Quantum Annealing Approach for Selective Traveling Salesman Problem

Thinh V. Le*, Manh V. Nguyen*, Sri Khandavilli*, Thang N. Dinh†, and Tu N. Nguyen*
*Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA.
†Department of Computer Science, Virginia Commonwealth University, VA 23284 USA.
{tle93, mnguy126, skhandav}@students.kennesaw.edu, tndinh@vcu.edu, and tu.nguyen@kennesaw.edu

***Abstract*—Quantum computing has paved a new way for faster and more efficient solutions to large-scale, real-world optimization problems that are challenging for classical computing systems. For instance, selective traveling salesman problem (sTSP) that is famous in such fields as logistic optimization and has attracted increasing attention from the research community, however, is known as an NP-Hard problem. Solving the sTSP is, therefore, extremely complex because the optimization function potentially comes with an exponential number of variables that cannot be solved in polynomial time in general.**

To this end, we propose a quantum annealing framework for time-bounded and near-optimal solutions for the sTSP, overcoming hardware limits of near-term quantum devices. In particular, we put forth an efficient Hamiltonian (QUBO) to encode the complex decision-making for the sTSP on noisy intermediate-scale quantum (NISQ) annealer. Furthermore, experimental results we obtained on the D-Wave 2000Q quantum hardware demonstrate that the optimal solutions for several instances can be attained.

***Index Terms*—Quantum computing, quantum annealing, optimization, and selective TSP.**

I. Introduction

Quantum computing is gaining momentum in finding applications in a wide range of domains, especially those requiring time-bounded computations. Fujitsu, a company from Japan, announced that they will be able to provide commercial quantum computer for researching purposes from 2023 [1]. D-Wave systems has successfully developed their superconducting quantum processing unit (QPU) based upon quantum annealing (QA) principles to solve complex combinatorial optimization problems and has provided quantum computing as a cloud service. Quantum annealing is an optimization approach that utilizes the quantum adiabatic computation proposed by Kadowaky and and Nishimori [2] to find the global minimum of a given objective function.

In order for a optimization problem to be solved in annealing machines, they must be expressed in the form of quadratic unconstrained binary optimization (QUBO). Several optimization problems, including partitioning problems, graph coloring, tree problems or resource allocation problems on emerging domains such as RAN and network slicing have been formulated as QUBO problems [3]. Traveling salesman problem (TSP) is one the most well-known NP-Hard combinatorial optimization problem in the field of computer science. In the classical formulation of TSP, there are n cities and a cost associated with traveling between each pair of cities. The goal is to find the shortest Hamiltonian path that passes through each city exactly once. TSP is simple to formulate, but difficult to solve, and it has a wide range real-world applications. TSP is used extensively in vehicle routing, printed circuit board drilling, gas turbine engine overhauling, X-ray crystallography, computer wiring, and the order-picking problem in warehouses [4]. Consequently, more and more TSP variants have been proposed to deal with constraint specific applications. These variants are categorized as profit-based, time windows-based, maximal-based, and kinetic-based TSP [5].

Selective travelling salesman problem (sTSP), which is also known as the orienteering problem (OP), is one of the critical problems in the profit-based variants of the TSP. In sTSP, each destination is associated with a prize. Given a budget that must not be exceeded, the salesperson is not necessarily required to visit all the cities, and his mission is to collect as many prizes as possible. There are some similarities between sTSP and the Knapsack problem (KP), in which we are given a set of items with weights and values. The objective is to maximize the total value of these items under a given weight limit. In other word, sTSP can be considered as a combination of KP and TSP, in which the profit collected is maximized with the traveling salesman being subjected to a time or cost constraint.

Motivation. Many existing works have been proposed to solve the TSP using quantum annealing. In [6], [7], the authors have comprehensively described the formulation of classical TSP into a QUBO problem and its solution using D-Wave quantum computer. The work of C. Papalitsa et al.,(2019)[8] proposed a QUBO approach for a variant of TSP, known as traveling salesman problem with time window (TSPTW). However, to the best of our knowledge, none of the existing works address the sTSP with the QUBO framework, or even quantum annealing in general. We summarize our research **contributions** as follows:

- *QUBO Formulation.* Our proposed framework encompasses a QUBO formulation for the sTSP, the first of its kind. The existence of the sTSP formulation is the precondition for its solution on the state-of-the-art D-Wave quantum annealer.
- *Analysis and Assessment.* The proposed QUBO formulation is validated using the 2000Q D-Wave quantum computer. In addition, we extend our discusses on detailed parameters that influence results of quantum annealer.

Corresponding author: Tu N. Nguyen. This research was supported in part by US NSF grants: AMPS-2229073, AMPS-2229075, and CNS-2103405.

Organization. The rest of this paper is organized as follows. In §II, we provide a background of profit-based TSP, quantum annealing, and how a QUBO is solved in D-Wave hardware. We discuss details of classical sTSP, and its QUBO formulation in §III. Evaluation results are presented in §IV. We conclude and highlight contribution and discuss the future works in §V.

II. Preliminaries

In the following sections, we first provide an overview of TSP and its variants in §II-A and provide the background of quantum annealing and how it can be applied in solving an optimization problem in §II-B.

A. Profit-based variants of the Traveling Salesman Problem

In the TSP, the traveling salesman is given a set of cities, each pair of cities is connected with a certain cost of traveling. His goal is to find a minimum Hamiltonian cycle on the graph, that starts and ends at the same node, visits every other nodes exactly once in the intermediate, and have the lowest total cost to travel. The problem can be formulated by using discrete variable, x_i, where $x_i \in \{1, 2, 3, 4, \cdots, n\}$, to represent the order in which each city is visited in the route. Each city must have its unique position in the tour, i.e. one city cannot be first and second in the route. In this paper, we adopt a binary decision variable approach to represent whether a path from one city to others is in the optimal path or not. The integer linear programming (ILP) for this problem is defined as:

$$\textbf{Obj} : \text{minimize} \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} c_{ij} \cdot x_{ij} \tag{1}$$

$$\text{Subject to:} \sum_{\substack{i=1 \\ i \neq j}}^{n} x_{ij} = 1, \sum_{\substack{j=1 \\ j \neq i}}^{n} x_{ij} = 1$$

where c_{ij} is the cost to travel between two cities and x_{ij} is a binary variable takes the value 1 if the salesman travels from city i to city j, 0 otherwise. TSP has a wide range of real-world applications. Consequently, the body of TSP variants grow overtime, to serve different fields of optimization. The variants are surveyed and categorized in [5].

Profit-based TSP is one of the most well-studied area of variants, where each city is associated with a prize to reward the traveling salesman when he visits. The strict requirement of visiting every vertex on the graph is removed. Instead, contention between of reaping the maximum amount of rewards and minimizing the total cost of traveling becomes the objective. There are three main sub-mutations stems from these aforementioned parameter settings:

1) *Selective Traveling Salesman Problem (sTSP):* In this variant, the total cost of travelling is a bounded parameter as the salesman is given a budget that cannot be exceeded. The best route is defined as one that amasses the highest amount of rewards from the trip. This is also known as the orienteering problem (OP) [9], [10], and such problem can have many practical applications in the area of scheduling, routing, where the resources might not be plentiful.
2) *Prize-Collecting TSP (PCTSP)*: Unlike sTSP where the total cost is upper-bounded, PCTSP put a lower-bound on the total reward collected, befitting the alternative name quota TSP [11]. First proposed by Dell'Amico et al. in 1995 [12], this variant apply a lower threshold in which the total reward collected by the salesman must be exceeded as a constraint. With this required quota, it is in the self-interest of the salesman to find the route that cost him the lowest to travel. This variant is widely useful, in contexts such as energy conservation.
3) *Profitable Tour Problem (PTP)*: The travelling salesman is neither being given a travel budget nor a reward quota. Instead the given objective is a direct contention between the total cost and the total reward. The best route is defined as one that maximizes the profit, i.e. the total collected rewards after subtracting the total cost of the trip.

Numerous approaches utilizing classical heuristic algorithms have been presented in the literature for solving profit based variants can be found in [13], [14]. In this work, we examine sTSP variant, reformulate it into the QUBO model, and implement the solution on quantum annealing platform.

B. Quantum Annealing

As stated by the principle of minimum energy, objects in a closed system tends to arrange themselves to find the lowest energy setting of the system, quantum annealing is an optimization strategy that leverage this quantum physical phenomenon. D-Wave Systems is the first company to commercialize a quantum computer based on the principle of quantum annealing. The rest of this section will describe the technical details of a D-Wave quantum processing unit, and the process of running an optimization with it.

The current Chimera topology QPU of D-Wave consists of 2048 superconducting qubits [15], each of them conduct a circulating current, with the Ising Hamiltonian mathematically describes the lowest energy state of such quantum system:

$$H(s) = -\sum_{i} h_i s_i - \sum_{i} \sum_{j<i} J_{i,j} s_i s_j \tag{2}$$

where $s_i \in \{-1, 1\}$ denotes the magnetic dipole moment, while h_i denotes the *bias*, the configurable external electromagnetic field setting, at site i. Entanglement of two sites (i, j) is facilitated by a *coupler* between them whose strength is denoted by $J_{i,j}$. The lowest energy state of a Hamiltonian encodes the optimal solution for the corresponding quadratic minimization problem $\min_{s \in \{-1,+1\}^n} H(s)$.

In general, three procedures are involved in optimizing with a quantum annealer. First, the optimization problem is framed as the problem of finding the minimal energy state of a quantum system. Second, the initialization of the QPU by putting all qubits in a superposition state and embedding the Ising formulation onto the QPU architecture with the corresponding the *bias* and *coupler*. Finally, the solution to the problem, which is the lowest energy configuration of an annealing process, is obtained by measuring the circulating current of each qubit that has collapsed into the classical state.

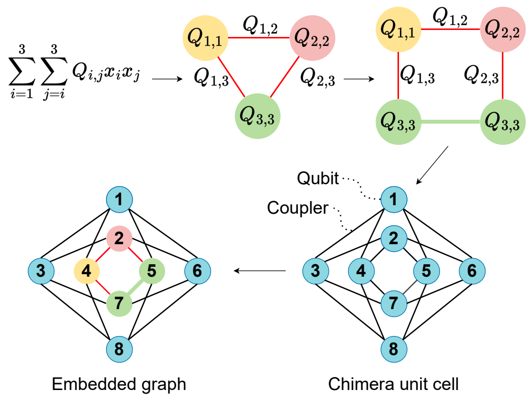

Fig. 1: Logical graph presentation of QUBO and the process of minor-embedding onto the D-Wave Chimera unit cell.

- **QUBO formulation.** The QUBO model, which is a prevalent tool to address discrete optimization problems, can be readily demonstrated its equivalence to the Ising model. The variable setting $\{-1, +1\}$ of the Ising model is not a natural formulation for the problem of concern, QUBO, on the other hand, is generally a much more preferable in model for such optimization problems and for the field of computer science in general. With the minimization objective equation:

$$H(x) = \sum_i Q_{i,i} x_i + \sum_{i<j} Q_{i,j} x_i x_j \tag{3}$$

  where $x_i, x_j \in \{0, 1\}$ is the binary variable, $Q_{i,i}$ is the linear coefficients, and $Q_{i,j}$ is the quadratic coefficients in relation to them. By replacing $x_i = \frac{s_i+1}{2}$, QUBO is easily convertible to Ising and vice versa. QUBO formulation is acceptable by D-wave quantum annealer, in that, the *bias* and *coupler* is derived from the linear and quadratic coefficients, respectively.
- **QUBO graph.** An undirected graph that helps visualize the embedding of the QUBO formulation and how it is mapped onto the QPU. To construct the graph, the set of vertices is plotted from the corresponding variables of the objective function, each vertex is labeled with the linear coefficient of the associated variable, while the quadratic coefficient of two variables labels the edge connecting the vertices.
- **QPU architectures.** Theoretically, if all qubits in a quantum annealer are fully coupled with one another, the formulated QUBO graph can be mapped node-to-qubit perfectly to the QPU. In reality, however, this is not physically achievable with current technology. Instead, different coupler-limited QPU architectures such as the D-Wave's Chimera is designed implemented to make the most of the current technological availability and maximize the utility of entanglement.
- **Minor-embedding.** Due to the previously mentioned limitation of the QPU architecture, the chained-qubits technique is proposed to transform the QPU topology into the unmatched QUBO graph. The chained-qubits are created by strengthening the *coupler* between adjacent qubits, binding their state together. The strongly coupled qubits then behave as one single qubit throughout the annealing process. This gives an option of node-to-chained-qubits mapping, on top of node-to-qubit, thus yields more flexibility to map any graph to a QPU topology. An example of the graph visualization of a QUBO formulation and the process of minor-embedding onto the D-Wave Chimera unit cell is illustrated in Fig. 1.

III. sTSP QUBO Formulation

In this section, we begin with an in-depth discussion of the sTSP, its constraints and their meanings in §III-A. Following that, we convert these objective and constraints (equalities and inequalities) to QUBO form, which is presented in §III-B.

A. The classical formulation of sTSP

Let $G = (V, A)$ be a complete undirected graph with a set of vertex denoted by $V = (\{v_1, ..., v_n\})$ and arc set $A = (\{(v_i, v_j) : v_i, v_j \in V, v_i \neq v_j\})$. Let p_i be a profit associated with each vertex $v_i \in V$. A cost c_{ij} is associated with each arc $(v_i, v_j) \in A$. The integer linear programming formulation of sTSP in [16] introduces the use of binary variable denoted by x_{ij} as follows:

$$x_{ij} = \begin{cases} 1, & \text{if a visit to } v_i \text{ is followed by a visit to } v_j \\ 0, & \text{otherwise.} \end{cases}$$

The comprehensive formal definition of sTSP discussed can be given as follows:

$$\textbf{Obj:} \text{ maximize} \sum_{i=2}^{n-1} \sum_{\substack{j=2 \\ j\neq i}}^{n} p_i \cdot x_{ij} \tag{4}$$

subject to:

$$\sum_{j=2}^{n} x_{1j} = \sum_{i=1}^{n-1} x_{in} = 1 \tag{5}$$

$$\sum_{i=1}^{n-1} x_{ik} = \sum_{j=2}^{n} x_{kj} \leq 1; \forall k = 2, ..., n-1 \tag{6}$$

$$\sum_{i=1}^{n-1} \sum_{j=2}^{n} c_{ij} \cdot x_{ij} \leq C_{max} \tag{7}$$

$$\sum_{i\in S} \sum_{\substack{j\in S \\ j\neq i}} x_{ij} \leq |S| - 1; \forall S \subset V, 2 \leq |S| \leq n-1 \tag{8}$$

In most of the cases, the tour of the salesman begins and ends at fixed vertex v_1 and v_n ($v_1 \not\equiv v_n$). However, in many applications, v_1 and v_n coincide, which form a Hamiltonian cycle such as the work from Masani et.al[17]. The difference between these two versions is negligible, and they are interchangeable by adding or removing a trivial arc between these two vertices. Constraint (5) employs non-Hamiltonian tour version of sTSP by making sure that the tour starts at v_1 and ends at v_n. Constraint (6) ensures that the salesperson visits and leaves each vertex at most once, which is consistent with the sTSP given that the tour does not have to traverse through all of the cities (or vertices in the graph). Constraint (7) limits the overall cost of the tour is less than or equal to a specified C_{max} constant. Finally, we prevent subtour in the route by adding constraint (8). This constraint is essential to exclude closed loop from the route. The intuition behind this constraint is that a cycle is formed in a graph when the number of vertices and arcs are equal. By ensuring that the number of arcs in each subset S is less than the number of vertices in S, the subtour can be prevented.

Motivation example. Given a set of four vertex $V = \{v_1, v_2, v_3, v_4\}$, choosing a subset $S = \{v_1, v_2 : v_1, v_2 \in V, S \subset V\}$, constraint (8) become:

$$\sum_{i \in \{1,2\}} \sum_{\substack{j \in \{1,2\} \\ j \neq i}} x_{ij} \leq |S| - 1 \tag{9}$$

which simplifies to:

$$x_{12} + x_{21} \leq 1$$

Above example gives an idea how constraint (8) can eliminate subtour in the optimal solution. Either of x_{12} or x_{21} takes the value 1, the other must equal to 0, which helps to eliminate subtour $v_1 \to v_2 \to v_1$ in S. Unfortunately, the number of constraints to eliminate subtours is exponential, requiring an efficient approach. In the following section, we consequently propose a QUBO formulation of classical sTSP that can be understood and solved by a quantum annealing computer.

B. QUBO formulation for the sTSP

In the literature, there are several problems that naturally fall into QUBO form, e.g., the number partitioning problem, the max-cut problem [18]. This is because solving these problems involves the use of binary decision variables and relations between these variables can be represented using linear and quadratic terms. To formulate a combinatorial problem to QUBO, we need to convert equality and inequality constraints into penalties (also called Hamiltonian) of the objective function. These penalties should be equal to zero for all feasible solutions to the problem and equal to some positive penalty if the solution violates constraints of the problem. In the rest of this section, we will use this approach for the construction of Hamiltonians for sTSP.

The first Hamiltonian is converted from the original objective function (4):

$$H = \sum_{i=2}^{n-1} \sum_{\substack{j=2 \\ j \neq i}}^{n} p_i \cdot x_{ij} \tag{10}$$

In QUBO, equality $Ax = b$ is usually treated by converting to square expression $(Ax - b)^2$ as mentioned [19]. Thus, equality constraint like (5) and can be easily represented as:

$$C_1 = P_1[(1 - \sum_{i=2}^{n} x_{1i})^2 + (1 - \sum_{j=1}^{n-1} x_{jn})^2] \tag{11}$$

Constraint (6) is separated into two Hamiltonians C_2, C_3:

$$C_2 = P_2 \sum_{k=2}^{n-1} \sum_{\substack{i=1 \\ i \neq k}}^{n-1} x_{ik} \sum_{\substack{j=1 \\ j \neq i, j \neq k}}^{n-1} x_{jk} \tag{12}$$

$$C_3 = P_3 \sum_{k=2}^{n-1} \sum_{\substack{i=2 \\ i \neq k}}^{n} x_{ki} \sum_{\substack{j=2 \\ j \neq i, j \neq k}}^{n} x_{kj} \tag{13}$$

Consider a vertex v_k, if there is only one visit to this vertex, the quadratic terms corresponding to v_k will be equal to 0. If the constraint hold for all vertices in $V \backslash \{v_1, v_n\}$, Hamiltonian C_2 will be equal to 0. Thus, it has no effect on the objective function. Otherwise, it will add a penalty to the objective function. This penalty can be calculated as:

$$P_2 \frac{n_L!}{(n_L - n_V)!} \tag{14}$$

where $n_L = \sum_{k=2}^{n-1} \sum_{\substack{i=1 \\ i \neq k}}^{n-1} x_{ik}$ and n_v is the number of vertices that violate the constraint. The same principle applies to Hamiltonian C_3, it will add a penalty equal to $P_3 \frac{n_L!}{(n_L - n_V)!}$ to the objective function where $n_L = \sum_{k=2}^{n-1} \sum_{\substack{i=1 \\ i \neq k}}^{n-1} x_{ki}$ and n_v is the number of vertices that violate the constraint.

Constraint (7) and (8) are in the form of $Ax \leq b$. In order to transform these constraints into the objective function as penalty terms, we introduce a set of slack binary variables as mentioned in [20]. The number of slack variables for constraints (7) and (8) are limited to $\lceil 1 + \log_2 C_{max} \rceil$ and $\lceil 1 + \log_2(|S| - 1) \rceil$ respectively, where $\lceil . \rceil$ is the ceiling function. Accordingly, the Hamiltonian terms for these constraints is expressed as:

$$C_4 = P_4 (\sum_{i=1}^{n-1} \sum_{\substack{j=2 \\ j \neq i}}^{n} c_{ij} \cdot x_{ij} + \sum_{k=0}^{\lceil 1 + \log_2 C_{max} \rceil} 2^k \cdot \lambda_k - C_{max})^2 \tag{15}$$

$$C_5 = P_5 (\sum_{i \in S} \sum_{\substack{j \in S \\ j \neq i}} x_{ij} + \sum_{l=0}^{\lceil 1 + \log_2(|S| - 1) \rceil} 2^l \cdot \lambda_l - |S| + 1)^2$$
$$\forall S \subset V, 2 \leq |S| \leq n - 1 \tag{16}$$

Finally, to solve sTSP using QUBO framework, the overall Hamiltonian H_F is given below:

$$H_F = H - \sum_{k=1}^{5} C_k \tag{17}$$

It is notable that in each Hamiltonian terms $C_i, i = 1, \cdots, 5$ having a numerical value P_i associated with each equation, which are referred to as a scalar penalty. These number are not required, and can be arbitrarily chosen based on different type of constraints. If the constraint is "hard", which mean it has to be absolutely satisfied, then the number scalar number should be large enough to preclude a violation. On the other hand, some "soft" constraints accept slight violation. In that case, a moderate P value is sufficient.

C. Analysis: Problem Complexity

This section is dedicated to examining the intricacies of problem's complexity. In our QUBO formulation, the total number of decision variables is equal to the number of variables x_{ij} that are used to describe the relation between v_i and v_j ($v_i, v_j \in V$) and the number of slack variables λ_k, λ_l in (15) and (16), respectively.

Lemma 1. *With the input size n, the total number of decision variables x_{ij}, denoted by $N_F = n^2 - n$.*

Proof.

$$\begin{aligned} N_F =& \frac{n!}{(n-2)!} \\ =& \frac{n(n-1)(n-2)!}{(n-2)!} \\ =& n^2 - n \end{aligned} \tag{18}$$

The total number of slack variables denoted by N_S is equal to the number of slack variables λ_k and λ_l. Given a constant C_{max} that must not be exceeded, the number of λ_k is equal to $\lceil 1+\log_2 C_{max} \rceil$. Let us consider subtour elimination constraint (8), the number of constraints is equal to $\sum_{|S|=2}^{n-1} \binom{n-1}{|S|}$, where $\binom{n-1}{|S|}$ is the number of combination when choosing $|S|$ elements from a set of $n-1$ elements. Given a subset S, the number of slack variables we need to convert inequality constraint (8) to equality constraint (16) is $\lceil 1+\log_2(|S|-1) \rceil$. Therefore, the total number of slack variables for subtour elimination constraint is $\sum_{|S|=2}^{n-1} \binom{n-1}{|S|} \lceil 1+\log_2(|S|-1) \rceil$. N_S can be calculated using the following formula:

$$N_S = \lceil 1 + \log_2 C_{max} \rceil + \sum_{|S|=2}^{n-1} \binom{n-1}{|S|} \lceil 1 + \log_2(|S|-1) \rceil \tag{19}$$

It takes $N_F + N_S$ decision variables to formulate sTSP using our QUBO formulation. This thus completes the proof. □

IV. Experiment and Evaluation

In the previous section, we proposed a QUBO formulation designated to determine the feasible solution of sTSP for execution on a quantum annealing computer. The QUBO formulation can be solved using either QPU solver or hybrid Quantum-Classical solver. The process explains how QPU solves a QUBO problem is discussed in section II-B. Solving a problem with D-Wave 2000Q QPU is only possible for problems with a limited number of instances. At a larger scale, we have the hybrid solver developed by D-Wave system, which utilizes both classical and quantum computing and can accept inputs that are considerably larger than QPU solver. In this section, we presents experimental results derived from the state-of-art D-Wave 2000Q QPU, along with their corresponding analytical insights.

Dataset. We apply our QUBO approach to a small number of cities denoted by $n = \{5, 6, 7\}$. In each test case, the set of cities is denoted by $V = \{v_1, \cdots, v_n\}$, with the starting and ending cities fixed to v_1 and v_n, respectively. The profit associated with each city and the cost to travel between cities v_i and v_j are randomly selected among the values $\{1, 2, 3, 4\}$. Constant C_{max} is set at 20% of the total cost to travel between cities. We arbitrarily choose $\{5\%, 10\%, 15\%, 20\%, 25\%\}$ of the total profit as the scalar penalty P_i to all Hamiltonians mentioned in (17). Once the Q matrix is obtained using our QUBO formulation, we proceed to embed it onto the Chimera graph of the D-Wave 2000Q QPU and perform evaluations.

n	var.	qubit	P_i				
			5%	**10%**	**15%**	**20%**	**25%**
5	16	84	8.67	17.34	26.01	34.69	43.36
6	25	214	17.25	34.50	51.76	69.01	86.27
7	35	418	20.24	40.48	60.73	80.97	101.21

TABLE I: Number of logical variables, physical qubits corresponding to number of cities $n = \{5, 6, 7\}$ and chain strength of different P_i.

Physical qubits. After embedding, the number of physical qubits are measured. Table I illustrates the quantity of physical qubits and logical variables that correspond to each city count. It can be observed that the number of physical qubits grows significantly as the number of instances increases. The most advantageous quantum annealer D-Wave 2000Q QPU supports 2048 qubits and 6016 couplers, our analysis leads us to predict that the QPU is unable to accommodate the problem when there are more than 15 instances.

Time. Each Ising Model or QUBO problem sent to the D-Wave quantum machine is known as a Quantum Machine Instruction (QMI). The overall QPU time that is allocated to a QMI is decomposed into three main components as follows:

- Access time: Defined as the time QPU execute a specific QMI, while being unavailable to other QMIs.
- Sampling time: This is the actual annealing time of the QPU and can be calculated as follows, $R \cdot (T_a + T_r + T_d)$, where R is the number of cycles, T_a, T_r,T_d is the annealing time, read-out time, delay time of one sample respectively.
- Post Process Time: During the *access time* of a QMI, QPU return sets of sample in batches. *Total post process time* is the total time to post process these sample batches.
- Range of *annealing_time* is $[0.01, 2000]$ μs, Fig. 2 breaks down the overall QPU time with different number of cities and the *annealing_time* parameter in $[5, 15, 25, 35, 45]$ μs. In general, the access time and sampling time grows when we increase the *annealing_time*, whereas the total post process time contributes little to the overall QPU time and remains mostly the same regardless of *annealing_time*.

Chain strength. One important parameter of QPU solver is *chain_strength*. This parameter specifies the relatives coupling strength of chains embedded onto the QPU hardware graph. There is a feature called auto-scaling that divides all terms by the largest QUBO weight if it larger than *chain_strength*. Otherwise, it will divide all the QUBO terms by the value of

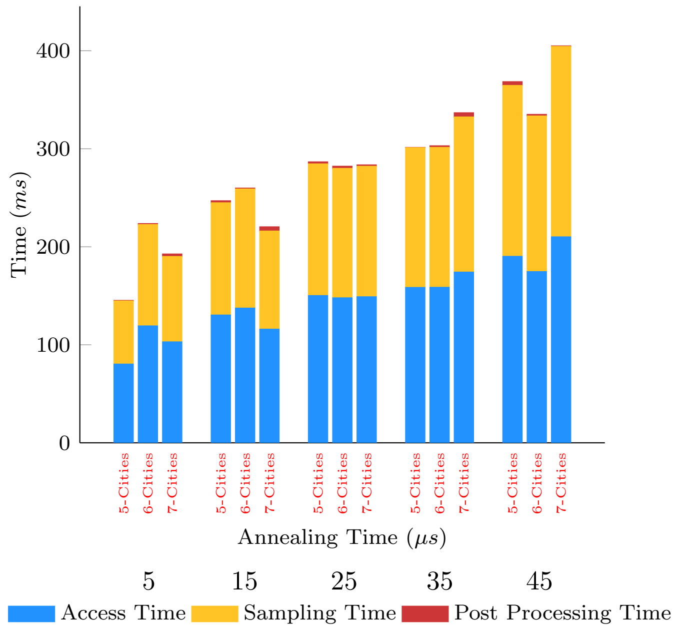

Fig. 2: Performance metrics in D-Wave 2000Q QPU Sampler with *num_reads*=1000 and *annealing_time* ranging from 5 to 45 μs when the number of cities $n = \{5, 6, 7\}$.

chain_strength parameter. If this parameter is set too large, all QUBO terms is shrink near to 0. Consequently, it no longer the original problem because variables become independent. If *chain_strength* is too small, the relationship between qubits dose not work as intended. Thus, the value of this parameter must be carefully chosen. By default, it is calculated using *uniform_torque_compensation* function provided by D-Wave System. Table I reports the chain strength obtained from *uniform_torque_compensation* with different number of cities. With $n = 5, 6$, there is no chain break using all P_i. However, when $n = 7$, all chains break with $P_i = 5\%$ and majority of chains break with $P_i = 10$. Thus, with a large number of instances, we suggest that P_i should be set greater than 20% of the total profit.

Solution. With n = 5, the optimal solution is obtained using $P_i = 25\%$ regardless *annealing_time*. When n = 6, we discover optimal solution with $P = 20\%$ and *annealing_time* of $15\mu s$. However, when n = 7, the optimal solution cannot be found using $P_i = \{5\%, 10\%, 15\%, 20\%, 25\%\}$. In this case, a larger scalar constant is required to preserve the relationship between variables. A good value of P_i can be found using specific domain knowledge and trial-and-error.

V. Conclusion and Future Work

Going beyond classical computing, this paper encloses a systematic study of quantum computing for selective traveling salesman problem (sTSP). Specifically, we proposed a quadratic unconstrained binary optimization (QUBO) formulation to encode the complex decision-making for the sTSP on noisy intermediate-scale quantum (NISQ) annealer. To derive the QUBO formulation, all classical constraints of sTSP were formulated into Hamiltonian terms. We have further sampled the proposed algorithm using D-Wave 2000Q QPU solver. Experimental results obtained on the D-Wave 2000Q quantum hardware demonstrate that the optimal solutions for several instances can be attained.

This work will the lay foundation for further research on "quantum annealing" for TSP and its variants. We look forward to extending future research topics that can extend experiments on a much larger dataset and compare the performance of a purely quantum annealing approach for sTSP with its conventional counterparts, such as the one using the tabu search. Our objective is to determine what extent pure quantum annealing approach can outperform the classical methods.

References

[1] M. Humphries. (2022) Fujitsu to make quantum computers available for research in 2023. [Online]. Available: https://www.pcmag.com/news/fujitsu-to-start-selling-quantum-computers-in-2023

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

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

[4] O. Cheikhrouhou and I. Khoufi, "A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy," *Computer Science Review*, vol. 40, p. 100369, 2021.

[5] K. Ilavarasi and K. S. Joseph, "Variants of travelling salesman problem: A survey," in *International Conference on Information Communication and Embedded Systems (ICICES2014)*. IEEE, 2014, pp. 1–7.

[6] S. Jain, "Solving the traveling salesman problem on the d-wave quantum computer," *Frontiers in Physics*, p. 646, 2021.

[7] R. H. Warren, "Solving the traveling salesman problem on a quantum annealer," *SN Applied Sciences*, vol. 2, no. 1, pp. 1–5, 2020.

[8] C. Papalitsas, T. Andronikos, K. Giannakis, G. Theocharopoulou, and S. Fanarioti, "A qubo model for the traveling salesman problem with time windows," *Algorithms*, vol. 12, no. 11, p. 224, 2019.

[9] G. Laporte and S. Martello, "The selective travelling salesman problem," *Discrete applied mathematics*, vol. 26, no. 2-3, pp. 193–207, 1990.

[10] P. Vansteenwegen and A. Gunawan, "Orienteering problems," *EURO Advanced Tutorials on Operational Research*, 2019.

[11] T. S. Marques, S. F. Cezario, E. F. G. Goldbarg, M. C. Goldbarg, and S. M. D. M. Maia, "Quota traveling salesman with passengers and collection time," in *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2019, pp. 299–304.

[12] M. Dell'Amico, F. Maffioli, and P. Värbrand, "On prize-collecting tours and the asymmetric travelling salesman problem," *International Transactions in Operational Research*, vol. 2, no. 3, pp. 297–308, 1995.

[13] K. V. Dasari, V. Pandiri, and A. Singh, "Multi-start heuristics for the profitable tour problem," *Swarm and Evolutionary Computation*, vol. 64, p. 100897, 2021.

[14] X. Chou, L. M. Gambardella, and R. Montemanni, "A tabu search algorithm for the probabilistic orienteering problem," *Computers & Operations Research*, vol. 126, p. 105107, 2021.

[15] M. Jünger, E. Lobe, P. Mutzel, G. Reinelt, F. Rendl, G. Rinaldi, and T. Stollenwerk, "Performance of a quantum annealer for ising ground state computations on chimera graphs," *arXiv preprint arXiv:1904.11965*, 2019.

[16] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: A survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[17] N. Labadie, R. Mansini, J. Melechovskỳ, and R. W. Calvo, "The team orienteering problem with time windows: An lp-based granular variable neighborhood search," *European Journal of Operational Research*, vol. 220, no. 1, pp. 15–27, 2012.

[18] H. N. Djidjev, G. Chapuis, G. Hahn, and G. Rizk, "Efficient combinatorial optimization using quantum annealing," *arXiv preprint arXiv:1801.08653*, 2018.

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

[20] A. Verma and M. Lewis, "Penalty and partitioning techniques to improve performance of qubo solvers," *Discrete Optimization*, p. 100594, 2020.