Haplotype Inference with Pure Parsimony: A Quantum Computing Approach

Nguyen-Viet-Dung Nghiem
Sy-Vinh Le
University of Engineering and Technology
Vietnam National University, Hanoi, Vietnam
{dung.nghiem, vinhls}@vnu.edu.vn

Tu N. Nguyen
Kennesaw State Univ.
Marietta, GA, USA
tu.nguyen@kennesaw.edu

Thang N. Dinh
Virginia Commonwealth Univ.
Richmond, VA, USA
tndinh@vcu.edu

***Abstract*—Haplotype inference with pure parsimony (HIPP) problem seeks to reconstruct a minimum set of haplotypes that explain a given set of genotypes observed from a population. This important problem is known to be NP-hard. In this paper, we explore the potential of *quantum computing* in retrieving optimal solutions for HIPP. We investigate several approaches to encode HIPP into quadratic unconstrained binary optimization (QUBO), which can be solved on quantum annealers. Further, we propose a new QUBO for HIPP, termed QHI, exploiting the structure of HIPP to reduce the QUBO size. Our comprehensive experiments on the state-of-the-art D-Wave annealer indicate comparable solution quality for quantum annealing approaches compared to classical simulated annealing. They also validate the effectiveness of our proposed QHI formulation in both solution quality and size.**

***Index Terms*—Haplotype inference, Quantum annealing**

I. Introduction

A haplotype (haploid genotype) is a group of alleles in an organism that can be used to infer various important information including ancestry or demographic history. However, it is difficult to obtain direct information on the haplotypes with current high-throughput sequencing methods. Haplotype inference methods aim to infer haplotypes from the genotype data, i.e., each genotype is detached into two haplotypes. Given a set of genotypes, the haplotype inference with pure parsimony (HIPP) uses the parsimony criterion to identify a set of the fewest possible haplotypes such that each genotype in the set can be explained by one pair of haplotypes.

Many methods have been proposed to solve this important problem. Gusfield [1] proposes an Integer Programming (IP) formulation for the HIPP that considers all possible pairs of haplotypes, leading to the worst-case of an exponential large formulation. Later, the polynomial-size models were proposed by [2]–[4]. However, there is no clear efficient approach to find exact solutions for the problem as the problem is shown to be NP-hard [1]. Other methods proposed for the problem include boolean satisfiability (SAT) and [5], pseudo-boolean optimization [6], local search [7], and swarm optimization [8]. These algorithms are, however, not guaranteed to produce an optimal solution. The recent exponential growth in *quantum computing* with a record number of breakthroughs [9]–[13] has opened new venues for solving NP-hard optimization problems. By encoding information using quantum bits (qubits), quantum computing can leverage the superposition of states [11] as well as quantum mechanics phenomena such as entanglement and quantum tunneling to explore exponential combinations of states at once. QC has paved the way for *faster, more efficient solutions to large-scale, real-world optimization problems* that are challenging for classical computers [9], [11].

In this paper, we explore approaches to solve HIPP using *quantum annealing* (QA), a method to search for global minimum using quantum fluctuation [14]. QA is currently the only quantum computing approach that provides a large enough number of qubits for real-world problems from life science [15], scheduling for car manufacturing [16] and many others [17], [18]. Will quantum annealing be also effective for haplotype inference?

We begin with the exploration of existing QUBO formulations for HIPP including the standard approach to convert the integer linear programming (ILP) [1] into QUBO [19] using penalties. We also adopt a QUBO in Cao et al. [20] for the set cover with pairs (SCP) problem that admits HIPP as a special case. Finally, we propose an efficient method, called Quadratic for Haploptype Inference or QHI. Our method transforms the ILP in [1] by iterative aggregating size-k groups of variables in each constraint to save on the size of the resulting QUBO, for some integer $k \geq 2$. Our formulation also leverages the preprocessing techniques in [1] to reduce the size by eliminating trivial constraints and haplotype candidates.

We provide a comprehensive evaluation of all the QUBO formulations on the latest D-Wave's quantum annealer advantage [21]. In addition, we also benchmark the QUBO formulations using the classical simulated annealing (SA) to seek evidence of quantum advantage. Finally, we analyze the efficiency of the proposed formulations in terms of the number of variables and physical qubits.

Organization. The rest of the paper is organized as follows. We introduce the HIPP and QA in section II. Section III presents QUBO formulations including our QHI formulation. The experiment results are shown in section IV. Finally, section V concludes with our discussion and future directions.

II. Preliminaries

We present the preliminaries on the Haplotype Inference (HIPP) problem and background on solving optimization prob-

Corresponding author tndinh@vcu.edu

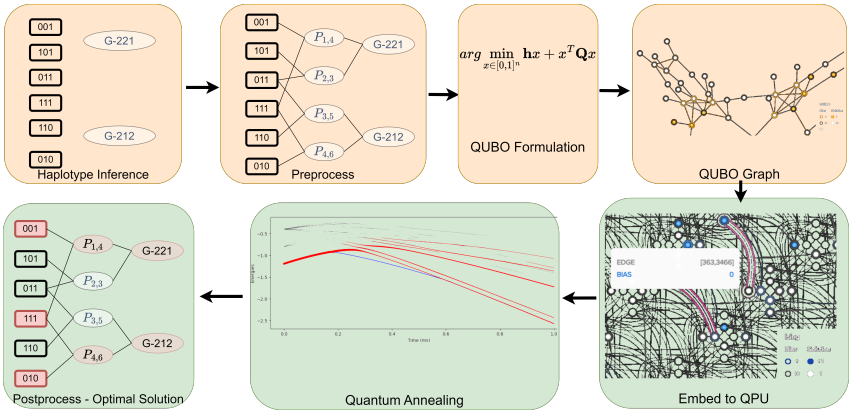

Fig. 1: Steps to solve Haplotype Inference using quantum annealing. In phase 1 (orange blocks), we find the set of pairs for each genotype and create QUBO graph for each problem instance. In phase 2 (green blocks), we embed the QUBO graph into physical QPU and find the optimal solutions through quantum annealing.

lems with quantum annealing (QA).

A. Haplotype Inference with Pure Parsimony Problem

As humans inherit a set of chromosomes from each parent, a genotype (a set of observed genetic variations) at a particular site may contain information from both parental chromosomes. However, directly observing the separate contributions of each parent (haplotypes) remains challenging. HIPP seeks to determine the minimal number of haplotypes that can explain a given set of genotypes, using the principle of parsimony—that is, explaining the data with the least number of unique haplotypes. Mathematically, HIPP can be visualized as a combinatorial optimization problem, where the goal is to minimize the number of haplotypes while satisfying the constraints imposed by the observed genotypes.

Given l sites of interest on the chromosome. Each site is either homozygous (if two haplotypes of a pair share the same value) or heterozygous (they are different). For simplicity, we assume that genotypes are depicted as sequences containing elements with values 0, 1, or 2. Here, values 0 and 1 signify homozygous sites, with 0 for the wild-type allele and 1 for the mutant. The value 2 indicates heterozygous sites. Consequently, haplotypes are represented by sequences consisting of values 0 or 1.

Definition 1 (Haplotype Inference [22])**.** *Given n genotypes $G = \{g_1, g_2, \ldots, g_n\}$, each represented as a length l string $g_i \in \{0,1,2\}^l$ and m candidate haplotypes $H = \{h_1, h_2, \ldots, h_m\}$, each represented as a string $h_j \in \{0,1\}^l$, such that each genotype can be explained by a pair of haplotypes. A pair of haplotypes $h_i, h_j \in H$ can explain a genotype g_t if and only if each position $p \in [1..l]$ satisfies:*

$$g_{t,p} = \begin{cases} 0 & \text{when } h_{i,p} = h_{j,p} = 0, \\ 1 & \text{when } h_{i,p} = h_{j,p} = 1, \\ 2 & \text{when } h_{i,p} + h_{j,p} = 1. \end{cases}$$

Definition 2 (Haplotype Inference by Pure Parsimony (HIPP))**.** *Given a collection of genotypes, a solution to the HIPP problem seeks a set of haplotypes that explains the genotypes using the fewest possible distinct haplotypes.*

An example of the problem is shown in Fig. 1. First, we find all pairs that are associated with one genotype in the set. Genotype 221 could be explained by $(001, 111)$ and $(101, 011)$. Whereas genotype 212 could be explained by $(011, 110)$ and $(111, 010)$. The optimal solution is to select $H = \{H_1, H_4, H_6\}$ to cover all genotypes.

HIPP is an NP-hard problem [1]; thus, we often need to rely on exact approaches with exponential running time like branch-and-bound or polynomial-time heuristics that may have exponentially bad performance guarantees [23].

B. Integer Program Formulation by Gusfield [24]

The HIPP problem can be formulated as an integer linear programming (ILP) [24]. Let $x_i \in \{0,1\}$ be binary variables in which $x_i = 1$ iff haplotype H_i is selected for $1 \le i \le m$. The objective is to minimize the number of selected haplotypes, i.e., $\min \sum_{i=1}^{m} x_i$.

For each genotype $g_t, t \in [1, n]$, we define a set $\mathbb{P}_t$ that contains all pairs of haplotypes (H_i, H_j) that can explain genotype g_t, i.e.,

$$\mathbb{P}_t = \{(i,j) | (H_i, H_j) \text{ explains } g_t\}.$$

Further, let $p_{ij}, i, j \in [1..m]$ be binary variables indicating whether or not the pair (H_i, H_j) is selected. The constraints $x_i \ge p_{ij}$ and $x_j \ge p_{ij}$ are to enforce that $p_{ij} = 1$ iff $x_i = x_j = 1$, i.e., both the haplotypes must be selected. For each genotype g_t, we need to ensure one pair of haplotypes that explain g_t is selected, i.e., $\sum_{(i,j)\in\mathbb{P}_t} p_{ij} = 1$. The complete ILP for HIPP by Gusfield [24], referred to as ILP$_G$, is shown below in equations (1)-(3).

$$\min \quad \sum_{j=1}^{m} x_j \tag{1}$$

$$s.t. \quad p_{ij} \le x_i \text{ and } p_{ij} \le x_j \qquad 1 \le i, j \le m \tag{2}$$

$$\sum_{(i,j)\in\mathbb{P}_t} p_{ij} = 1 \qquad 1 \le t \le n \tag{3}$$

C. Quantum Annealing and QUBO

Quantum Annealing (QA) provides an approach for finding near-optimal solutions for NP-hard problems that can be encoded into a quadratic unconstrained binary optimization (QUBO) [19]. A QUBO minimizes a quadratic polynomial over binary variables

$$\mathbf{x}^* = \arg\min_{\mathbf{x}\in\{0,1\}^n} Q(\mathbf{x}) = \sum_{i,j\in[1..n]} q_{ij}x_ix_j,$$

where $\mathbf{x} = (x_1, \cdots, x_n) \in \{0,1\}^n$.

By changing variables $x_i = \frac{s_i+1}{2}$, a QUBO can be easily converted back and forth to an Ising Hamiltonian [25]

$$H(\mathbf{s}) = -\sum_{i=1}^{n} h_i s_i - \sum_{i,j=1}^{n} J_{ij}s_is_j = -\mathbf{h}^T s - \mathbf{s}^T\mathbf{J}\mathbf{s} \quad (4)$$

Here, each discrete variable $s_i \in= \{-1,+1\}$ represents the site's spin. Each assignment of spin value $s \in \{-1,+1\}^n$, called a spin configuration, is associated with an energy of the system; h_i is the external magnetic field at site i and J_{ij} is the coupling strength between sites i and j. Then, minimizing the QUBO is equivalent to finding the lowest energy state, called the ground state, of the Hamiltonian.

In the optimization approach on D-Wave Quantum annealers [21], each variable s_i is assigned a value from the set $\{-1,+1\}$, indicating the site's spin. The energy of a system is associated with a spin configuration, which is described by a spin value s from $\{-1,+1\}^n$. The external magnetic field at a specific site is represented by h_i, while J_{ij} denotes the coupling strength between sites i and j.The optimal solutions of the optimization problem, represented using QUBO, are encoded within the ground states of the Hamiltonian. The ground state of a Hamiltonian is associated with the spin configuration of the lowest energy and can be searched for using the quantum annealing process. The objective is to minimize the Quadratic Unconstrained Binary Optimization (QUBO) to determine the Hamiltonian's ground state.

The Ising Hamiltonian is then mapped to the quantum processing unit (QPU), which is a fixed hardware graph. Due to the limited connectivity in the QPU, each logical spin may be mapped to multiple physical spin qubits, with strong coupling strength among them, through a process called minor-embedding [25]. The success probability in obtaining optimal solutions with QA hinges on formulating effective QUBOs and optimizing QA solver parameters such as chain strength, annealing time, and so on [21].

III. HIPP QUBOs for Quantum Annealing

We investigate three different QUBO formulations for HIPP. The first approach, named DI, is a direct transformation of ILP$_G$ in Eq. 1 into QUBO. The second formulation is a QUBO for the Set Cover with Pairs by Cao et al. [20]. Lastly, we propose a new QUBO, called QHI^k, that repeatedly forms penalties on groups of k variables in the constraints.

A. A Direct Penalty-based Approach (DI)

We follow the standard approach in [26] to transform ILP$_G$ into an equivalent QUBO. Consider a binary integer linear programming (ILP)

$$\begin{aligned} &\text{minimize} && \mathbf{c}^{\mathrm{T}}\mathbf{x} \\ &\text{subject to} && \mathbf{A}\mathbf{x} \le \mathbf{b}, \\ & && \mathbf{x} \in \{0,1\}^n \end{aligned}$$

where $\mathbf{c} \in \mathbb{R}^n, \mathbf{b} \in \mathbb{R}^m$ are vectors and $A \in \mathbb{R}^{m\times n}$ is a matrix. We can transform the above ILP into a QUBO by introducing *binary* slack variables $\mathbf{s}$ to obtain the equivalent equalities

$$\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{s} = \mathbf{b}$$

and convert all the (hard) constraints into (soft) penalties to obtain a QUBO

$$\mathbf{c}^{\mathrm{T}}\mathbf{x} + \lambda\|A\mathbf{x} + \mathbf{B}\mathbf{s} - \mathbf{b}\|_2^2,$$

where $\|.\|_2$ denotes the norm 2 and $\lambda > 0$ is a sufficiently large constant. Thus, the constraints in Eq. 3 on the explanation of genotypes can be converted into a soft penalty

$$\lambda_1 \sum_{t=1}^{n} \Big(\sum_{(i,j)\in\mathbb{P}_t} p_{ij} - 1 \Big)^2,$$

where $\lambda_1 > 0$ is a sufficiently large constant.

For the constraints in (2), we adopt a more succinct encoding into penalties of the inequality $x \le y$ into a penalty $x - xy$ in [19] to obtain the second penalty term

$$\lambda_2 \sum_{1\le i<j\le m} \Big(p_{ij} - p_{ij}x_i + p_{ij} - p_{ij}x_j \Big) \quad (5)$$

Thus, we obtain a direct conversion of the ILP$_G$ into QUBO

$$\begin{aligned} Q_{DI} = \sum_{j=1}^{m} x_j + \lambda_1 \sum_{t=1}^{n} \Big(\sum_{(i,j)\in\mathbb{P}_t} p_{ij} - 1 \Big)^2 \\ \lambda_2 \sum_{i=1}^{m} \sum_{j=1+1}^{m} \big(2p_{ij} - p_{ij}x_i - p_{ij}x_j \big) \end{aligned} \quad (6)$$

While the transformation of Q_{DI} is simple, its major disadvantage is the large number of variables, up to $O(n^4)$ in the worst case. To cope with the limited number of qubits on existing quantum devices, we will explore two more efficient QUBOs with significantly smaller sizes of $O(n^2)$ variables.

B. Set Cover with Pairs approach [20] - SCP

Since HIPP can be seen as a special case of the Set Cover with Pairs (SCP) problem [27], QUBO formulations for SCP can also be adapted to solve the HIPP. We continue by presenting the definition of SCP, a generalization of the NP-hard *Set Cover* problem [28], how HIPP can be mapped to SCP, and the QUBO formulation for SCP.

Definition 3 (Set Cover with Pairs (SCP) [28])**.** *Let U be a ground set of elements and let S be a set of objects, where each object i has a non-negative cost w_i. For every pair (i,j), let $Q(i,j)$ be the collection of elements in U covered by the*

pair (i,j). The set cover with pairs (SCP) problem asks to find a subset $A \subseteq S$ of minimum cost $\sum_{i\in A} w_i$ such that $\bigcup_{(i,j)\in A} Q(i,j) = U$.

The HIPP is a special case of SCP with the set of objects $S = H = \{h_1, \ldots, h_m\}$, uniform unit cost $w_i = 1, i = 1..m$, and coverage function $Q(i,j) = \{g_t\}$ for haplotype pair $(i,j) \in \mathbb{P}_t$. In fact, the ILP for SCP in Cao et al. [20] is similar to the below ILP for HIPP in Lancia et al. [23].

$$\begin{aligned} \min \quad & \sum_{j=1}^{m} x_j \\ s.t. \quad & p_{ij} \le x_i \text{ and } p_{ij} \le x_j \qquad 1 \le i,j \le m \\ & \sum_{(i,j)\in\mathbb{P}_t} p_{ij} \ge 1 \qquad 1 \le t \le n \end{aligned} \tag{7}$$

Note that the above ILP for SCP/HIPP shares the same objective and the constraints $p_{ij} \le x_i, x_j$ with ILP$_G$ in Eqs.(2)-(3). The only difference is that the equality in Eq. (3) is changed into an inequality in Eq. (7). Thus, we allow multiple pairs of haplotypes that explain the same genotype.

Cao et al. [20] proposed an Ising Hamiltonian formulation for the set cover with pair problem using logical operators. Specifically, to enforce the logical OR operation $\vee$ for $s_* = s_1 \vee s_2$, we will use the below equivalent Ising [20]

$$\frac{1}{4}\left(3\mathbb{I} - \sigma_1^z - \sigma_2^z + 2\sigma_*^z + \sigma_1^z\sigma_2^z - 2\sigma_1^z\sigma_*^z - 2\sigma_2^z\sigma_*^z\right),$$

and to enforce $s_1 \le s_2$, e.g., the constraints in Eq. (2), we will convert into

$$\frac{1}{4}(\mathbb{I} - \sigma_1^z + \sigma_2^z - \sigma_1^z\sigma_2^z).$$

Let $N_t = |\mathbb{P}_t|$ and let (i,j) be the k^{th} pairs in $\mathbb{P}_t$ for $k = 1 \ldots N_t$. We also write $p_t^{(k)}$ in place of p_{ij}. The constraint in Eq. (7) can be encoded as

$$\bigvee_{(i,j)\in\mathbb{P}_t} p_{ij} = 1, \tag{8}$$

or, equivalently,

$$s_t^{(i)} = \begin{cases} p_t^{(1)}, & \text{if } i = 1, \\ s_t^{(i-1)} \vee p_t^{(i+1)}, & \text{if } 2 \le i \le N_t - 1, \\ 1, & \text{if } i = N_t. \end{cases}$$

where $s_t^{(i)}, i = 1, \ldots, N_t$ are auxiliary binary variables to break the long $\vee$ operator in Eq. (8) into elementary $\vee$ operator that can be translated into Ising Hamiltonian.

Denote by $\mathbb{T}$ the set of triples $x = y \vee z$ needed to encode the above long logical $\vee$ operator. We convert the Ising Hamiltonian in [20] into an equivalent QUBO, minus constant terms, as follows

$$\begin{aligned} Q_{SCP} = \sum_{j=1}^{m} x_j + \lambda_1 \sum_{(x,y,z)\in\mathbb{T}} (x + y + z + xy - 2yz - 2zx) \\ + \lambda_2 \sum_{1\le i<j\le m} (2p_{ij} - p_{ij}x_i - p_{ij}x_j). \end{aligned} \tag{9}$$

C. QUBO for HIPP through k-binding (QHI^k)

We present a new QUBO formulation for ILP$_G$, aiming to simultaneously reduce the formulation size and improve the solution quality. For an integer $k \ge 2$, we will divide variables in constraint (3) into groups of size at most k. For each $t = 1, \ldots, n$ constraint (3) $\sum_{i=1}^{N_t} p_t^{(i)} = 1$ will be transformed into a set of roughly $\frac{N_t}{k-1}$ equalities.

$$s_t^{(i)} = \begin{cases} p_t^{(1)} + \ldots + p_t^{(k)}, & \text{if } i = 1, \\ s_t^{(i-1)} + \sum_{j=(i-1)(k-1)+2}^{i(k-1)+1} p_t^{(j)} & \text{if } 2 \le i \le \lceil \frac{N_t}{k-1} \rceil \\ 1 & \text{if } i = \lceil \frac{N_t}{k-1} \rceil \end{cases} \tag{10}$$

where $s_t^{(i)}$ are binary auxiliary variables. Remark that this formulation is possible due to the equalities in the constraint (3) in ILP$_G$. The same approach would *not work for the ILP of SCP* as the value of $s_t^{(i)}$ can be as large as k in that case.

Each equality of the form $s = \sum_{i=1}^{k} y_i$ is then transformed into penalty as

$$\lambda_1\Big(\big(s - \sum_{i=1}^{k} y_i\big)^2 - \sum_{1\le i<j\le k} y_i y_j\Big). \tag{11}$$

The extra term $\sum_{1\le i<j\le k} y_i y_j$ allows a zero-penalty when either one or two of y_i equal one, by setting $s = 1$. This relaxes the constraints and leads to a higher probability of finding optimal solutions in our experiments.

Let $\mathbb{T}_k = \{(s, \mathbf{y})\}$ denote the set of equalities of form $s = y_1 + y_2 + ... + y_k$ in Eq. (10). Our proposed QUBO formula, QHI^k, can be written as

$$\begin{aligned} \sum_{j=1}^{m} x_j + \lambda_1 \sum_{(s,\mathbf{y})\in\mathbb{T}_k} \Big(\big(s - \sum_{i=1}^{k} y_i\big)^2 - \sum_{1\le i<j\le k} y_i y_j\Big) \\ + \lambda_2 \sum_{1\le i<j\le m} (2p_{ij} - p_{ij}x_i - p_{ij}x_j). \end{aligned} \tag{12}$$

We show below the minimum penalties to preserve optimal solutions in the proposed QUBO QHI^k.

Lemma 1. *For $\lambda_1 > 2$ and $\lambda_2 > 2$, any optimal solution $(\mathbf{x}, \mathbf{p}, \mathbf{s})$ of the QUBO in Eq. (12) induces an optimal solution $(\mathbf{x})$ for Haplotype Inference problem in Eq. 1 and vice versa.*

We omit the proof due to the space limit.

Variable Reduction. We apply the haplotype reduction technique in [1] to reduce the QUBO size. For each set $\mathbb{P}_t$, we remove all of p_{ij} (except one if it makes $\mathbb{P}_t$ empty) such that both haplotype h_i and h_j can only create genotype g_t.

IV. Experiments

We perform experiments to compare the effectiveness of different QUBO formulations and the performance of QA versus its classical counterpart SA for HIPP.

A. Setup

System. We present experimental results on D-Wave Advantage [21], the latest quantum annealer from D-Wave. The D-Wave Advantage system 4.1, an AQC with 5627 qubits, is based on Pegasus architecture. Each qubit is connected via internal coupling to 12 other qubits while has 3 connections for two external couplers and one odd couplers. For the simulated annealing, we run on Intel Core i7 (2.60GHz) machine.

Datasets. The dataset containing 20 instances is generated using Gusfield's method [1]. First, we generate $m' = 10$ 'ground truth' haplotypes for each test using Hudson's software [29]. Secondly, the haplotypes are randomly paired to create a random set of $n \in (m'/2, 2m']$ genotypes data by the coalescent process, modeling a reproduction rate smaller than 2.0 [24]. From the generated genotypes, we find all possible candidate haplotypes.

Methods. We compare three QUBO formulation methods presented in Section III, namely, DI, SCP, and QHI^k, with $k = 2$ as the default value. We run each formulation with two different QUBO solvers, simulated annealing, and quantum annealing. Both are from D-Wave's Ocean SDK.

Parameters. We use the default minorminer in D-Wave's Ocean SDK to find the minor-embedding of QUBO onto the Pegasus hardware graph. The chain strength prefactor is reduced to 0.15 to lower the noise level. For the D-Wave annealer, we set the annealing time to $150\mu s$ with 1000 samples, i.e., a total annealing time of $0.15s$. For the simulated annealing, we set the number of sweeps equal to 100 and the number of runs to 1000.

Postprocess and Error-correction. The annealing results do not always satisfy all the constraints, meaning some haplotypes must be added to explain all genotypes. Our post-process finds the second haplotype if the first haplotype is already in the set. Otherwise, we pick an arbitrary pair of haplotypes. Not only does the postprocessing provide an effective error correction but also enables the lowering of the penalties λ_1 and λ_2 to the minimum values of 2.0 for all QUBO formulations.

Metrics. For evaluation of the formulation's size and the embedding, we report the problem size ($m+N$- the number of haplotypes plus the number of pairs), the number of variables (#Var), and the number of non-zeros (#NonZ) in the QUBO formulations; the number of qubits (#Qubits) and the max chain length (MaxLen).

For the evaluation of solution quality and runtime, we have the optimal gap - the ratio between the best solution and the optimal one, the percentage of finding at least one optimal solution in one run (%Solve), the percentage of finding an optimal solution in each run (%Opt) and the average time to obtain the first optimal solution (TTS).

B. Results

a) Comparison between DI, SCP, and QHI^k: We show #Qubits, %Solve, and %Opt with quantum annealing using SCP, DI, QHI^2 in Table I. DI requires more physical qubits than SCP. The number of qubits in QHI^2 is on average 15% less than the other two due to the variable reduction technique. In terms of success probability, the fraction of test cases in which optimal solutions are found, QHI^2 find optimal solutions in all 100% test cases while SCP and DI can find optimal solutions in 95% and 10% of test cases, respectively.

Formulation	DI	SCP	QHI^2
#Var	**118.9**	165.4	142.0
#NonZ	573.8	418.2	**365.6**
#Qubits	248.2	240.5	**206.5**
%Solve	10.0	95.0	**100.0**
%Opt	0.08	6.73	**10.17**

TABLE I: Comparision between DI, SCP, and QHI^2

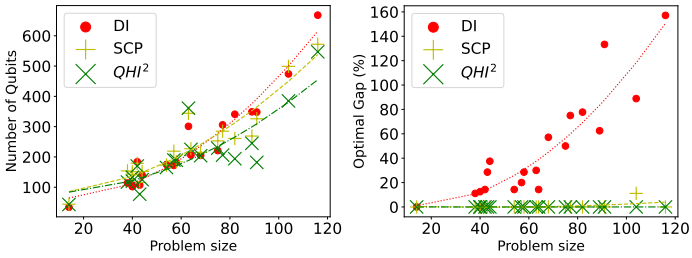

Fig. 2: Comparison of (Left) Number of physical qubits and (Right) The optimality gaps for each test case.

Formulation	Penalty	1	2	3	4
QHI^2	#Vio.†	4.27	0.025	0.0007	0
	%Solve	95	100	100	90
	%Opt	49.63	49.92	13.49	6.79

TABLE II: Relation of penalty values and solution quality using Simulated Annealing. †#Vio. is the average number of violated constraints.

b) Setting Penalties: As shown in Table II, the success probability decreases as the penalties go higher. The best penalty value is 2, the minimum value to guarantee the optimal QUBO formulation inducing an optimal solution of HIPP. Setting penalty factors λ_1 and λ_2 less than 2 results in violation of constraints and lower success probability.

k	2	3	4	5	6
#Var	142.0	127.7	120.2	116.9	**115.9**
#NonZ	365.6	**354.6**	359.0	366.6	377.5
#Qubits	206.5	197.2	190.7	**184.8**	190.4
MaxLen	**3.8**	4.0	4.0	3.9	4.1

TABLE III: Comparison of the number of variables (#Var), non-zeros (#NonZ), and qubits (#Qubits) with QHI^k.

c) Group size k for QHI^k: We report QUBO sizes of different k values for QHI^k in Table III. The minimum number of terms used in formulations is 354.6 with QHI^3. As expected, k increase will reduce the number of variables. However, it is surprising that the number of physical qubits increases when $k \geq 4$. This indicates the complexity associated with higher k makes it harder to embed the QUBO onto the hardware graph.

d) Quantum Annealing vs. Simulated Annealing: We report TTS for quantum annealing and simulated annealing with the formulations QHI^2 in Table 3. Simulated annealing's

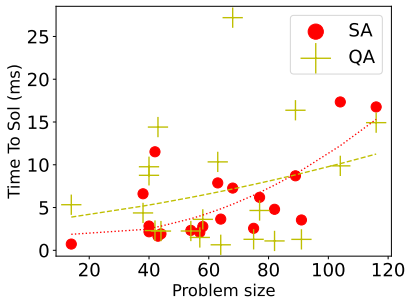

Fig. 3: Time to solution (ms) of Simulated Annealing (SA) and Quantum Annealing (QA) with QHI^2

TTS is on average 5.67 ms, while the annealing time for QA is 7.11 ms. However, the curve shows that QA runs faster on larger instances, indicating that we need to investigate more to verify the quantum speedup for this problem. The time to solution t_{sol} for QA takes into account the initial programming time (t_{prog}) for each problem; the annealing time ($t_{annealing}$) and the readout time ($t_{readout}$) for each sample. Therefore, it does not include the access time and the time to embed the QUBO into the hardware graphs.

V. Conclusion

We investigate three QUBO formulations for the HIPP problem. Both the QUBO in [20] and our proposed QHI^k method outperform the direct conversion of the ILPs for HIPP [22]. Overall, our proposed method QHI^k gives the best success probability in finding optimal solutions while requiring fewer qubits. Compared to classical simulated annealing, quantum annealing has shown promising speed-up, ignoring communication and minor embedding costs. Future development in both hardware and software for quantum annealing can further expand this gap towards a true quantum advantage.

Acknowledgment

We thank the anonymous reviewers for their suggestions and feedback. This research was supported in part by the US NSF Grants AMPS-2229075, AMPS-2229073, and CNS-2103405, VCU Quest Award, and VNU UET project number CN23.13. We thank Duc Dong Do for his assistance in generating data.

References

[1] D. Gusfield, "Haplotype inference by pure parsimony," in *Annual Symposium on Combinatorial Pattern Matching*. Springer, 2003, pp. 144–155.

[2] B. V. Halldórsson, V. Bafna, N. Edwards, R. Lippert, S. Yooseph, and S. Istrail, "A survey of computational methods for determining haplotypes," in *RECOMB Workshop on Computational Methods for SNPs and Haplotype Inference*. Springer, 2002, pp. 26–47.

[3] G. Lancia and R. Rizzi, "A polynomial case of the parsimony haplotyping problem," *Oper. Research Let.*, vol. 34, no. 3, pp. 289–295, 2006.

[4] P. Bertolazzi, A. Godi, M. Labbé, and L. Tininini, "Solving haplotyping inference parsimony problem using a new basic polynomial formulation," *Comp. & Math. w. App.*, vol. 55, no. 5, pp. 900–911, 2008.

[5] I. Lynce and J. Marques-Silva, "Haplotype inference with boolean satisfiability," *Int. J. on Artificial Intelligence Tools*, vol. 17, no. 02, pp. 355–387, 2008.

[6] A. Graça, J. Marques-Silva, I. Lynce, and A. L. Oliveira, "Efficient haplotype inference with combined cp and or techniques," in *Integration of AI and OR Techniques in Constraint Prog. for Comb. Opt. Problems: 5th Int. Conf., Proc. 5*. Springer, 2008, pp. 308–312.

[7] L. Di Gaspero and A. Roli, "Stochastic local search for large-scale instances of the haplotype inference problem by pure parsimony," *Journal of Algorithms*, vol. 63, no. 1-3, pp. 55–69, 2008.

[8] D. D. Do, S. V. Le, and X. H. Hoang, "Acohap: an efficient ant colony optimization for the haplotype inference by pure parsimony problem," *Swarm Intelligence*, vol. 7, pp. 63–77, 2013.

[9] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[10] T. Honjo, T. Sonobe, K. Inaba, T. Inagaki, T. Ikuta, Y. Yamada, T. Kazama, K. Enbutsu, T. Umeki, R. Kasahara *et al.*, "100,000-spin coherent ising machine," *Science advances*, vol. 7, no. 40, 2021.

[11] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, "Noisy intermediate-scale quantum algorithms," *Rev. of Mode. Phys.*, vol. 94, no. 1, p. 015004, 2022.

[12] A. Mills, C. Guinn, M. Gullans, A. Sigillito, M. Feldman, E. Nielsen, and J. Petta, "Two-qubit silicon quantum processor with operation fidelity exceeding 99%," *arXiv preprint arXiv:2111.11937*, 2021.

[13] X. Wang, C. Xiao, H. Park, J. Zhu, C. Wang, T. Taniguchi, K. Watanabe, J. Yan, D. Xiao, D. R. Gamelin *et al.*, "Light-induced ferromagnetism in moiré superlattices," *Nature*, vol. 604, no. 7906, pp. 468–473, 2022.

[14] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse ising model," *Physical Review E*, vol. 58, no. 5, p. 5355, 1998.

[15] D. M. Fox, C. M. MacDermaid, A. M. Schreij, M. Zwierzyna, and R. C. Walker, "Rna folding using quantum computers," *PLOS Compu. Bio.*, vol. 18, no. 4, p. e1010032, 2022.

[16] S. Yarkoni, A. Alekseyenko, M. Streif, D. Von Dollen, F. Neukart, and T. Bäck, "Multi-car paint shop optimization with quantum annealing," in *QCE Int.Conf.* IEEE, 2021, pp. 35–41.

[17] F. Neukart, G. Compostella, C. Seidel, D. Von Dollen, S. Yarkoni, and B. Parney, "Traffic flow optimization using a quantum annealer," *Frontiers in ICT*, vol. 4, p. 29, 2017.

[18] M. Kim, D. Venturelli, and K. Jamieson, "Leveraging quantum annealing for large mimo processing in centralized radio access networks," in *SIGCOMM '19: ACM SIGCOMM Int.Conf*, 2019, pp. 241–255.

[19] F. Glover, G. Kochenberger, and Y. Du, "A tutorial on formulating and using qubo models," *arXiv preprint arXiv:1811.11538*, 2018.

[20] Y. Cao, S. Jiang, D. Perouli, and S. Kais, "Solving set cover with pairs problem using quantum annealing," *Scientific reports*, vol. 6, no. 1, p. 33957, 2016.

[21] C. McGeoch and P. Farré, "The advantage system: Performance update," https://www.dwavesys.com/media/kjtlcemb/14-1054a-a_advantage_system_performance_update.pdf, accessed May 6, 2022.

[22] D. Gusfield, "Inference of haplotypes from samples of diploid populations: complexity and algorithms," *Journal of computational biology*, vol. 8, no. 3, pp. 305–323, 2001.

[23] G. Lancia, M. C. Pinotti, and R. Rizzi, "Haplotyping populations by pure parsimony: Complexity of exact and approximation algorithms," *INFORMS Journal on computing*, vol. 16, no. 4, pp. 348–359, 2004.

[24] D. G. Brown and I. M. Harrower, "Integer programming approaches to haplotype inference by pure parsimony," *IEEE/ACM Trans. on Comp. Biology and Bioinformatics*, vol. 3, no. 2, pp. 141–154, 2006.

[25] V. Choi, "Minor-embedding in adiabatic quantum computation: I. the parameter setting problem," *Quantum Inf. Pro.*, vol. 7, no. 5, pp. 193–209, 2008.

[26] A. Lucas, "Ising formulations of many np problems," *Frontiers in physics*, vol. 2, p. 5, 2014.

[27] L. B. Gonçalves, S. de Lima Martins, L. S. Ochi, and A. Subramanian, "Exact and heuristic approaches for the set cover with pairs problem," *Optimization Letters*, vol. 6, pp. 641–653, 2012.

[28] R. Hassin and D. Segev, "The set cover with pairs problem," in *International Conference on Foundations of Software Technology and Theoretical Computer Science*. Springer, 2005, pp. 164–176.

[29] R. R. Hudson, "Generating samples under a wright–fisher neutral model of genetic variation," *Bioinformatics*, vol. 18, no. 2, pp. 337–338, 2002.