Quantum Social Computing Approaches for Influence Maximization

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

An Nguyen, Uyen Nguyen
Inst. of Eng. and Tech.
Vinh Univ., Nghe An, Vietnam
{annc, uyennt}@vinhuni.edu.vn

Giang Nguyen
Inst. of Info. Technology
Vietnam Acad. of Sci. and Tech., Ha Noi
nlgiang@ioit.ac.vn

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10000698

***Abstract*—Influence Maximization (IM), which seeks a small set of important nodes that spread the influence widely into the network, is a fundamental problem in social networks. It finds applications in viral marketing, epidemic control, and assessing cascading failures within complex systems. Despite the huge amount of effort, finding near-optimal solutions for IM is difficult due to its NP-completeness. In this paper, we propose the *first social quantum computing approaches* for IM, aiming to retrieve near-optimal solutions. We propose a two-phase algorithm that 1) converts IM into a Max-Cover instance and 2) provides efficient quadratic unconstrained binary optimization formulations to solve the Max-Cover instance on quantum annealers. Our experiments on the state-of-the-art D-Wave annealer indicate better solution quality compared to classical simulated annealing, suggesting the potential of applying quantum annealing to find high-quality solutions for IM.**

***Index Terms*—Influence Maximization, Social Networks, Quantum Annealing**

I. Introduction

The prevalence of social networks has made them a leading channel for *information diffusion*, spreading various kinds of information from opinions, and behaviors to fake news. The rapid propagation of information in this channel has also altered human societies at a fundamental level, as seen on many occasions. Examples include the critical role of social media in swaying public opinion in the 2016 U.S. presidential election [1] and the use of social media 'like a weapon' in the Ukraine war [2]. Thus, understanding the mechanisms behind information diffusion, such as how to engineer a viral campaign or mitigate fake news, is of uttermost importance.

In social networks, *influence maximization* (IM) problem [3] seeks for a small subset of nodes whose activation will result in the largest influence propagation in the network. For example, a company can send free sample products to an initial set of users (called seed nodes) who will advertise the product via the word-of-mouth effect to create a viral marketing campaign. The problem, proposed by Kempe et al. [3], was investigated on two pioneering diffusion models, namely, Independent Cascade (IC) and the Linear Threshold (LT) models that capture social-psychological factors in the adoption process [3]. It has remained one of the most well-studied problems in social networks. Many of its applications can be found in viral marketing and political campaigning [4], outbreak detection [5], and rumor mitigation [6].

Corresponding author: tndinh@vcu.edu

Despite a plethora amount of research on IM [7]–[13], obtaining *near-optimal solutions for the problem remains a difficult task* due to several theoretical challenges. Kempe et al. prove the NP-hardness of the problem [3] and propose $(1 - 1/e - \epsilon)$ approximation algorithms using Monte-Carlo simulation [3]. Indeed, IM cannot be approximated within a factor $(1-1/e+\epsilon)$ [14] under a typical complexity assumption. Further, computing the exact influence is shown to be #P-hard [8]. A few works attempt to obtain exact and near-optimal solutions for IM [15], [16], however, they all have exponential time complexities in the worst-case.

The last few years have witnessed an exponential growth in *quantum and quantum-inspired computing* with a record number of breakthroughs [17]–[21]. Instead of encoding information with binary bits as in classical computing, quantum computing (QC) utilizes qubits to encode superposition of states [19] 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 [17], [19]. The two major paradigms in quantum computing are gate-based QC and adiabatic quantum computation (AQC) [22]. AQC has been shown to be equivalent to gate-based QC with at most polynomial overhead [23]. In AQC, NP-hard optimization problems are encoded into Ising Hamiltonians, whose ground states induce optimal solutions.

One promising near-term avenue for QC is *quantum annealing* (QA) [24], which can be seen as a relaxation of AQC, where the condition of adiabaticity is relaxed. QA is the only computing approach that provides a large enough number of qubits for real-world problems from life science [25], portfolio optimization [26], scheduling for car manufacturing [27] and many others [28], [29]. Except for a few recent works in community detection using quantum computing [30]–[32], there is a largely untapped opportunity in applying QC for social networks. Such *quantum social computing* approaches can lead to a potential boost in finding near-optimal solutions for many social network tasks including IM.

In this paper, we provide the first quantum social computing approach for IM, investigating the feasibility of obtaining near-

optimal solutions for IM on state-of-the-art quantum annealers [33]. We propose a novel two-phase approach that 1) converts the IM into a maximum coverage (Max-Cover) instance using reverse influence sketch (RIS) proposed in [34] and 2) efficient quadratic unconstrained binary optimization (QUBO) formulations and robust parameter settings to solve the problem on quantum annealers. The first phase provides a *space-efficient representation* of the network's influence landscape that is critical due to the limitation on the number of qubits on the existing quantum solvers. The second phase addresses the challenges in implementing all-to-all coupling [33] among the qubits, i.e., sparse QUBO formulations are easier to program on existing quantum processing units (QPUs). Finally, we provide comparisons between QA, on D-Wave Advantage, the latest D-Wave annealer [33], and the classical simulated annealing (SA) in terms of solution quality. Further, we analyze the efficiency of the proposed formulations in terms of the number of qubits and solution quality.

Related works. Kempe et al. [3] proposed two fundamental cascade models, namely Linear Threshold (LT) and Independent Cascade (IC) models, for IM problem. They showed that IM is an NP-hard problem and proposed a greedy algorithm with the ratio of approximations as $1-1/e-\epsilon$. The lazy greedy approach was proposed in [5] to avoid the recomputation of marginal gains for nodes at each iteration. Many heuristics have been proposed to solve IM in large networks such as LDAG [35], CELF++ [36], and so on.

Several sketching methods have been introduced to solve IM in very large networks. Cohen et al. [37] investigate the combined reachability sketch using bottom-k min-hash sketch of the best reachable nodes to show small errors in estimating influences. Borgs et al. [34] present a pioneer approach called reverse influence sketch (RIS) that estimates the influence of any seed set S by taking random nodes and measures the fraction of those nodes reachable by S. The RIS has inspired a collection of large-scale methods for IM such as TIM [38], IMM [39], SSA/DSSA [12], and other [13], [40], [41].

QA has a wide range of applications [25]–[29]. However, the only quantum social computing approaches are on the community detection problem [30]–[32]. To the best of our knowledge, there is no quantum computing method for IM. There is a generic approach for solving Linear Integer Programming, of which maximum coverage is a special case, on quantum annealers [42]. However, such a generic approach unlikely performs well without taking into account the limits in existing QPUs including the limited connectivity and the limited precious in coupling strength among the qubits. Thus, effective formulations are needed for maximum coverage instances generated from IM.

Organization. We introduce background on IM and QA in section II. Section III proposes our two-phase algorithm including RIS samples generation and QUBO formulations for maximum coverage problem. The experiment results are shown in section IV. Finally, section V presents our conclusion and future directions.

II. Preliminaries

We present the preliminaries on IM and solving optimization with QA.

A. Influence Maximization Problem

Given graph $G = (V, E, w)$ and an integer $1 \leq k \leq |V|$, the Influence Maximization (IM) problem asks for a subset $S \subset V$ of at most k nodes that maximize the influence $\mathbb{I}(S)$, defined as the number of active nodes at the end of a diffusion process started at S. For simplicity, we adopt Independent Cascade (IC) as the propagation model [3]. However, our approach can be extended easily to Linear Threshold (LT) model [3].

Independent cascade model. In the IC model, each edge $(u, v) \in E$ is associated to a weight $w(u, v) \in [0, 1]$ that indicates the probability that u influences v. By convention, if $(u, v) \notin E$, we also set $w(u, v) = 0$. Let $S \subseteq V$ be a subset of nodes, called seed nodes. The activation happens in rounds. At round zero, all the nodes in S are active, and all the others are inactive. In each round, each newly activated node u in the previous round will have a single chance to activate its neighbors: for each neighbor v of u, v becomes active with the probability $w(u, v)$. Once a node gets activated, it will remain active till the end. The process stops when there are no more newly activated nodes in a round.

B. Quantum Annealing and QUBO

Quantum Annealing (QA) provides an approach to find near-optimal solutions for NP-hard problems that can be encoded into a quadratic unconstrained binary optimization (QUBO) or, equivalently, an Ising Hamiltonian [42]. A QUBO minimizes a quadratic polynomial over binary variables

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

where $\mathbf{x} = (x_1, \cdots, x_n) \in \{0, 1\}^n$. Since, QUBO is an NP-complete problem, there are polynomial-time reductions to reduce all NP-complete problems to QUBO.

By changing variables $x_i = \frac{s_i+1}{2}$, a QUBO can be easily converted back and forth to an Ising Hamiltonian [43]

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

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 find the lowest energy state, called ground state, of the Hamiltonian.

The process of solving the NP-hard problems by the QA method consists of three major steps: transform the optimization problem to a QUBO, embed the QUBO graph to QPU's hardware graph, and perform an annealing schedule to find ground states. First, we need to transform the optimization problem into a QUBO formulation, the de facto input format for quantum annealing. The QUBO is immediately turned into

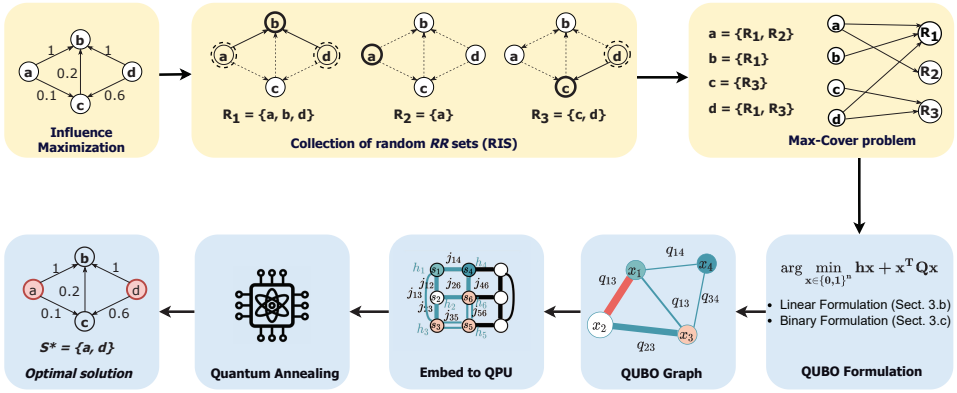


Fig. 1. Steps to solve Influence Maximization (IM) using quantum annealing. In phase 1 (yellow blocks), we convert an IM problem into Max-cover by generating random RR sets. In phase 2 (blue blocks), we propose QUBO formulations to find the optimal solutions for Max-Cover using quantum annealing.

a logical graph after definition, with each node representing a variable and each edge denoting the interaction term between two variables. In formulating QUBO, it is important that *the range of J_{ij} does not exceed the hardware precision*. Otherwise, rounding errors will result in a programmed QUBO that is different from the intended one [33].

Minor-embedding. Since, all-to-all connectivity among all qubits is difficult to achieve on QPUs, we need to perform a mapping between logical variables to physical qubits in a procedure called *minor-embedding*. Each logical qubits may be mapped into multiple physical qubits. Those qubits will be coupled with sufficient strong interactions. Setting the strength of those interactions, known as *chain strength*, is critical to guarantee that all those physical qubits will obtain the same values. Since, the hardware graph in existing quantum annealers are sparse graphs, minor-embedding of complete QUBO will require a quadratic number of qubits [43]. Thus, to limit the exploding of the required number of qubits, a *sparse QUBO* is often desired.

III. Quantum Social Computing for IM

We introduce a new quantum annealing approach to solve IM. Our two-phase approach, shown in Fig. 1 consists of 1) converting IM to maximum coverage (Max-Cover) [44] by applying the RIS sample [34] and 2) constructing efficient QUBO formulations for Max-Cover and robust parameter settings for QA.

A. Phase 1 - Converting to Max-Cover using Reverse Influence Sketch (RIS)

Most existing approaches for IM including heuristics [35], [36] and forwarding Monte-Carlo simulation [7], [45], [46] to estimate influence are too complex to transform into QUBO formulations. Fortunately, the pioneering approach of using RIS [34] results in Max-Cover instances [44] that are simple enough to program on quantum annealers.

Given a graph $G = (V, E, w)$, RIS captures the influence landscape of G by generating a collection $\mathcal{R}$ of random *Reverse Reachable (RR) sets*. A random RR set $R_j,\ j = 1..n$ is generated from G as follows

1) Choose a random node $v \in V$ as a target node,
2) Generate a sample graph g from G in which we keep each edge $(u, v) \in E$ with a probability $w(u, v)$ and discard (u, v) otherwise, and
3) Return R_j as the set of nodes that can reach to v in g.

The second step can be replaced by a reverse graph traversal algorithm from v [34], revealing the existence of edges only when needed. Once a collection $\mathcal{R}$ of sufficiently large RR sets has been computed, the influence of any seed set S can be approximated by the fraction of RR sets that have at least one node in S. Thus, IM can be solved by finding a subset of k nodes that 'covers' most of the generated RR sets. This problem is the well-known Max-Cover problem (MC) [44].

Max-Cover [44]. Given a collection of subsets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ over am universal set of elements $E = \{e_1, e_2, \ldots, e_n\}$. The goal is to find a subset $S \subseteq \mathcal{S}$ of sets, such that $|S| \leq k$ and the number of covered elements $|\bigcup_{S_i \in S'} S_i|$ is maximized. Max-Cover is an NP-hard problem [44] and cannot be approximated within a factor $1 - 1/e + \epsilon$, unless P=NP [14].

An example of the conversion into Max-Cover is shown in Fig. 1. Nodes b, a, and c are selected at the target nodes, respectively. The *RR* sets are generated by finding the nodes that can reach to the target nodes, i.e., $\mathcal{R} = \{R_1 = \{a, b, d\}, R_2 = \{a\}, R_3 = \{c, d\}\}$. Finally, we construct an instance of Max-Cover problem with four sets including S_a = $\{R_1, R_2\}$, S_b = $\{R_1\}$, S_c = $\{R_3\}$, and S_d = $\{R_1, R_3\}$. The optimal solution for $k = 2$ is to select $S = \{a, d\}$ to cover all RR sets. That also returns the optimal seed set for the IM instance.

B. Phase 2 - Efficient QUBO formulations for Max-Cover

In phase 2, we take the Max-Cover instance produced in phase 1 and transform it into a QUBO formulation (Fig. 1). The QUBO formulation, can be seen as a graph, will be mapped into physical qubits on the QPU, and solved through an annealing process to obtain near-optimal solutions.

We begin by presenting the Integer Linear Programming for Max-Cover problem in [44]. Then, we propose two QUBO formulations: 1) Linear formulation that adds a linear, in terms of element frequencies, number of slack variables and 2)

Binary formulation that adds a logarithmic number of slack variables using binary encoding [42].

a) Integer Linear Programming for Max-Cover: Let x_i be binary variables for $1 \leq i \leq m$ denoting whether or not subset S_i is selected. In other words, $x_i = 1$ iff S_i is selected. In addition, $y_j = 1$ if e_j is covered, and $y_j = 0$ otherwise for $1 \leq j \leq n$. The Max-Cover objective is to maximize the sum of y_j as shown in the Integer Linear Program (ILP) formulation in Eq. 2.

$$\begin{aligned} \max \ & \sum_{j=1}^{n} y_j \\ s.t. \ & \sum_{i=1}^{m} x_i = k \\ & \sum_{e_j \in S_i} x_i \geq y_j, \ j = 1..n \\ & \mathbf{x} \in \{0,1\}^m, \mathbf{y} \in \{0,1\}^n \end{aligned} \tag{2}$$

b) QUBO Linear Formulation: First, we convert inequality constraints to equalities by adding slack variables s_j

$$y_j - \sum_{e_j \in S_i} x_i + s_j = 0, \ j = 1..n \tag{3}$$

where $f_j = |\{S_i | e_j \in S_i\}$ is the frequency of e_j, i.e., the number of times that e_j appears in the subsets and $s_j \in \{0, 1, \ldots, f_j\}$.

In the second constraint, the upper bound of s_j depends on the number of sets containing the element e_j in Eq. 4

$$\sum_{e_j \in S_i} x_i - y_j = \sum_{i=1}^{f_j - 1} z_{ij}, z_{ij} \in \{0,1\}, j = 1..n, z. \tag{4}$$

To limit the number of equivalent optimal solutions, we add a penalty $\sum_{i=1}^{n-1}(z_i - z_i z_{i+1}) = 0$ to the objective (Wall-encoding). The domain-wall encoding improves the solution quality as investigated in [47].

We formulate a base QUBO Linear formulation as follow

$$\begin{aligned} \min f(\mathbf{x}, \mathbf{y}, \mathbf{z}) = & -\sum_{j=1}^{n} y_j \\ & + P_1 \left(\sum_{i=1}^{m} x_i - k \right)^2 \\ & + P_2 \sum_{j=1}^{n} \left(y_j - \sum_{e_j \in S_i} x_i + \sum_{i=1}^{f_j - 1} z_{ij} \right)^2 \\ & + \sum_{i=1}^{f_j - 1} (z_{i+1} - z_i z_{i+1}) \end{aligned} \tag{5}$$

where $P_1, P_2 > 0$ are penalties.

Settings the penalties values is important to obtain high-quality solutions. On the one hand, the penalties need to be sufficiently large to preserve the optimal solutions. On the other hand, the penalties should not be set too large or they will exceed the precision limit of the QPUs. We show below the feasible range for the penalties.

Lemma 1. *For $P_1 > \max_j f_j$ and $P_2 > 1$, any optimal solution $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ of the QUBO in Eq. (5) induces an optimal solution $(\mathbf{x}, \mathbf{y})$ for Max-Cover in Eq 2 and vice versa.*

We omit the proof due to the space limit.

Despite the simplicity of the linear formulation, it has one advantage that most interaction J_{ij} are in $\{-1, 1\}$, i.e., it reduces the requirement on the coupling precision of the quantum annealers.

We continue with the analysis of the sparsity of the proposed formulation.

Lemma 2. *QUBO Linear formulation (LF) in Eq 5 contains $n_L = m + \sum_{j=1}^{n} f_j$ variables and $\frac{1}{2}\left(m^2 + 3\sum_{j=1}^{n} f_j^2 + n_L\right)$ non-zeros.*

Proof. The number of variables, summing up the length of $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ is

$$\begin{aligned} n_L &= m + n + \sum_{j=1}^{n-1} f_j = m + n + \sum_{j=1}^{n} f_j - n \\ &= m + \sum_{j=1}^{n} f_j \end{aligned} \tag{6}$$

To count the number of non-zeros, we note that the non-zeros on the diagonals is n_L, the number of variables. Thus, the rest is to count the number of off-diagonal non-zeros.

Summing up the number of off-diagonal terms in each 'constraint' in Eq. 5, the number of non-zeros is at most

$$m + \sum_{j=1}^{n} f_j + \frac{m(m-1)}{2} + \frac{1}{2}\sum_{j=1}^{n} f_j(3f_j - 1) \tag{7}$$

$$= \frac{1}{2}\left(2m + 2\sum_{j=1}^{n} f_j + m^2 - m + 3\sum_{j=1}^{n} f_j^2 - \sum_{j=1}^{n} f_j\right) \tag{8}$$

$$= \frac{1}{2}\left(m^2 + 3\sum_{j=1}^{n} f_j^2 + n_L\right) \tag{9}$$

This yields the proof. □

c) Binary Formulation: QUBO with Binary Encoding: We provide a more 'compact' formulation, at the expense of higher interaction J_{ij} values.

For $f_j > 1$, we replace the slack variables in the linear formulation with $N_j = \lfloor \log(f_j - 1) \rfloor + 1$ new binary variables, following the binary encoding in [42], as shown in the Eq 10.

$$\begin{aligned} 0 \leq s \leq f_j \Rightarrow s = & \sum_{k=0}^{N_j - 1} 2^k z_{k_j} + (f_j + 1 - 2^{N_j}) z_j^{(N_j)}, \\ & z_{k_j} \in \{0,1\} \end{aligned} \tag{10}$$

TABLE I
COMPARING SIMULATED ANNEALING AND QUANTUM ANNEALING USING LINEAR FORMULATION (LF) AND BINARY FORMULATION (BF)

n	Linear Formulation						Binary Formulation					
	var.	*qubit*	*inf.*		*time(s)*		*var.*	*qubit*	*inf.*		*time(s)*	
			SA	*QA*	*SA*	*QA*			*SA*	*QA*	*SA*	*QA*
10	157	266	**7.1**	**7.1**	0.36	0.1	152	254	7.0	**7.1**	0.35	0.1
15	158	269	**8.1**	**8.1**	0.38	0.1	151	247	7.95	**8.1**	0.34	0.1
20	182	449	9.4	10.0	0.45	0.1	171	341	9.2	**10.2**	0.42	0.1
25	173	390	10.5	**10.75**	0.41	0.1	167	324	10.0	10.25	0.42	0.1
30	185	529	11.7	**12.3**	0.46	0.1	185	529	11.7	11.1	0.43	0.1

This results in our new QUBO binary formulation

$$\min f(\mathbf{x},\mathbf{y},\mathbf{z}) = -\sum_{j=1}^{n} y_j + P_1\Big(\sum_{i=1}^{m} x_i - k\Big)^2 + P_2\sum_{j=1}^{n}\Big(y_j - \sum_{e_j\in S_i} x_i + \sum_{k=0}^{N_j-1} 2^k z_{k_j} + (f_j+1-2^{N_j})z_j^{(N_j)}\Big)^2 \tag{11}$$

Similarly, we can set the penalty values the same way we did for the linear formulation.

Lemma 3. *For $P_1 > \max_j f_j$ and $P_2 > 1$, any optimal solution $(\mathbf{x},\mathbf{y},\mathbf{z})$ of the QUBO in Eq. (11) induces an optimal solution $(\mathbf{x},\mathbf{y})$ for Max-Cover in Eq 2 and vice versa.*

This new binary formulation has substantially fewer variables non-zeros as shown in the below lemma.

Lemma 4. *QUBO Binary formulation (BF) in Eq 11 contains $n_B = m + n + \sum_{j=1}^{n} N_j$ variables and $\frac{1}{2}\left(m^2 + 2(N_j+1)\sum_{j=1}^{n} f_j + \binom{N_j}{2} + 2n_B - m\right)$ non-zeros where $N_j = \lfloor \log(f_j - 1)\rfloor + 1$ for $f_j > 1$ and 0 otherwise.*

The proof is similar to that of the linear formulation and skipped to save space.

IV. EXPERIMENTS

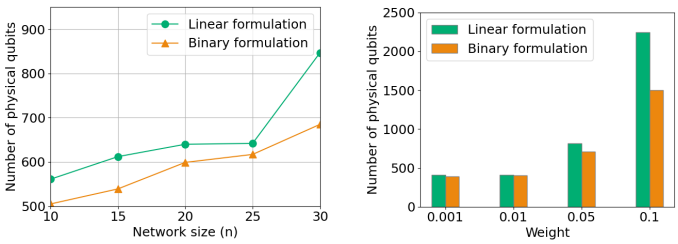
We present experimental results on D-Wave Advantage [33], the latest quantum annealer from D-Wave.

A. Setup

Datasets. We generate random networks following Erdos-Renyi model using networkx package [48]. The network size is chosen in $\{10, 15, 20, 25, 30\}$. The edge probability p is set so that the average degree of each node is 5. All edges are set to have the same weight (probability) and the weight is selected among the values $\{0.001, 0.01, 0.05, 0.1\}$.

Parameters. We set the penalty P_1 to be the number of samples $P_2 = 2$. The default parameters include the network size $n = 30$, the edge weight is $w = 0.05$, and the number of seed nodes $k = 5$.

Solvers. We compare two different solvers for QUBO, simulated annealing and quantum annealing, both from D-Wave's Ocean SDK. For the D-Wave annealer, we use the default annealing time of $20\mu s$ with 5000 samples, i.e., a total annealing time of $0.1s$. We set the chain strength to be the largest absolute value of J_{ij} in the QUBO. For the simulated annealing, we repeat 100 times and report the best result.



(a) $w = 0.05$ (b) $n = 30$

Fig. 2. Comparison of the number of qubits with QUBO Linear and QUBO Binary formulations

B. Results

Physical qubits. We measure the number of physical qubits, after embedding into D-Wave QPUs, for the two QUBO formulations. The results are shown in Fig. 2. As expected, the binary formulation consistently requires fewer qubits and the gap gets wider with the increase of the network size and the edge weight.

Solution Quality. We show the solution quality for simulated annealing and quantum annealing using two different formulations in Table I. We set the number of nodes $n = 10, \ldots, 30$, $k = 5$, the weights $w = 0.05$, and the number of RR sets is 100. The solution quality is measured by the estimated influence (Inf.) (the fraction of covered RR sets times n).

Overall, the solution quality, Inf., of QA is slightly better than those of SA for both linear and binary formulations. Both return the same Inf. for small networks, $n = 10, 15$. QA returns slightly higher Inf. for larger sizes, $n = 15, 20, 25, 30$. While linear formulation requires more variables and qubits, it frequently produces better solutions which can be attributed to the small weights in its QUBO.

Time. The running time for simulated annealing ranges from 0.3 to 0.5 seconds, while the annealing time for QA is 0.1 second. Notice that the running time for QA does not include the access time and the time to embed the QUBO into the hardware graphs. While the embedding time is currently high, the advance in algorithms and future hardware-software co-design for quantum annealers will bring down this overhead.

V. CONCLUSION

We propose the first quantum annealing approach for IM. The results demonstrate some advantage of quantum annealing over the classical simulated annealing on the problem. Thus, it advocates for further investigation of quantum computing approaches for IM and quantum social computing in general. Future directions include designing specialized annealing schedule, reverse annealing, customized minor-embedding approaches for the problem.

REFERENCES

[1] A. Bovet and H. A. Makse, "Influence of fake news in twitter during the 2016 us presidential election," *Nature communications*, vol. 10, no. 1, pp. 1–14, 2019.

[2] M. Specia, "'like a weapon': Ukrainians use social media to stir resistance," Mar 2022. [Online]. Available: https://www.nytimes.com/2022/03/25/world/europe/ukraine-war-social-media.html

[3] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD03: 9^{th} of ACM SIGKDD Int.Conf*, 2003, pp. 137–146.

[4] P. Domingos and M. Richardson, "Mining the network value of customers," in *KDD01: ACM SIGKDDg Int.Conf*, 2001, pp. 57–66.

[5] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *KDD07: The 13^{th} ACM SIGKDD Inf.Conf*, 2007, pp. 420–429.

[6] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *The 20^{th} WWW '11 Inf.Conf*, 2011, pp. 665–674.

[7] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *ACM KDD '07*. New York, NY, USA: ACM, 2007, pp. 420–429.

[8] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *ACM KDD '10*. New York, NY, USA: ACM, 2010, pp. 1029–1038.

[9] A. Goyal, W. Lu, and L. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *Proceedings of ICDM*. IEEE, 2011, pp. 211–220.

[10] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *Proceedings of CIKM*. ACM, 2014, pp. 629–638.

[11] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of ACM SIGMOD*. ACM, 2014, pp. 75–86.

[12] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," 2017. [Online]. Available: https://arxiv.org/abs/1605.07990v3

[13] J. Tang, X. Tang, X. Xiao, and J. Yuan, "Online processing algorithms for influence maximization," in *SIGMOD/PODS '18 Int. Conf*, 2018, pp. 991–1005.

[14] U. Feige, "A threshold of ln n for approximating set cover," *Journal of ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[15] X. Li, J. D. Smith, T. N. Dinh, and M. T. Thai, "Tiptop:(almost) exact solutions for influence maximization in billion-scale networks," *IEEE/ACM Trans. on Net.*, vol. 27, no. 2, pp. 649–661, 2019.

[16] G. Nannicini, G. Sartor, E. Traversi, and R. Wolfler Calvo, "An exact algorithm for robust influence maximization," *Math. Pro.*, vol. 183, no. 1, pp. 419–453, 2020.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] T. Albash and D. A. Lidar, "Adiabatic quantum computation," *Rev. of Mode. Phys.*, vol. 90, no. 1, p. 015002, 2018.

[23] D. Aharonov, W. Van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, "Adiabatic quantum computation is equivalent to standard quantum computation," *SIAM review*, vol. 50, no. 4, pp. 755–787, 2008.

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

[25] 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.

[26] C. Grozea, R. Hans, M. Koch, C. Riehn, and A. Wolf, "Optimising rolling stock planning including maintenance with constraint programming and quantum annealing," *arXiv preprint arXiv:2109.07212*, 2021.

[27] 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.

[28] 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.

[29] 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.

[30] F. G. Gemeinhardt, R. Wille, and M. Wimmer, "Quantum k-community detection: algorithm proposals and cross-architectural evaluation," *Quantum Inf. Pro.*, vol. 20, no. 9, pp. 1–21, 2021.

[31] R. Shaydulin, H. Ushijima-Mwesigwa, I. Safro, S. Mniszewski, and Y. Alexeev, "Network community detection on small quantum computers," *Advanced Quantum Technologies*, vol. 2, no. 9, p. 1900029, 2019.

[32] C. F. Negre, H. Ushijima-Mwesigwa, and S. M. Mniszewski, "Detecting multiple communities using quantum annealing on the d-wave system," *Plos one*, vol. 15, no. 2, p. e0227538, 2020.

[33] 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.

[34] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *SODA '14: ACM-SIAM Sym. on Dis. Alg. Int. Conf*, 2014, pp. 946–957.

[35] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *KDD '10: The 16^{th} ACM SIGKD*, 2010, pp. 1029–1038.

[36] "Celf++ optimizing the greedy algorithm for influence maximization in social networks," in *The WWW '11 Int.Conf*, 2011, pp. 47–48.

[37] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *CIKM '14: 2014 ACM Int.Conf*, 2014, pp. 629–638.

[38] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *SIGMOD/PODS'15 Int.Conf*, 2014, pp. 75–86.

[39] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *SIGMOD/PODS'15 Inf.Conf*, 2015, pp. 1539–1554.

[40] H. T. Nguyen, T. P. Nguyen, N. Phan, and T. N. Dinh, "Importance sketching of influence dynamics in billion-scale networks," in *2017 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2017, pp. 337–346.

[41] H. T. Nguyen, T. P. Nguyen, T. N. Vu, and T. N. Dinh, "Outward influence and cascade size estimation in billion-scale networks," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1, no. 1, pp. 1–30, 2017.

[42] F. Glover, G. Kochenberger, R. Hennig, and Y. Du, "Quantum bridge analytics i: a tutorial on formulating and using qubo models," *Annals of Oper. Res.*, pp. 1–43, 2022.

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

[44] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Information processing letters*, vol. 70, no. 1, pp. 39–45, 1999.

[45] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD'03*. ACM New York, NY, USA, 2003, pp. 137–146.

[46] A. Goyal, F. Bonchi, L. Lakshmanan, and S. Venkatasubramanian, "On minimizing budget and time in influence propagation over social networks," *Social Network Analysis and Mining*, pp. 1–14, 2012.

[47] J. Chen, T. Stollenwerk, and N. Chancellor, "Performance of domain-wall encoding for quantum annealing," *TQE*, vol. 2, pp. 1–14, 2021.

[48] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using networkx," in *The 7^{th} SciPy Inf.Conf*, G. Varoquaux, T. Vaught, and J. Millman, Eds., Pasadena, CA USA, 2008, pp. 11 – 15.