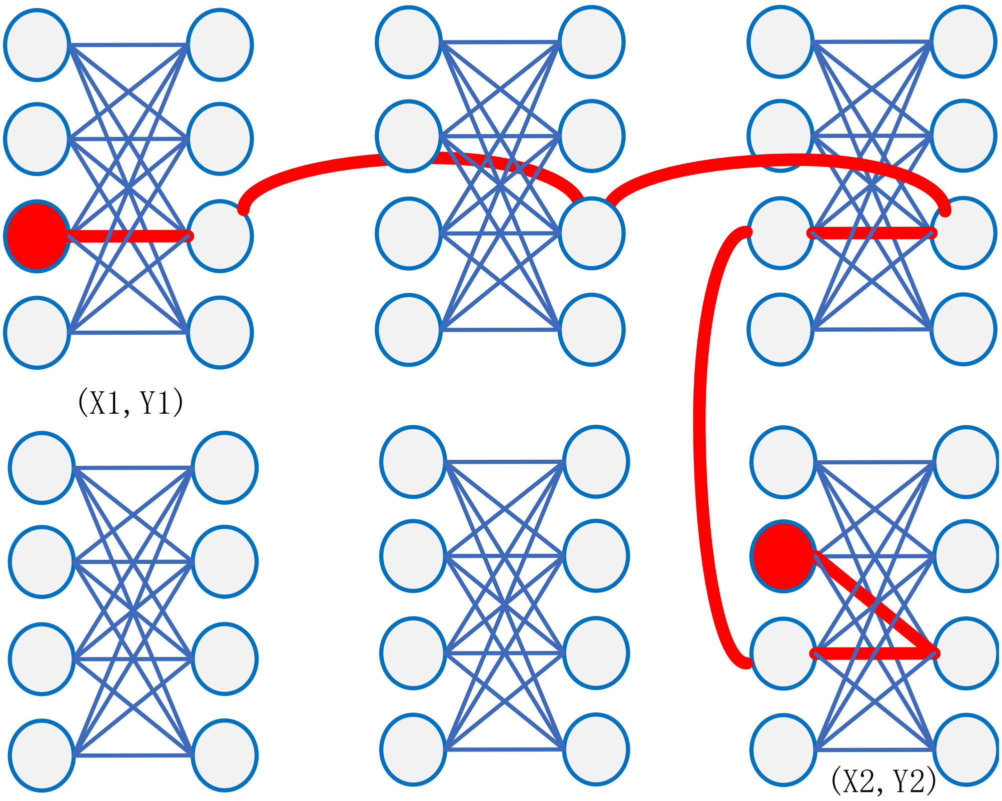


Figure 4.3: The Shortest Path

Compared to [LKJ11], we also modify the main routing cost function as the summation of the costs in order to balance their magnitude, as shown in Eq. (4.4). The negotiation happens between two independent chains trying to use the same vertex, and the criticality eventually decides which chain takes the vertex and which chain should be detoured.

A bad routing result is often caused by the under estimation of the chain criticality. For example, a chain has low criticality in the last iteration will detour significantly in the new iteration. However, in the new iteration it may become the longest chain because of that.

Dynamic criticality uses prediction length to project its future criticality hence limits the chance of finding a worse result. At the beginning of the wave front propagation, the prediction length is short because the shortest path length contributes to the most of the path. As the wave keep propagating, the prediction length may increase due to the detour, and consequently the D_{cr} will also increase. the increased D_{cr} will encourage the router to use the short path rather than detour to reduce the congestion.

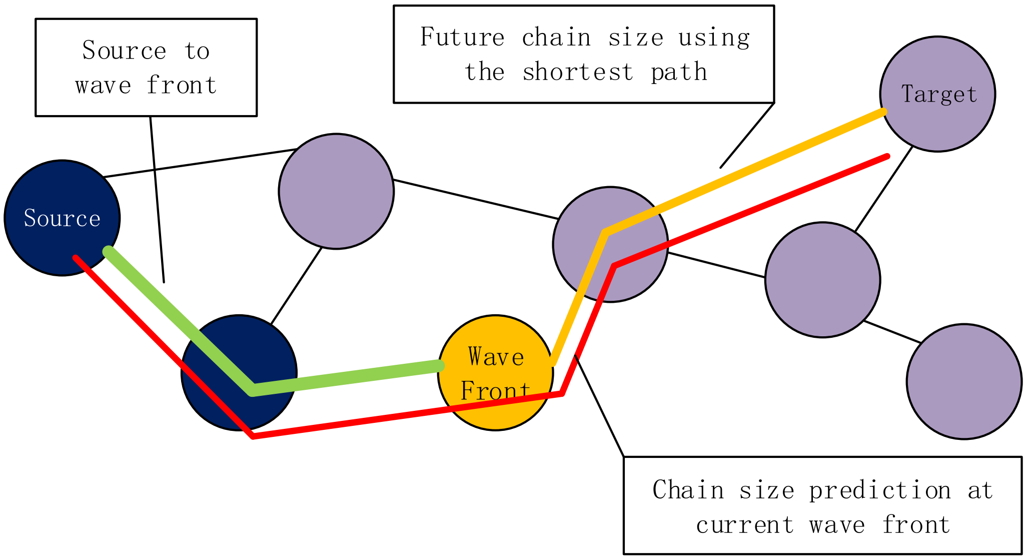

Source to wave front
Future chain size using the shortest path
Target
Source
Wave Front
Chain size prediction at current wave front

Figure 4.4: Chain Size Prediction

4.3.3 Rip-up and Reroute

The negotiation based routing dose not necessarily guarantee the optimal routing result. At the first several iterations, the vertexes in the routing graph is allowed to be shared by different chains. Then, the overflow is gradually solved by the increasing $C_s * P_i$. Ideally, the critical chains will take the shortest path, while others will be detoured. However, if P_i is not properly designed, the routing may either too slow (P_i is too small) or the negotiation is not sufficient (P_i is too large). Both scenarios will leave the routing result sub-optimal.

To further improve the routing quality, we propose a rip-up and reroute stage, as described in Figure 4.5. Because the best routing solution is often hard to find, instead of trying to find the best solution, our strategy is to set a goal and then to find a solution that meets the goal.

The rip-up and reroute starts from a valid routing result, which means that there is no vertex or edge shared by different chains. Then we set a limit on the longest chain length in the next iteration based on the longest chain length in the current iteration. Then we perform a similar negotiation-based routing to reroute some of the chains. A chain will be rerouted if meets the following criteria: (1) the chain uses a overflow vertexes or edges shared by other chains. (2) the chain violates the limit of the longest length.

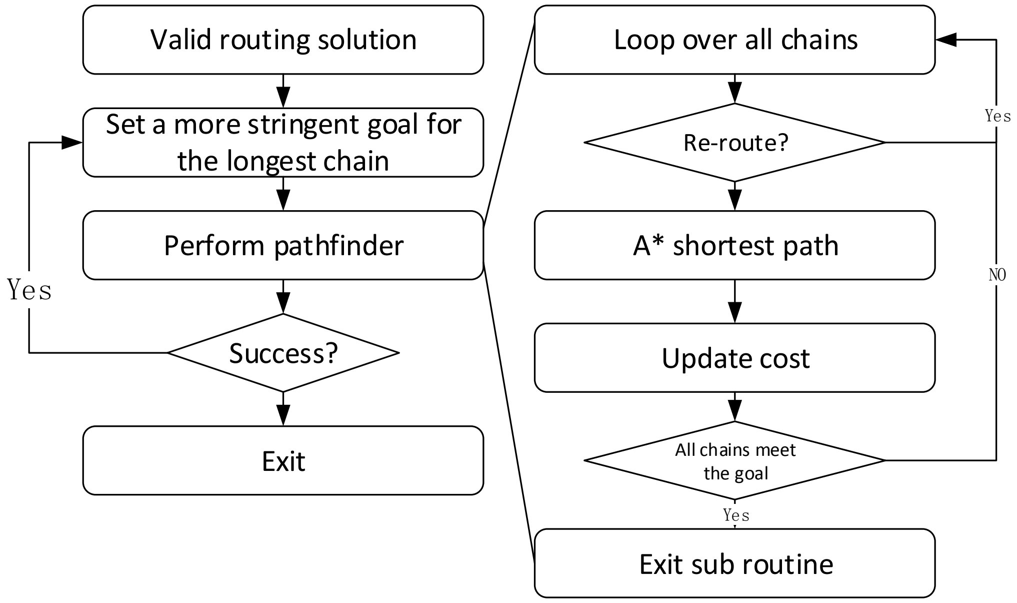


Figure 4.5: Rip-up and Reroute Flow

At the beginning of the rip-up and reroute stage, there is no overflow vertexes or edges, so only those chains that violate the limit will be rerouted, and then some of the vertexes and edges will become overflow because of the rerouting. In the following iterations, the overflow will be gradually resolved as the penalty factor increases. Once a new result that has no overflow and do not violate the chain length limit is found, the negotiation routing will stop. Then, a more stringent limit will be set based on the newly found longest chain length and the procedures will be repeated. The stop condition for the rip-up and reroute is either the routing hits the iteration bounds or there is no room to improve as all the long chains are already taking the shortest path.

4.4 Experiments

Test cases are selected from MCNC benchmark suite. We implemented the proposed techniques in C++, and ran it over a Xeon-E5 2680 linux server with 64GB memory. As even the latest D-Wave device is too small, to better present the benefit of the proposed technique, we first conducted experiments by embedding the Boolean network onto a hypothetical C100 Chimera architecture and compare it against the result using technique described in Chapter

3, which is denoted as QSAT. Secondly, we embedded the test cases to the real D-Wave 2x quantum annealer, using the proposed techniques in the placement and routing.

Table 4.1: Embedding Experiment on D-Wave 2X

Design	The Longest Chain Size		Improvement	
	QSAT	FastEmbedding	TheLongestChain	P_{GS}
sct	52	36	30.77%	33.41%
b9	26	19	26.92%	39.39%
cordic	37	27	27.03%	27.72%
pcler8	58	41	29.31%	39.11%
parity	29	20	31.03%	22.96%
pcle	46	34	26.09%	21.97%
cc	62	44	29.03%	27.67%
cu	32	24	25.00%	31.24%
cm85a	30	23	23.33%	26.62%
x2	23	17	26.09%	28.33%
Average			27.46%	29.84%

4.4.1 C100 Chimera

In the first experiment, we embedded the Boolean network onto a C100 Chimera architecture, which comprises 100 X 100 cells. To accurately analyze and identify the effectiveness of our approach, we present the experimental result from placement and routing, respectively.

We compare the placement result in Table 4.2. The input of CSP is an netlist that represent a set of locally embedded constraints and their connections. By using our method, the placement run-time is improved by 1.8X on average along with reduction in the largest half perimeter wire length (HPWL). The result suggests our approach is much more efficient, despite an additional term in the cost function. This is achieved because our approach works in an incremental manner so that unnecessary and repetitive cost re-computation can be avoided.

Table 4.2: Placement Experiments on Architecture with 100 x 100 Cells

Design	Placement					
	QSAT		FastEmbedding		Improvement	
	PlaceTime(S)	Max HWPL	PlaceTime(S)	Max HWPL	PlaceTime	Max HWPL
C6288	300.5	396	190.6	280	1.6X	29%
C5315	178.9	326	131.3	310	1.4X	5%
pair	164.1	294	99.1	213	1.7X	27%
dalu	118.3	149	77.1	125	1.5X	16%
frg2	83.8	98	54.5	81	1.5X	17%
C3540	73.3	162	50.3	160	1.5X	1%
i7	63.4	60	39.8	57	1.6X	5%
x3	49.2	91	37.7	81	1.3X	11%
apex6	48.4	142	35.5	120	1.4X	15%
i9	41.9	120	27.0	112	1.6X	7%
Average					1.5X	13.4%

We also compare the routing results in Table 4.3. For fair comparison, we started with the same placement result generated by the proposed placement algorithm, and compared routing using both our proposed techniques and the technique used in QSAT. Experimental results show that our approach greatly reduced the run-time by 1.8x to 3.3x and meanwhile is very effective in minimizing the longest chain length by 19% on average. We also notice that the result varies possibly because of the connection is very different in the original problem.

4.4.2 Ground State Probability Improvement

We report the performance improvement by embedding the Boolean network onto the D-Wave 2x quantum annealer in Table 4.1. For each test case, we repeat the quantum annealing 10,000 times and gauge the state of the qubit. As the ground state energy can be calculated beforehand, using the summation of the lowest energy of each constraint and chain, we can find the probability of the ground state by counting the number of gauge that achieves the

Table 4.3: Routing Experiments on Architecture with 100 x 100 Cells

Design	Routing					
	QSAT		FastEmbedding		Improvement	
	LongestChain	RouteTime(S)	LongestChain	RouteTime(S)	LongestChain	RouteTime
C6288	809	140.0	625	79.8	23%	1.8X
C5315	201	56.5	171	17.1	15%	3.3X
pair	495	51.2	364	27.4	26%	1.9X
dalu	447	75.0	436	26.3	3%	2.9X
frg2	326	42.4	261	21.0	20%	2.0X
C3540	273	65.8	221	24.0	19%	2.7X
i7	169	11.1	129	5.4	24%	2.1X
x3	137	17.2	130	8.4	5%	2.1X
apex6	220	13.0	157	5.6	29%	2.3X
i9	30	15.9	22	7.1	27%	2.2X
Average					19%	2.3X

lowest energy. According to the experimental result, our approach improves the ground state probability by 29% on average.

4.5 Conclusions

EDA has witnessed a great success in assisting circuit design. In this work, we demonstrate an excellent example of using EDA techniques and philosophies to improve the performance of the D-Wave quantum annealer. As an emerging technology and a new computing paradigm, the quantum annealing is faced with many practical limitations. We see that there is a strong demand to bring EDA techniques into the quantum computing area to address those challenges, as many EDA techniques are known to be effective in performing optimization under the presence of practical constraints.

CHAPTER 5

Post Embedding Optimization

5.1 Chapter Introduction

Quantum annealing guarantees the finding of ground state in the final Hamiltonian if the system evolves sufficiently slowly, yet errors are still observed when experimenting with the practical quantum annealer [DJA13][LAB14]. The errors can be attributed to the following reasons. Firstly, the required annealing time that ensures the system evolves sufficiently slowly is still elusive[CFP01][FGG00][SL05]. As suggested by the research[CFP01], the minimal required annealing time is related to the energy gap Δ between the ground state and the first excited state. Specifically, the minimal required annealing time is inversely proportional to the Δ^2. Therefore, during quantum annealing, a small energy gap may result in computational error, which manifests as the appearance of non-ground state. Secondly, practical quantum annealer also suffers intrinsic control error (ICE), which describes a scenario that instead of solving the given Hamiltonian H, the practical quantum annelaer actually solves a problem with slightly altered Hamiltonian[NTK15][JAG11][KM14]. The deviation from the original problem may lead to erroneous. In both cases, enlarging the Δ plays a key role to make sure quantum annealer produces correct result with high fidelity[YSB13][TCT17].

In this chapter, we present a post embedding optimization technique to improve the ground state probability for satisfiability problem. Unlike other generic embedding and error correction methods, which aim at improve embedding quality for all problems, our method is exclusively tailored for satisfiability problem. Here, we summarize the advantages of the proposed embedding technique as follows.

- Optimal energy gap for each Ising instance in a satisfiability problem. Instead of enlarging the energy gap for the entire problem, our technique focuses only on a small portion of the problem. We employ Boolean analysis and further proposed a MIQCP formulation to enlarge energy gap for each Ising gate and Ising chain.

- No impact on embedding. Our technique does not affect existing embedding, it works as a post-processing step following regular embedding. In fact, the proposed technique utilizes the remaining available qubits as auxiliary qubits; hence, it will not introduce any embedding overhead.
- Topology independence. The proposed method can be easily extended to different hardware topology. In this paper, we use D-Wave architecture to demonstrate the proposed technique.

We performed our tests on a D-Wave 2000Q device, which is the largest quantum annealer in the market containing 2048 qubits. Instead of using toy Ising models, our test cases are originated from logic circuit design and SAT problem in CNF format, through which we can examine the performance of quantum annealer in dealing with real world applications. The experimental result suggests that our technique significantly improves the ground state probability. In addition, as a side product, we also empirically demonstrate the existence of optimal relative strength r_s, which keeps a balance between the logic gate Hamiltonian and chain Hamiltonian.

5.2 Improving Ground State Probability

Quantum annealers are susceptible to errors: the result returned by a quantum annealer is not always the ground state. Therefore, methods to improve the ground state probability should be carefully designed in quantum annealing algorithm. Additionally, improving ground state probability also means improving the performance of quantum annealer. ST99 is a performance metrics, which indicates the number of quantum annealing runs needed to identify the ground state with 99% probability. The equation of ST99 is shown in Eq. (5.1).

$$ST99 = \frac{log(1-0.99)}{log(1-P)} \tag{5.1}$$

where P is the ground state probability for a single run. Apparently, high ground state probability P will lead to smaller number of runs to achieve $ST99$.

5.2.1 Auxiliary Qubit

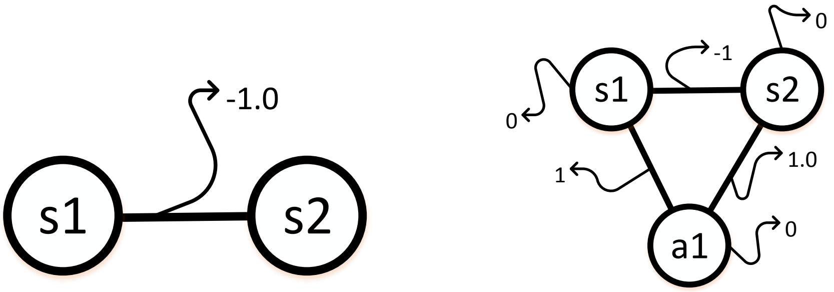
In the previous Chapters, we discussed the embedding of satisfiability problem which consists a logic gate network. After embedding, we notice that in many cases there are many unused qubits and couplers, can we use them to improve the the ground state probability?

In fact, numerous researches have proposed solutions to improve fault tolerance in gate model quantum computing by introducing auxiliary qubits, which are not involved in computation but make the system more robust[DS96][AGP06]. However, a few researches [YSB13][QL12] have discussed fault tolerance for quantum annealing by introducing auxiliary qubits. [PAL14] proposed adiabatic quantum correction for D-Wave device at the cost of sacrificing the precious qubit and connectivity resource. [BCI14] also proposed a method to enlarge the energy gap, however, it suffers long run-time to find the solution. In this section, we propose a post-processing embedding technique to improve the ground state probability by enlarging the energy gap, featuring high efficiency without embedding overhead.

In quantum annealing, auxiliary qubit does not encode the problem solution, but it helps to improve the energy gap Δ. We use a two-qubit chain to explain how auxiliary qubit aids in enlarging the energy gap, as shown in Figure 5.1.

By comparing Table 5.1 and Table 5.2, the energy gap in the 3-qubit system is 4 which is twice as the energy gap in the 2-qubit system. More importantly, the 3-qubit system maintains the property that the system energy is minimized if and only if s_1 and s_2 take the same value.

The toy example shows that the auxiliary qubit does improve energy gap while preserving



(a) without auxiliary qubit (b) with auxiliary qubit

Figure 5.1: Auxiliary Qubit in Embedding

Table 5.1: Energy Landscape for Two-qubit Chain

s1	s2	Energy
-1	-1	-1
-1	1	1
1	-1	1
1	1	-1

logic property. It immediately lead to a question, that is, how to systematically introduce auxiliary qubit. [BCI14] pointed out that auxiliary qubit is a function of all problem qubits. Therefore, the problem to find optimal energy gap under the presence of auxiliary qubits can be formulated as follows

$$\begin{array}{llcll} \max & g & & & \\ \text{s.t.} & H_G(\boldsymbol{s},\boldsymbol{a}) & \geq & k & ifs \in C \\ & H_G(\boldsymbol{s},\boldsymbol{a}) & = & k+g & ifs \notin C \\ & \boldsymbol{a} & = & f(\boldsymbol{s}) & f:\{-1,1\}^n \rightarrow -1,1 \\ & -1 \leq & J_{ij} & \leq 1 & \\ & -2 \leq & h_i & \leq 2 & \\ & g & > & 0 & \end{array} \tag{5.2}$$

Table 5.2: Energy Landscape with Auxiliary Qubit

s1	s2	a1	Energy
-1	-1	-1	1
-1	-1	1	-3
-1	1	-1	1
-1	1	1	1
1	-1	-1	1
1	-1	1	1
1	1	-1	-3
1	1	1	1

where $\boldsymbol{a}, \boldsymbol{s}$ are vectors, indicating the auxiliary qubits and problem qubits respectively.

5.2.2 MIQCP Formulation

We employ Boolean analysis to formulate the problem described in Eq. (5.2). According to [OD14], the Boolean function $f : \{-1, 1\}^n \rightarrow -1, 1$ can be written as a real multlinear polynomial. For example, an f_{OR} operation maps $\{(-1, -1), (-1, 1), (1, -1), (1, 1)\}$ to $\{(-1), (1), (1), (1)\}$ can be expressed as Eq. (5.3)

$$f_{OR}(s_1, s_2) = \frac{1}{2} + \frac{1}{2}s_1 + \frac{1}{2}s_2 - \frac{1}{2}s_1s_2 \tag{5.3}$$

To generalize the expression for any given logic, we use the 2-input truth table below as an example.

where $v0, v1, v2, v3 \in \{0, 1\}$. Then, the multilinear polynomial can be written as Eq. (5.4)

Table 5.3: Truth Table for 2-Input Gate

S1	S2	Value
-1	-1	v0
-1	-1	v1
-1	1	v2
-1	1	v3

$$\begin{aligned} f(s_1, s_2) &= v_0(\tfrac{1-s_1}{2})(\tfrac{1-s_2}{2}) \\ &+ v_1(\tfrac{1-s_1}{2})(\tfrac{1+s_2}{2}) \\ &+ v_2(\tfrac{1+s_1}{2})(\tfrac{1-s_2}{2}) \\ &+ v_3(\tfrac{1+s_1}{2})(\tfrac{1+s_2}{2}) \end{aligned} \tag{5.4}$$

Thus, we can replace the logic function for auxiliary qubits with $a = f(\boldsymbol{s}, \boldsymbol{v})$, and re-write Eq. (5.2) as Eq. (5.5)

$$\begin{array}{llcll} \max & g & & & \\ \text{s.t.} & H_G(\boldsymbol{s}, \boldsymbol{a}) & = & k & if s \in C \\ & H_G(\boldsymbol{s}, \boldsymbol{a}) & = & k+g & if s \notin C \\ & \boldsymbol{a} & = & f(\boldsymbol{s}, \boldsymbol{v}) & f : \{-1,1\}^n \to -1, 1 \\ & \boldsymbol{v} & \in & \{0,1\} & \\ & -1 \leq & J_{ij} & \leq 1 & \\ & -2 \leq & h_i & \leq 2 & \\ & g & > & 0 & \end{array} \tag{5.5}$$

Assuming we have a 3-qubit system with full connectivity, the system Hamiltonian is shown in Eq. (3.1). We use s_3 as the auxiliary qubit, and we let s_1 and s_2 to take 1. Then, we write the $H_G(1, 1, a(1, 1))$ as Eq. (5.6) and (5.7).

$$\begin{gathered} H_G(1, 1, a(1, 1, (v))) = h_1 + h_2 + h_3 a(1, 1, \boldsymbol{v}) + J_{12} \\ + J_{13} a(1, 1, \boldsymbol{v}) + J_{23} a(1, 1, \boldsymbol{v}) \end{gathered} \tag{5.6}$$

$$\begin{aligned} H_G(1,1,a(1,1)) &= h_1 + h_2 + h_3 \cdot v_3 \\ &+ J_{12} + J_{13}v_3 + J_{23}v_3 \end{aligned} \tag{5.7}$$

Based on Eq. (5.7), the original programming problem is essentially a mixed integer quadratic constraint programming (MIQCP). Many commercialized optimizers, such as [Ibm11] and [Gur16], can be used to solve MIQCP problem. As an alternative, the MIQCP problem can also be solved by feasibility solver combined with binary search. For example, we can cast the MIQCP problem into a satisfiable module theory problem (SMT) [BBH09] given a target energy gap g, and we examine the feasibility with given target energy gap g. Next, we use binary search to narrow down the optimal energy gap.

The MIQCP formulation requires 2^n equality constraints to penalize the energy for states out of C, where n is the number of problem variable. Additionally, each auxiliary qubit will introduce a equality constraint. The total number of variables are $n + n_a \cdot 2^n$, where n_a is the number of auxiliary qubit.

5.2.3 Post Processing Embedding Optimization

In fact, MIQCP problem is a NP-hard problem. Thus, it cannot be used to solve large problems. Therefore, we propose a technique that only focus on improving energy gap for small instances rather than optimizing the entire problem. We propose a post embedding optimization step follows the initial embedding solution, hence avoiding any embedding overhead.

Figure 5.2 depicts a typical embedding solution of a AND Ising gate model. The $K_{4,4}$ cell uses 6 qubits to implement a 2-input Ising gate model, which seems quite wasteful as a 3-qubit system with full connectivity is able to achieve the same effect. There are two reasons for using 6 qubits. Firstly, it is the most straight forward way to embed the K_3 graph, using a K_4 graph with 4 logic qubits. Secondly, the logic qubit cell structure ease the routing problem. This is because the $K_{4,4}$ graph is symmetric in terms of connectivity. Therefore, we can use virtual MUX to resolve the qubit assignment problem. Without

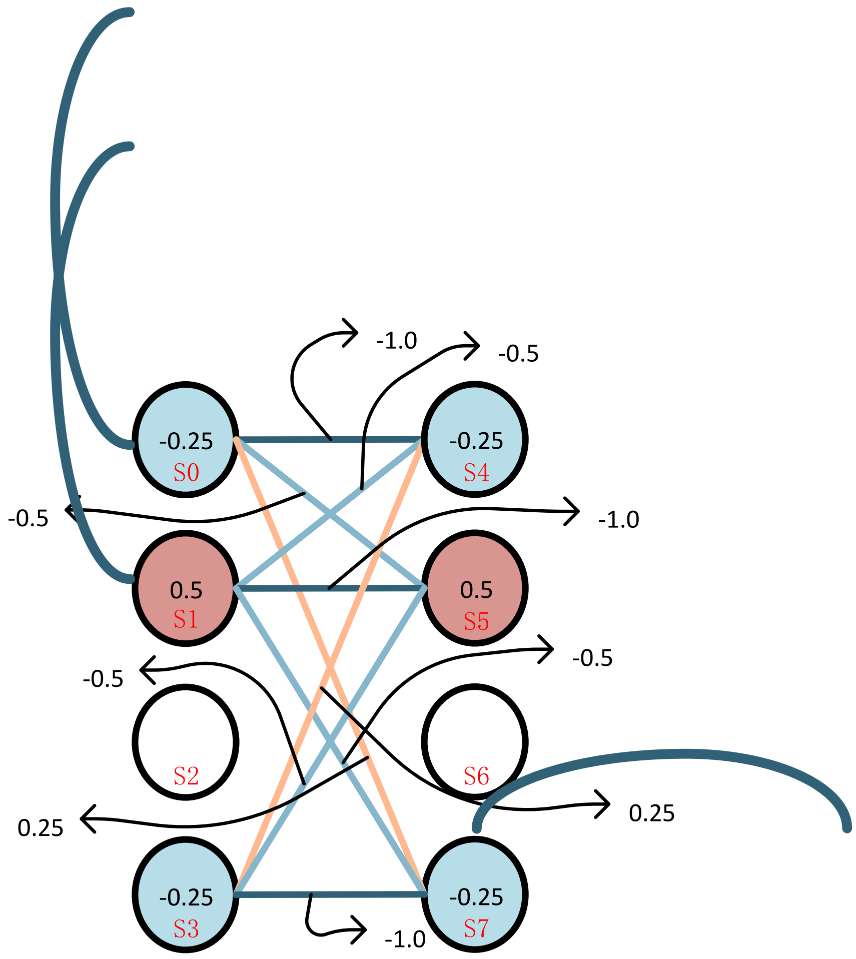


Figure 5.2: AND gate with unused qubits

symmetric property, an additional step is required to determine which qubit to implement the variable in an Ising gate model.

Figure 3.4 depicts a routing graph of 2 x 2 cells. For the cells where Ising gate is implemented, a virtual MUX is introduced so that the qubit assignment problem can be resolved using the routing result.

Based on the observations of MIQCP formulation and existing embedding technique, we discuss the post-processing technique for Ising gate and Ising chain separately.

5.2.3.1 Ising gate

We can use the MIQCP formulation to enlarge the energy gap of an embedded Ising gate. To reduce the problem size, we only identify auxiliary qubits in the same cell to enlarge the energy gap.

After embedding, We still notice unused qubits and couplers, as shown in Figure 5.2. S_2 and S_6 can be used as auxiliary qubit. We can further identify more auxiliary qubit by combining chain routing result. As shown in the Figure 5.2 the S_4 is not used to connect other qubit sitting in different cell. Therefore, we can use it as auxiliary qubit.

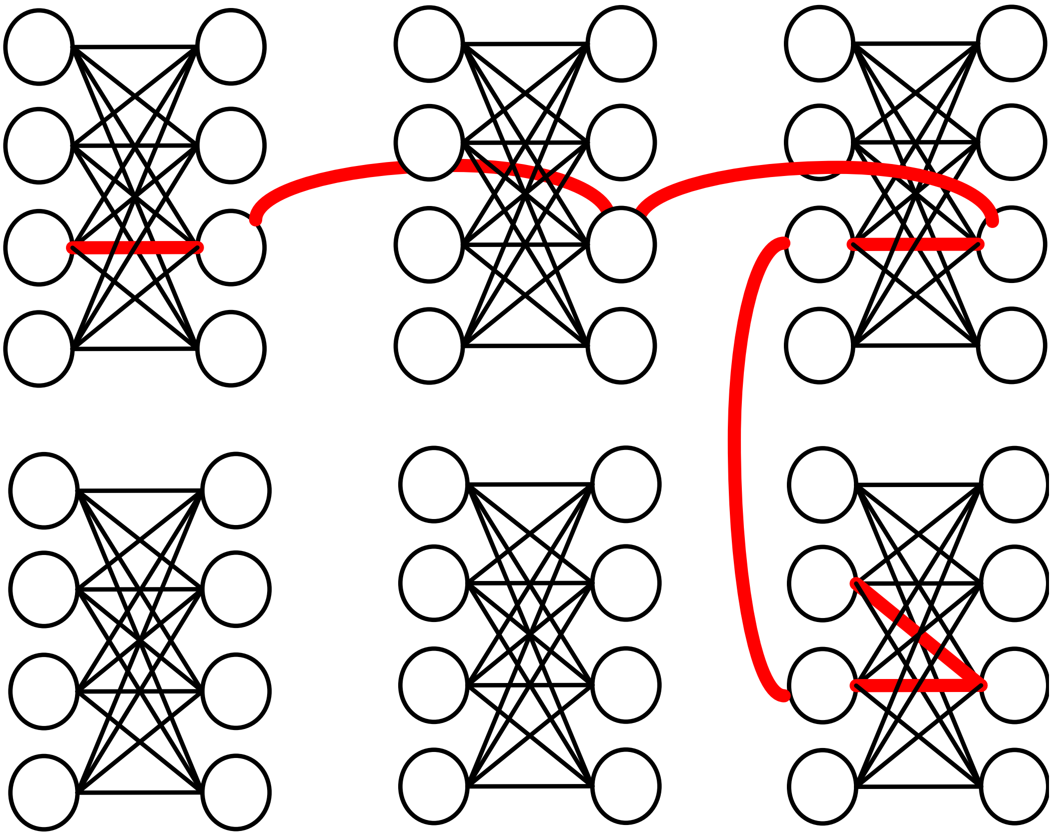


Figure 5.3: Chain Embedding on Chimera Graph

5.2.3.2 Ising chain

Interconnected chains are used to ensure that every qubits to take the same state at the end of quantum annealing. A typical embedding result of a chain is shown in Figure 5.3. It consists many couplers with -1 strength. The difficulty in improving the energy gap is that the most fragile part is in every qubit. Therefore, to enlarge the energy gap for entire chain requires strengthening every qubits. Moreover, chains often consist of a large number of qubit, making the MIQCP problem hard to solve. To overcome those difficulties, in stead of enlarging the energy gap for entire chains, our strategy is to break chains into small instances and only enlarge the energy gap for those Small instances. Although we may always enlarge the energy gap for entire chain, the embedding of chains are still improved as we break long chains into smaller ones which are less likely to suffer early freeze-out[VMK15].

For D-Wave architecture, a coupler in the chain can be categorized as intra-cell coupler and inter-cell coupler. And a single chain consists of those couplers whose energy gap can be enlarged one by one. For intra-cell coupler, the energy gap is boosted by using available qubits in the same cell, similar to that of Ising gate. Sometimes, a single cell may contain two or more chains, in which cases the set of interest C can be formulated accordingly, and the MIQCP problem is solved in one shot. For inter-cell chain, the available qubits and couplers around them are much less than inter-cell couplers. Therefore, we need to first identify the possible embedding pattern. Figure 5.4 shows a possible embedding pattern for two horizontally adjacent cells to strengthen the inter-cell chain. The bold line in the figure is the inter-cell chain, and the dashed line indicates a possible embedding pattern. To enlarge the energy gap for inter-cell, we enumerate all possible embedding patterns until sufficient auxiliary qubits are identified.

To enlarge the energy gap for the entire chain, every coupler has to be strengthened, which are hard to achieve in many cases due to limited resources. Our priority is to optimize long chains first and then we will use the remaining resources to optimize all short chains, leading to an embedding solution that only contains short chains.

5.2.4 Variable Elimination

We can accelerate the MIQCP solving time by eliminating redundant variables. Again, using Figure 5.2 as an example, we can eliminate the input number of assisting function

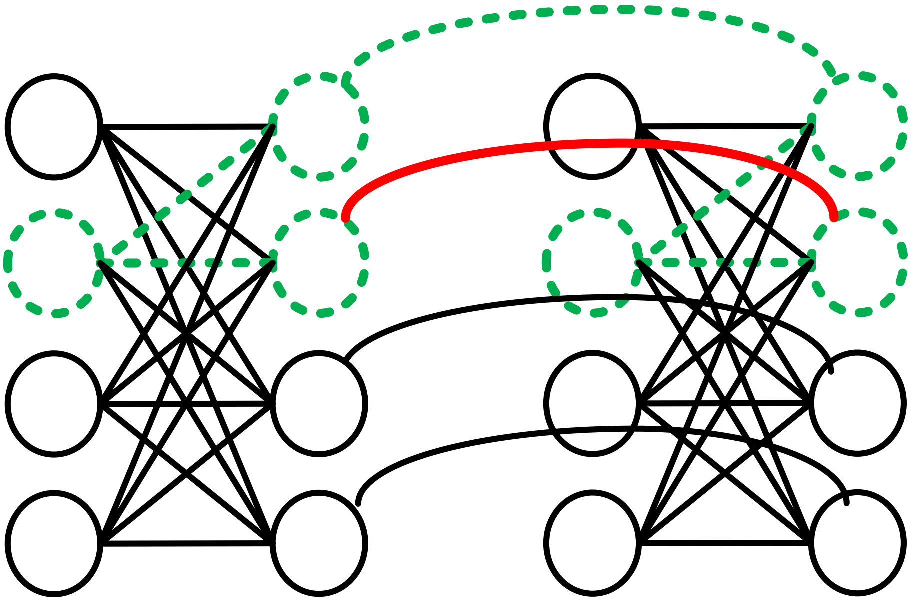


Figure 5.4: Embedding Pattern for Inter-cell Coupler

$f : \{-1, 1\}^n \to \{-1, 1\}$ from 6 to 2. First of all, as S_0,S_1,S_3 are chained with other qubits, we can write f with 3 inputs. Additionally, S_0,S_1,S_3 have logic relationship that S_0 and S_3 determine the value of S_1, hence we can further write f with only 2 qubit inputs S_0 and S_3. We can also eliminate the variables for Ising chains. Furthermore, we can reduce the number of variable by knowing the fact that the S_2 and S_6 can will take the same value. Similarly, for a inter-cell chain, we can also eliminate the number of variables by knowing the fact that chained qubits will take the same value, as shown in Figure 5.5

5.2.5 Annealing between Gate and Interconnect

In Eq. (5.8), the system Hamiltonian is divided into logic gate Hamiltonian and chain Hamiltonian, we can manipulate the relative strength between them.

$$H_{sys} = \alpha \sum H_{O_i} + \beta \sum H_{C_j} \tag{5.8}$$

When $\alpha \gg \beta$, the system will behave just like a group of disconnected Ising gate model,

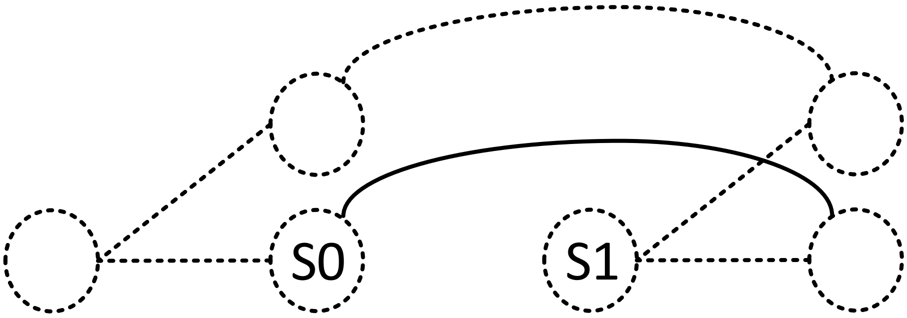


Figure 5.5: Variable Elimination for Inter-Cell Chain

where as when $\beta \gg \alpha$, the system will be reduced to disconnected chains. During experiments, we notice that chain is more easily to break, leaving qubits on the chain take different value. Many researches have also reported similar issue and attribute it to ICE. Although the accurate physical model under the presence of ICE is not yet clear, we employ Eq. (2.2) and the proposed method to further explore the potential to boost the ground state probability.

5.3 Tool Flow and Testing Harness

To investigate and validate the effectiveness of our proposed energy gap optimization strategy, we developed a software tool that automatically enlarges the energy gap for existing embedding solution for satisfiability problem.

Our tool flow is depicted in Figure 5.6. Initially, DIMACS CNF format[cit93] is converted to BLIF[19996]. As CNF is the conjunction of clauses and each clause is the disjunction of a number of variables, an OR gate can be used to represent each clause, and all output of the OR gates will be connected to a fat AND gate. In the regime of quantum annealing, the fat AND gate is removed by forcing the output of each OR gate to a logic one resulting smaller utilization of qubits and couplers. Then, the converted BLIF file will be sent to ABC synthesis tool [BM10]. The purpose of ABC synthesis tool is to generate a logic equivalent network with exclusive 2-input gate. Then, we embed the Boolean network Ising model onto the physical architecture. Lastly, the proposed energy gap optimization step kicks in and generates the configuration with qubit weight and coupler strength.

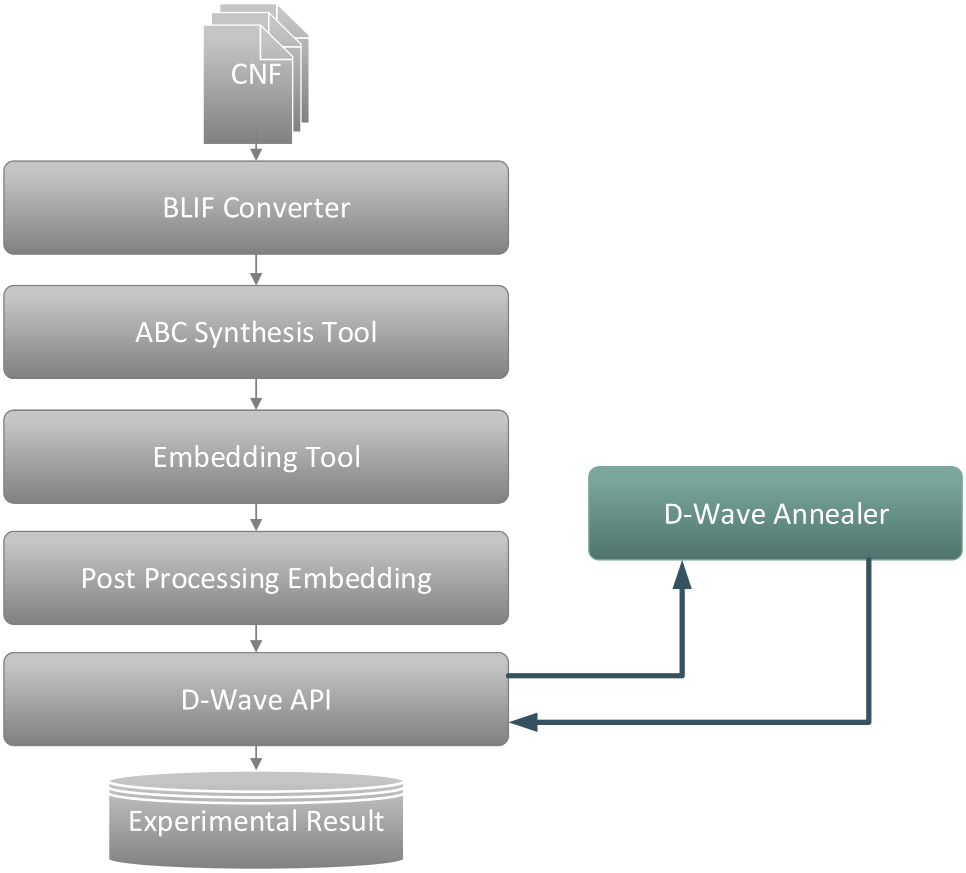


Figure 5.6: Tool Flow

Our energy gap optimization step iterates all the Ising gates in the embedding solution, applying the technique described in this chapter. For chains, as the qubits and couplers are limited, the energy gaps for long chains are optimized followed by short chains.

We also developed testing harness, which remotely connects to the D-Wave quantum annealer through D-Wave APIs[D W17]. The testing harness is able to download the generated configuration to the real device and collect result returned by real machine.

5.4 Experiments

This section demonstrates the experimental setup and results. Our experiments is performed on a D-Wave 2000Q quantum annealer. We compare the experiment result with original embedding and optimized embedding with enlarged energy gap. The purpose of the experiments is to demonstrate the effectiveness of the proposed technique in improving the ground state probability.

5.4.1 Test Case

We use SAT problem generator described in [Spe10] to generate CNF test cases. Since the input is a logic gate network required by the embedding algorithm, logic circuit design can also be used as test case, as long as all the gates in the circuit are 2-input ones. For CNF test cases, we only select the satisfiable cases as their ground state probability can be conveniently determined. As for circuit test cases, the ground state energy can also be calculated, since the ground energy is achieved by the states that is one of the possible states in the original Boolean network.

In our experiment, we selected 11 test cases with different utilization rate. The detail information of each test case is listed in Table 5.4.

5.4.2 Ground State Probability

The ground state probability is highly dependent on the annealing time. Therefore, instead of using the default $20\mu s$, we perform ground state probability experiments by sweeping the annealing time from $1\mu s$ to $500\mu s$. In addition, to smooth out spurious experimental data, we run 10 gauges for each test case. In each gauge, we repeat quantum annealing 1000 times to get the averaged ground state probability. Figure 5.7 and Figure 5.8, show the experimental result between the original embedding and the embedding with the proposed technique applied. The error bar indicates the standard deviation σ, which is calculated by Eq. (5.9). Obviously, the ground state probability increases with annealing time.

Table 5.4: Test Case Resource Utilization

Name	Gate#	Chain#	Qubit#	Coupler#
test0	39	50	409	568
test1	9	13	87	123
test2	32	41	292	405
test3	47	67	558	770
test4	41	50	445	618
test5	22	32	247	342
test6	8	11	63	90
test7	24	22	248	334
test8	69	90	769	1087
test9	34	37	387	536
test10	46	61	545	757

$$\sigma = \sqrt{\frac{\sum (x_i - x_m)^2}{S-1}} \tag{5.9}$$

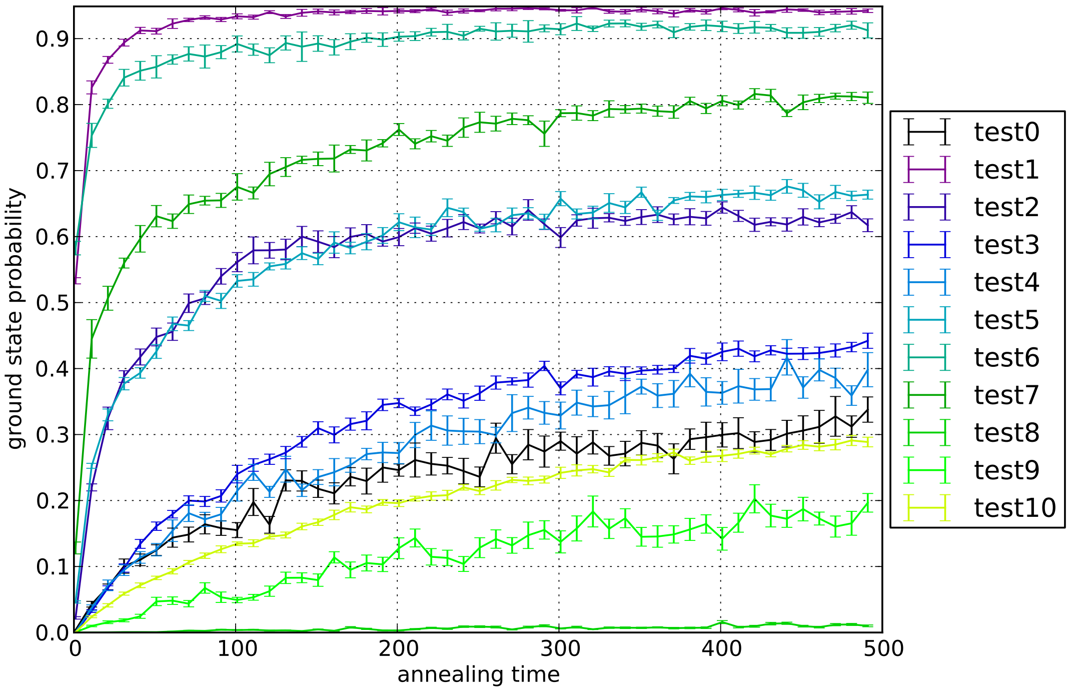


Figure 5.7: Ground State Probability w/o Post-processing

Moreover, the ground state probability is significantly improved with proposed technique, validating the effectiveness. Figure 5.9 clearly shows the improvement by taking the differences between the original embedding and the post-processed embedding result. Despite a few test cases without improvements, most of the cases have shown substantial improvement of ground state probability, ranging from 10% to 20%.

To further investigate the performance of D-Wave quantum annealer, we analyzed the success rate for both Ising gates and Ising chains. The success rate of Ising gate is defined as the probability of the final result being correctly annealed on all Ising gates, that is, the final state satisfying all corresponding logic relations. Similarly, the success rate of chains means the probability of every chain being in the correct state, that is, on each chain all the qubits taking the same value. Figure 5.10-5.13 show the success rate of Ising gates and Ising chains with and without applied the proposed technique. Based on the success rate result,

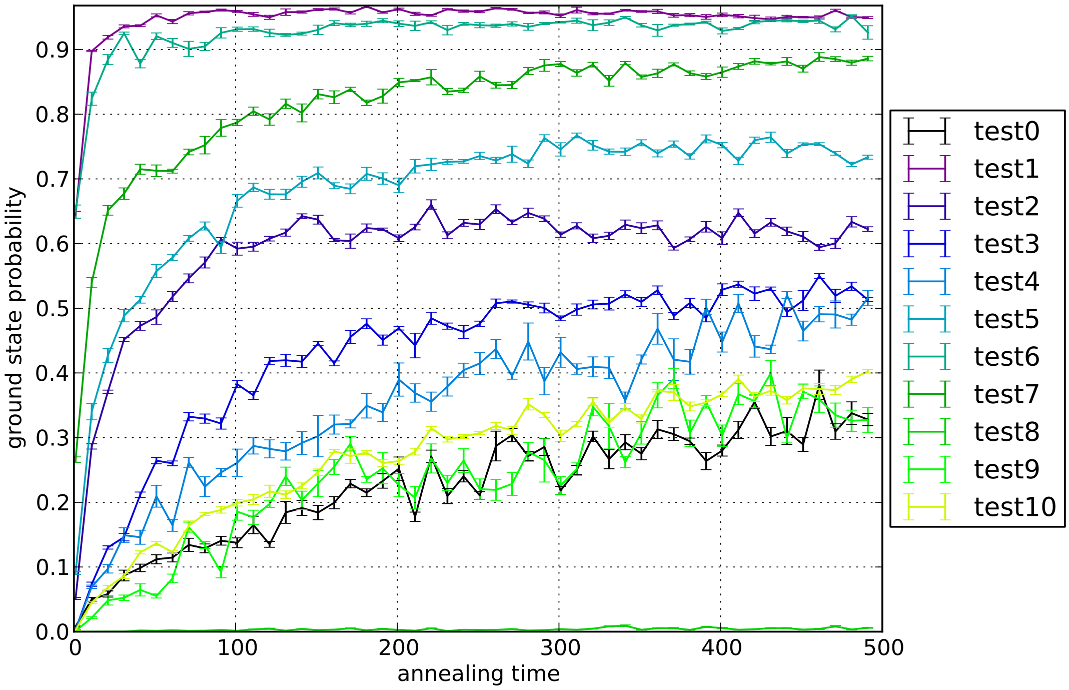


Figure 5.8: Ground State Probability Post w/ Post-processing

Ising chains are have much lower success rate than Ising gates, correlating other research observation[VMK15].

5.4.3 Discussion

In our experimental result, Ising chain shows less improvement in success rate compared with Ising gate. This is because in our post-processing embedding technique, Ising gates have more available auxiliary qubits compared to Ising chains, as imposed by topology limitation. Therefore, Ising gates have higher chance to improve their energy gap. In addition, as discussed in previous sections, long chains are hard to improve yet they are dominating in deciding the ground state probability. Depending on original embedding result, the proposed method may not always increase the energy gap for the entire chain, but rather breaks the long chains into small pieces with enlarged energy gap. Thus, we argue that shortening the chain length becomes the main reason of ground state probability improvement.

The experimental result suggests that the proposed technique improves the ground state

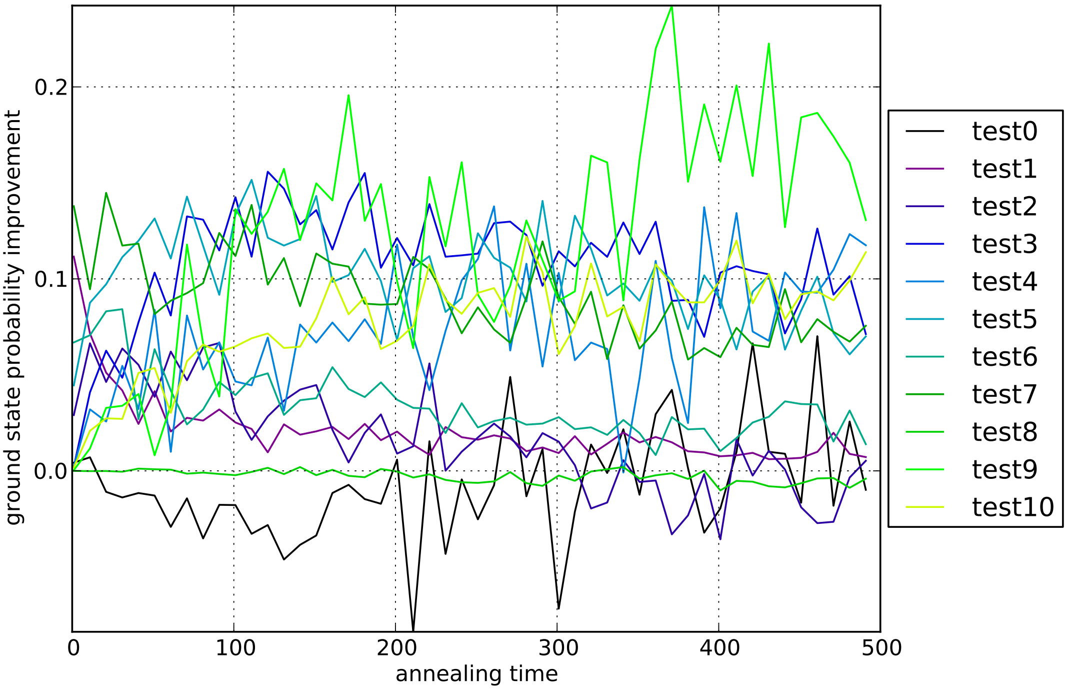


Figure 5.9: Ground State Probability Improvement

probability, showing a compelling method of using quantum annealing device to solve practical problems. Moreover, our technique can be applied to real problems that can be solved by D-Wave quantum annealer, indicating an empirical evidence that quantum annealer does perform expected computation at some degree of confidence.

5.5 Conclusions

Designing and implementing algorithms for practical quantum annealer is never an easy task. In addition to embedding problem, quantum annealers are susceptible to errors, leading to unreliable computation result. In this work we proposed a post-processing embedding technique that improves ground state probability for SAT problems. We have demonstrated that the proposed technique is fully compatible with existing embedding technique. We studied and validated the effectiveness of the proposed technique using the D-Wave 2000Q quantum annealer. Experimental results suggest that the proposed post-processing embedding technique substantially improves the ground state probability, ensuring the quantum annealer

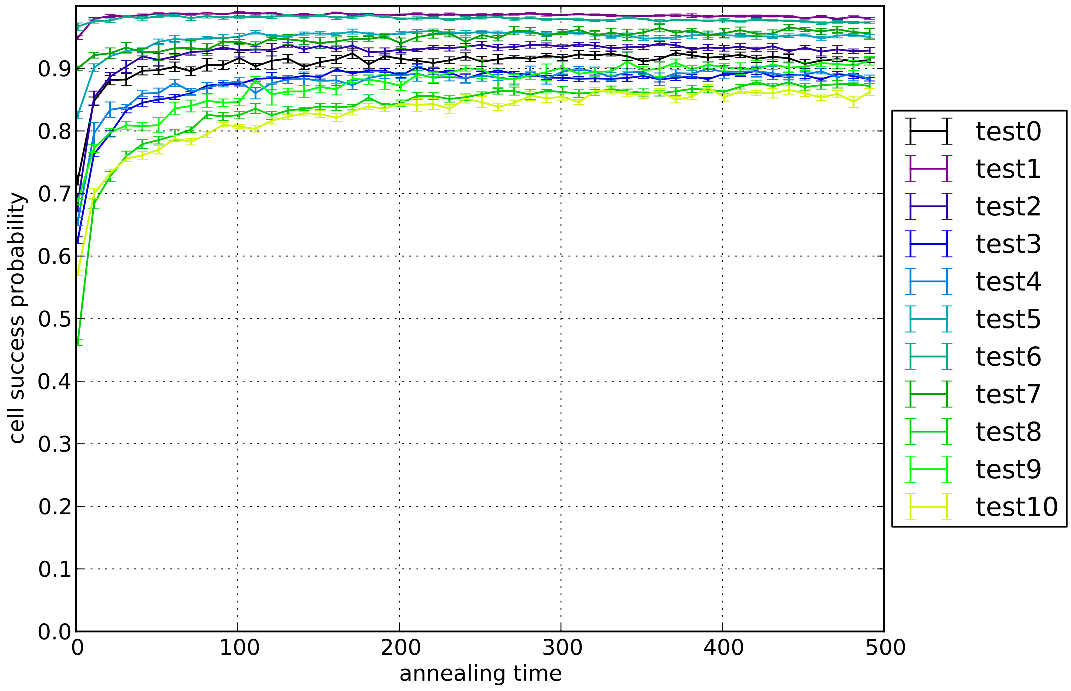


Figure 5.10: Ising Gate Success Rate w/o Post-processing

produce high fidelity results. Moreover, we have also demonstrated the first comprehensive software stack that compiles SAT problems to quantum annealer as well as collects experimental results.

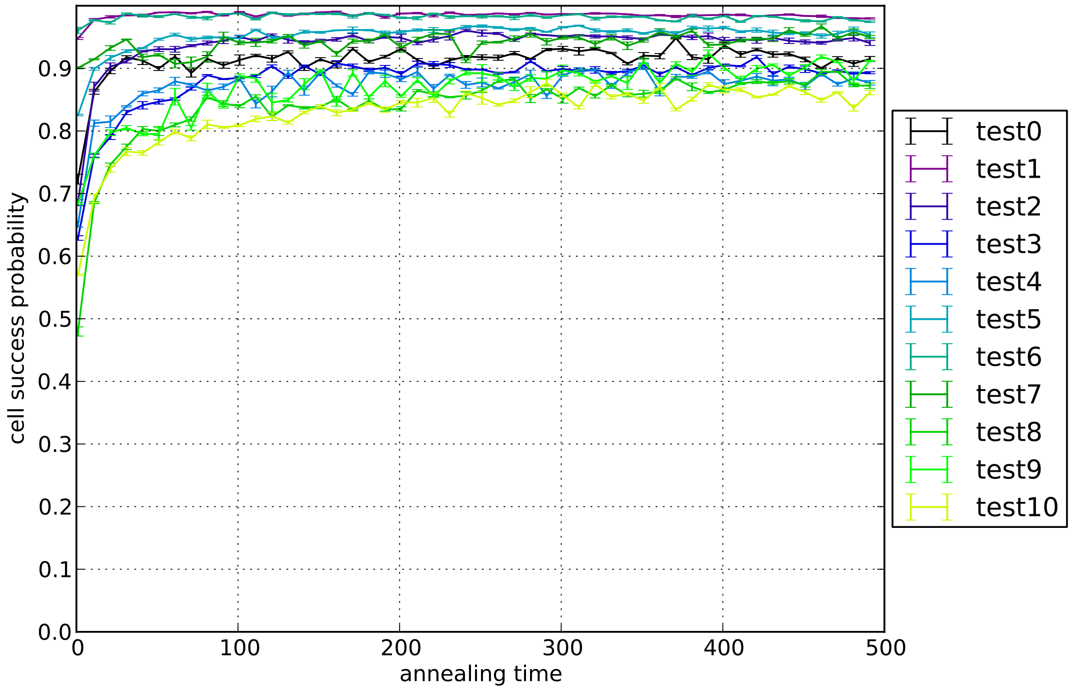


Figure 5.11: Ising Gate Success Rate w/ Post-processing

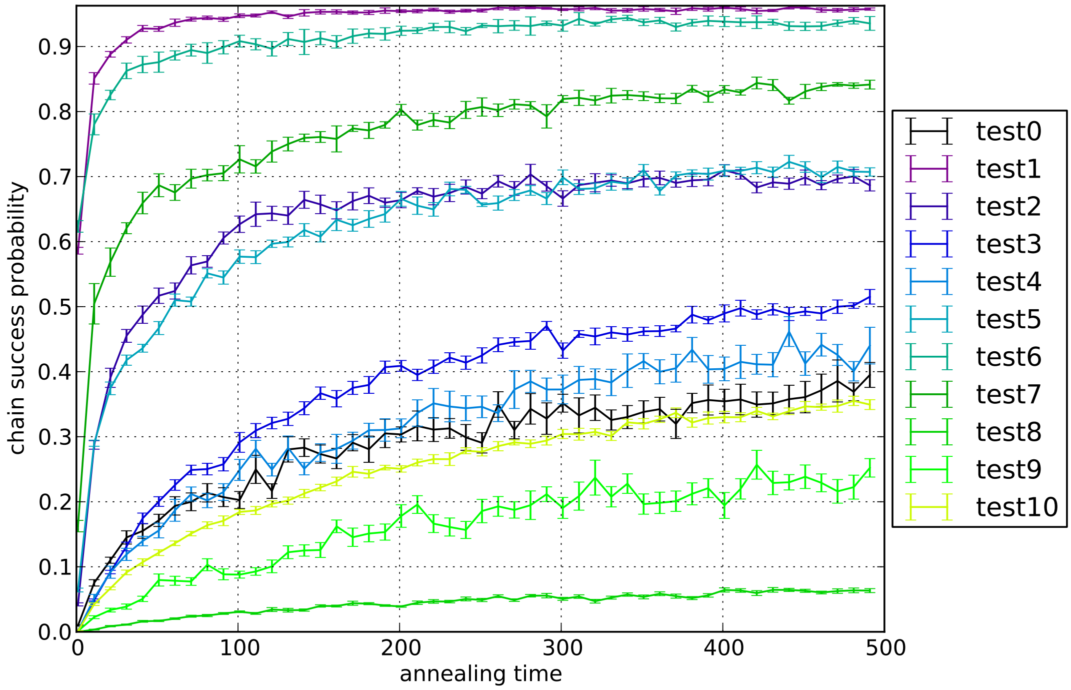


Figure 5.12: Chain Success Rate w/o Post-processing

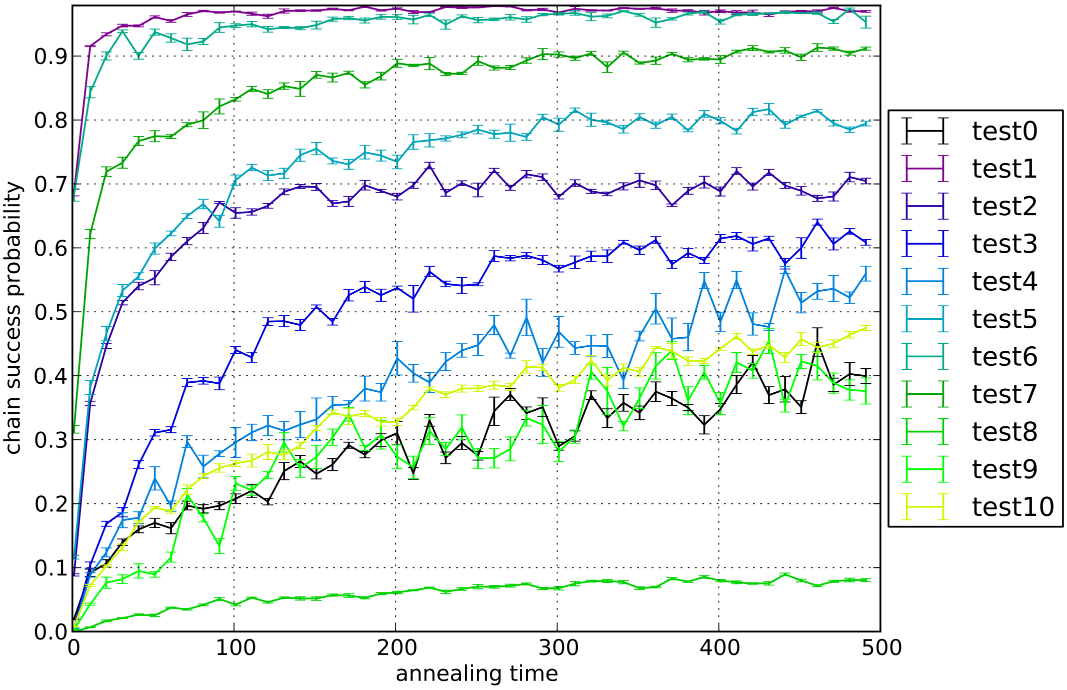


Figure 5.13: Chain Success Rate w/ Post-processing

CHAPTER 6

Summary

With the emergence of practical quantum computer, designing algorithms to harness its enormous computational power becomes an urgent problem. We proposed a detailed flow that maps SAT problem into Ising model energy minimization problem. We proposed that the embedding problem to D-Wave architecture can be solved using place-and-route technique. We proposed an approach to optimize embedding chain length at reduced run-time. We proposed an energy gap optimization method to improve the reliability of quantum annealing.

Throughout the work, we have demonstrated the significance of our method which bridges the gap between the end users and the actual quantum annealing device, awaiting for the realization of the next generation of quantum annealer. As compared to semiconductor industry where chip designers rely on computer-aided design tools, there is a huge demand for software tool that eases the programming task for quantum annealers, paving the road to achieve quantum supremacy.

REFERENCES

[19996] "Berkeley Logic Interchange Format (blif)." 1996.

[ADF11] Isolde Adler, Frederic Dorn, Fedor V. Fomin, Ignasi Sau, and Dimitrios M. Thilikos. "Faster parameterized algorithms for minor containment." *Theoretical Computer Science*, **412**(50):7018–7028, November 2011.

[AGP06] Panos Aliferis, Daniel Gottesman, and John Preskill. "Quantum Accuracy Threshold for Concatenated Distance-3 Codes." *Quantum Info. Comput.*, **6**(2):97–165, March 2006.

[AH15] Steven H. Adachi and Maxwell P. Henderson. "Application of Quantum Annealing to Training of Deep Neural Networks." *arXiv:1510.06356 [quant-ph, stat]*, October 2015. arXiv: 1510.06356.

[Comment: 18 pages.]

[BBH09] A. Biere, A. Biere, M. Heule, H. van Maaren, and T. Walsh. *Handbook of Satisfiability: Volume 185 Frontiers in Artificial Intelligence and Applications.* IOS Press, Amsterdam, The Netherlands, The Netherlands, 2009.

[BCI14] Zhengbing Bian, Fabian Chudak, Robert Israel, Brad Lackey, William G. Macready, and Aidan Roy. "Discrete optimization using quantum annealing on sparse Ising models." *Interdisciplinary Physics*, **2**:56, 2014.

[BCI16] Zhengbing Bian, Fabian Chudak, Robert Israel, Brad Lackey, William G. Macready, and Aidan Roy. "Mapping constrained optimization problems to quantum annealing with application to fault diagnosis." *arXiv:1603.03111 [quant-ph]*, March 2016. arXiv: 1603.03111.

[Comment: 22 pages, 4 figures.]

[BCM10] Zhengbing Bian, Fabián A. Chudak, William G. Macready, and Geordie Rose. "The Ising model: teaching an old problem new tricks." 2010.

[BDH14] Anton Belov, Daniel Diepold, Marijn J. H. Heule, and Matti Jrvisalo. "Proceedings of SAT Competition 2014." 2014.

[BM10] Robert Brayton and Alan Mishchenko. "ABC: An Academic Industrial-strength Verification Tool." In *Proceedings of the 22Nd International Conference on Computer Aided Verification*, CAV'10, pp. 24–40. Springer-Verlag, 2010.

[BRI14] Sergio Boixo, Troels F. Rnnow, Sergei V. Isakov, Zhihui Wang, David Wecker, Daniel A. Lidar, John M. Martinis, and Matthias Troyer. "Quantum annealing with more than one hundred qubits." *Nature Physics*, **10**(3):218–224, February 2014. arXiv: 1304.4595.

[Comment: 23 pages, 38 figures. Revised version. Text rewritten for clarity, added comparison with spin dynamics model.]

[CD10] Andrew M. Childs and Wim van Dam. "Quantum algorithms for algebraic problems." *Reviews of Modern Physics*, **82**(1):1–52, January 2010.

[CFP01] Andrew M. Childs, Edward Farhi, and John Preskill. "Robustness of adiabatic quantum computation." *Physical Review A*, **65**(1), December 2001. arXiv: quant-ph/0108048.

[Comment: 11 pages, 5 figures, REVTeX.]

[Che94] Chih-liang Eric Cheng. "Risa: Accurate And Efficient Placement Routability Modeling." In *, IEEE/ACM International Conference on Computer-Aided Design, 1994*, pp. 690–695, November 1994.

[Cho08] Vicky Choi. "Minor-embedding in adiabatic quantum computation: I. The parameter setting problem." *Quantum Information Processing*, **7**(5):193–209, October 2008.

[cit93] "Satisfiability Suggested Format." Technical report, 1993.

[CMR14] Jun Cai, William G. Macready, and Aidan Roy. "A practical heuristic for finding graph minors." *arXiv:1406.2741 [quant-ph]*, June 2014. arXiv: 1406.2741.

[Comment: 16 pages, 7 figures.]

[D W17] Inc. D-Wave System. "The D-Wave 2000Q Quantum Computer Technology Overview.", 2017.

[DJA13] N G Dickson, M W Johnson, M H Amin, R Harris, F Altomare, A J Berkley, P Bunyk, J Cai, E M Chapple, P Chavez, F Cioata, T Cirip, P deBuen, M Drew-Brook, C Enderud, S Gildert, F Hamze, J P Hilton, E Hoskinson, K Karimi, E Ladizinsky, N Ladizinsky, T Lanting, T Mahon, R Neufeld, T Oh, I Perminov, C Petroff, A Przybysz, C Rich, P Spear, A Tcaciuc, M C Thom, E Tolkacheva, S Uchaikin, J Wang, A B Wilson, Z Merali, and G Rose. "Thermally assisted quantum annealing of a 16-qubit problem." *Nature Communications*, **4**:1903, May 2013.

[DS96] David P. DiVincenzo and Peter W. Shor. "Fault-Tolerant Error Correction with Efficient Quantum Codes." 1996.

[FGG00] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. "Quantum Computation by Adiabatic Evolution." *arXiv:quant-ph/0001106*, January 2000. arXiv: quant-ph/0001106.

[Comment: 24 pages, 12 figures, LaTeX, amssymb,amsmath, BoxedEPS packages; email to farhi@mit.edu.]

[FGG01] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. "A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem." *Science*, **292**(5516):472–475, April 2001.

[FGS94] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll. "Quantum Annealing: A New Method for Minimizing Multidimensional Functions." *Chemical Physics Letters*, **219**(5-6):343–348, March 1994. arXiv: chem-ph/9404003.

[Gur16] Inc. Gurobi Optimization. "Gurobi Optimizer Reference Manual.", 2016.

[Hea13] Corporate Headquarters. "Programming with D-Wave: Map Coloring Problem." 2013.

[HJL10] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose. "Experimental Investigation of an Eight Qubit Unit Cell in a Superconducting Optimization Processor." *Physical Review B*, **82**(2), July 2010. arXiv: 1004.1628.

[Comment: 16 pages, 12 figures. Expanded data analysis as compared to version 1.]

[Ibm11] Ibm. *IBM ILOG CPLEX Optimization Studio CPLEX User's Manual*, 2011.

[JAG11] M. W. Johnson, M. H. S. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, P. Bunyk, E. M. Chapple, C. Enderud, J. P. Hilton, K. Karimi, E. Ladizinsky, N. Ladizinsky, T. Oh, I. Perminov, C. Rich, M. C. Thom, E. Tolkacheva, C. J. S. Truncik, S. Uchaikin, J. Wang, B. Wilson, and G. Rose. "Quantum annealing with manufactured spins." *Nature*, **473**(7346):194–198, May 2011.

[KM14] Andrew D. King and Catherine C. McGeoch. "Algorithm engineering for a quantum annealing platform." *arXiv:1410.2628 [quant-ph]*, October 2014. arXiv: 1410.2628.

[Comment: 16 pages. V2: minor edits.]

[LAB14] T. Lanting, M. H. Amin, A. J. Berkley, C. Rich, S.-F. Chen, S. LaForest, and Rogerio de Sousa. "Evidence for temperature dependent spin-diffusion as a mechanism of intrinsic flux noise in SQUIDs." *Physical Review B*, **89**(1), January 2014. arXiv: 1306.1512.

[Comment: Expanded version to appear in Phys. Rev. B.]

[LD88] J. Lam and J.-M. Delosme. "Performance of a new annealing schedule." In *, 25th ACM/IEEE Design Automation Conference, 1988. Proceedings*, pp. 306–311, June 1988.

[LKJ11] Jason Luu, Ian Kuon, Peter Jamieson, Ted Campbell, Andy Ye, Wei Mark Fang, Kenneth Kent, and Jonathan Rose. "VPR 5.0: FPGA CAD and Architecture Exploration Tools with Single-driver Routing, Heterogeneity and Process Scaling." *ACM Trans. Reconfigurable Technol. Syst.*, **4**(4):32:1–32:23, December 2011.

[MBR00] Alexander Marquardt, Vaughn Betz, and Jonathan Rose. "Timing-driven Placement for FPGAs." In *Proceedings of the 2000 ACM/SIGDA Eighth International Symposium on Field Programmable Gate Arrays*, FPGA '00, pp. 203–213, New York, NY, USA, 2000. ACM.

[ME95] L. McMurchie and C. Ebeling. "PathFinder: A Negotiation-Based Performance-Driven Router for FPGAs." In *Proceedings of the Third International ACM Symposium on Field-Programmable Gate Arrays, 1995. FPGA '95*, pp. 111–117, 1995.

[NDR12] H. Neven, V.S. Denchev, G. Rose, and W.G. Macready. "QBoost: Large Scale Classifier Training withAdiabatic Quantum Optimization." In Steven C. H. Hoi and Wray Buntine, editors, *Proceedings of the Asian Conference on Machine Learning*, volume 25 of *Proceedings of Machine Learning Research*, pp. 333–348, Singapore Management University, Singapore, 04–06 Nov 2012. PMLR.

[NTK15] Hidetoshi Nishimori, Junichi Tsuda, and Sergey Knysh. "Comparative Study of the Performance of Quantum Annealing and Simulated Annealing." *Physical Review E*, **91**(1), January 2015. arXiv: 1409.6386.

[Comment: 19 pages.]

[OD14] Ryan O'Donnell. *Analysis of Boolean Functions.* Cambridge University Press, 2014.

[PAL14] Kristen L. Pudenz, Tameem Albash, and Daniel A. Lidar. "Error-corrected quantum annealing with hundreds of qubits." *Nature Communications*, **5**:3243, February 2014.

[PDD12] Alejandro Perdomo-Ortiz, Neil Dickson, Marshall Drew-Brook, Geordie Rose, and Aln Aspuru-Guzik. "Finding low-energy conformations of lattice protein models by quantum annealing." *arXiv:1204.5485 [quant-ph]*, April 2012. arXiv: 1204.5485.

[PFN15] Alejandro Perdomo-Ortiz, Joseph Fluegemann, Sriram Narasimhan, Rupak Biswas, and Vadim N. Smelyanskiy. "A Quantum Annealing Approach for Fault Detection and Diagnosis of Graph-Based Systems." *The European Physical Journal Special Topics*, **224**(1):131–148, February 2015. arXiv: 1406.7601.

[QL12] Gregory Quiroz and Daniel A. Lidar. "High Fidelity Adiabatic Quantum Computation via Dynamical Decoupling." 2012.

[RHG16] Gili Rosenberg, Poya Haghnegahdar, Phil Goddard, Peter Carr, Kesheng Wu, and Marcos Lpez de Prado. "Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer." *IEEE Journal of Selected Topics in Signal Processing*, **10**(6):1053–1060, September 2016. arXiv: 1508.06182.

[Comment: 7 pages; expanded and updated.]

[RVO15] Eleanor G. Rieffel, Davide Venturelli, Bryan O'Gorman, Minh B. Do, Elicia M. Prystay, and Vadim N. Smelyanskiy. "A case study in programming a quantum annealer for hard operational planning problems." *Quantum Information Processing*, **14**(1):1–36, January 2015.

[SL05] M. S. Sarandy and D. A. Lidar. "Adiabatic Quantum Computation in Open Systems." *Phys. Rev. Lett.*, **95**:250503, Dec 2005.

[Spe10] Ivor Spence. "Sgen1: A Generator of Small but Difficult Satisfiability Benchmarks." *J. Exp. Algorithmics*, **15**:1.2:1.1–1.2:1.15, March 2010.

[TCT17] Shu Tanaka, B. K. Chakrabarti, and Ryo Tamura. *Quantum spin glasses, annealing and computation.* Cambridge University Press, Cambridge, 2017. OCLC: ocn973907527.

[VMK15] Davide Venturelli, Salvatore Mandr, Sergey Knysh, Bryan O'Gorman, Rupak Biswas, and Vadim Smelyanskiy. "Quantum Optimization of Fully-Connected Spin Glasses." *Physical Review X*, **5**(3), September 2015. arXiv: 1406.7553.

[Comment: includes supplemental material.]

[YSB13] Kevin C. Young, Mohan Sarovar, and Robin Blume-Kohout. "Error Suppression and Error Correction in Adiabatic Quantum Computation: Techniques and Challenges." *Physical Review X*, **3**(4):041013, November 2013.