Case Study

Solving the traveling salesman problem on a quantum annealer

Richard H. Warren[1]

Received: 23 August 2019 / Accepted: 29 November 2019 / Published online: 13 December 2019

Abstract

The paper contains an analysis of four software programs that solve the symmetric traveling salesman problem on a quantum annealer. Three are designed to find approximate solutions. One is designed to find an optimal tour. These programs demonstrate that an application can run across both classical and quantum computing platforms and take advantage of what each can do best. We add value by using a uniform structure for our analysis so that a consistent standard is used to evaluate the software programs. Also we add value by designing a software experiment to test the ability of the D-Wave quantum computer to optimally solve the traveling salesman problem. Our design combines the best attributes of the programs that are reviewed in this paper. Our design assumes that the variables of the traveling salesman problem can be embedded in the qubits, which excludes the problems in the TSP Library until the D-Wave Pegasus computer is available. We note applications of the asymmetric traveling salesman problem that are in the literature and include these problems in the recommendation for an experiment.

Keywords Traveling salesman problem · Optimal tour · Quantum computer · Quantum annealer

1 Introduction

The traveling salesman problem (TSP) is a well-known optimization problem [1, 2] due to its computational complexity and real-world applications, such as routing school buses and scheduling delivery vehicles. Asymmetric applications are described in [3, 4].

Given *n* cities and the distance between city *i* and city *j*, the symmetric TSP asks for a shortest route through the *n* cities visiting each city once and returning to the originating city. When the distance from city *i* to city *j* is different than the distance from city *j* to city *i*, it is called an asymmetric TSP (ATSP). The adaption of the TSP to a quantum annealer in [5, 6] is for the ATSP. This is natural and prevents subloops through proper subsets of the *n* cities. Therefore, we advocate ATSPs in this paper. A symmetric TSP can be processed by an ATSP application. A tour for the salesman is one loop through all the cities, i.e., a cyclic permutation of the cities. The length of a tour is the distance traveled through the loop. An optimal tour is a shortest tour.

Quantum annealing is a new method of computing that has the potential to solve optimization problems faster than classical means. It has been used to solve some problems that are beyond current classical techniques [7]. The theory for quantum annealing implies that the qubits will achieve an optimal state of low energy when super cooled. This is represented by the following expression where an initial Hamiltonian H_o evolves to its low energy state in a final Hamiltonian H_p according to

$$H(t) = \left(1 - s\left(\frac{t}{T}\right)\right)H_o + s\left(\frac{t}{T}\right)H_p \text{ for } 0 \le t \le T \tag{1}$$

as $s(\tau)$ increases from $s(0) = 0$ to $s(1) = 1$ and if H_o and H_p do not commute. In theory, T is the time imposed by the Schrödinger equation for the initial Hamiltonian to evolve to its low energy state. On a D-Wave computer, time T is in microseconds. The Hamiltonian H_o is established by

---

✉ Richard H. Warren, rhw3@psu.edu | [1]Lockheed Martin Corporation – Retired, King of Prussia, PA 19406, USA.

SN Applied Sciences (2020) 2:75 | https://doi.org/10.1007/s42452-019-1829-x

D-Wave for all problems. The Hamiltonian H_p represents the combinatorial problem to be solved and is an input. Besides H_p, other inputs include the number of samples of the problem, the time T within a given range, and scaling factors. Essentially, a result $H(t)$ is a sample from a Boltzmann distribution. A physical implementation of (1) does not strictly meet the conditions for quantum annealing. In addition, values loaded by the user may differ slightly from the machine interpretation of the numbers. These difficulties are partially overcome by taking multiple samples and choosing a valid solution that has minimum energy.

Section 2 of this report cites related work. Sections 3–6 examine four pieces of software for the TSP that execute on a 2000 qubit, quantum computer manufactured by D-Wave Systems [8] that is termed as a quantum annealer. The following properties of the software are described and examined:

(A) Is the software designed to produce an optimal or approximate solution?
(B) What percent of 100 different TSPs with a fixed number of cities and random distances are solved optimally? See [12].
(C) Is the software designed for symmetric or asymmetric TSPs?
(D) What is the maximum number of cities that the software can handle?
(E) What number of cities less than the maximum can be processed?
(F) What limitations are designed into the software? What limits the software?
(G) How can the software be improved to raise the percent of solutions that are optimal?
(H) Designation of software application.
(I) Statistics about data from many executions of the software.

Section 7 contains conclusions and Sect. 8 contains a challenge to build software for a quantum annealer that is designed to find an optimal tour for ATSPs with random distances. A top-level design of the software is included.

2 Related work

Results about ATSPs are in references [9–12]. Applications of the ATSP include many real-world examples reported in Section 5 of [3], an optimal chemical composition for shipbuilding steel [4], and vehicle routing instances in the ftv collection of the TSPLIB [13]. A literature search revealed no published papers about a new algorithm to find an optimal tour for the ATSP on a quantum computer since Ref. [5] was published in 2013.

Article [14] has difficulties due to conflicting definitions of a_{ik} on pages 1 and 2 that affect Eqs. (4) and (6)–(8). Paper [15] has excellent theoretic results for special TSPs with degree restrictions. Document [16] contains work about the TSP on a quantum annealer, but needs verification by solving several TSPs with at least six cities on a quantum annealer.

3 D-Wave traveling salesman seven cities [17]

This software package in [17] attempts to solve a 7-city, symmetric TSP based on the mileage between seven US cities. The software has a design flaw that allows subloops to be part of some solutions. For example, if the cities are A, B, …, G, then A → B → C → A and D → E → F → G → D decompose a 7-city problem into two subloops. Due to these invalid solutions, no attention is given to this software.

4 D-wave traveling salesman 48 cities [17]

We analyze the software package in [17] for solving a 48-city TSP. This TSP finds a loop through the capitals of the contiguous 48 US states. The D-Wave hardware limits TSP processing to 9 cities. The design uses the software package QBSolv to decompose, solve the pieces, and combine the solutions for a global solution. Please refer to Sect. 1 for properties A–I that follow.

(A) Optimal? The software finds an approximate solution because the 48 cities are decomposed into a family of disjoint sets that can be solved by classical or classical/quantum techniques, then recombined into one loop.
(B) Percent? Distances are not random. Distances are between 48 US state capitals.
(C) ATSP? Symmetric since code Section 1.2 of [17] has D[citya][cityb] = D[cityb][citya] where D is distance.
(D) The maximum number of cities for this software is 48.
(E) Less than maximum number of cities? When the number of cities is less than 48, an index error occurs.
(F) Limitations? The software is designed for 48 US state capitals and cannot handle other input.
(G) Suggested improvements? The software could be redesigned to handle x cities and benchmark against the x-city TSP in the TSPLIB [13] for $x = 14, 16, 22$. These three TSPs are listed in Table 1 in [18]. Suggest the same for the 29-city and 38-city TSPs in Table 1 in [18]. They originate in [19].

(H) The software for this application is available. It uses Python 3.

(I) Statistics? Data in Table 1 is for the lowest energy sample from two executions that we made, each with 100 repeats, on the quantum processing unit designated solver DW_2000Q_2_1. All lowest energy solutions were valid tours of the 48 cities. The software does not indicate the number of samples that had lowest energy or the mileage for a shortest tour.

Because the six mileages in Table 1 are distinct, we made a third execution of the software. It resulted in three additional mileages for a total of nine distinct mileages for a shortest tour from 100 samples. We conclude that nine different tours produced these nine mileages and that only one of these tours can be optimal, or quite possibly none of them is optimal. According to the trend of average deviation in Table 1 in [18], this major difficulty is expected for a 48-city TSP on a current D-Wave processor.

5 TSP quantum solver [20]

Reference [20] is a demo that allows 4 to 9 cities to be selected from 28 European capitals and finds a shortest tour through the selected cities. The result is a map of a shortest-found tour through the selected cities. There is no data about the energy for this tour or the length of an optimal tour. Please refer to Sect. 1 for properties A–I that follow.

(A) Optimal? The blog indicates the software is designed to find optimal tours.

(B) Percent? Distances are not random.

(C) ATSP? Symmetric because inputs are coordinates of cities.

Table 1 Data from two executions of the 48 city TSP

	Classical	Quantum with embedding composite	Quantum with fixed embedding composite
4/26/2019			
Energy	−800,300	−799,945[b]	−800,304[a]
Mileage	15,700	16,055[b]	15,696[a]
Wall time (s)	58	67[b]	55[a]
4/30/2019			
Energy	−799,424[a]	−798,008	−796,583[b]
Mileage	16,576[a]	17,992	19,417[b]
Wall time (s)	50.8236	57.7255[b]	33.2532[a]

[a] Best of three results in the row

[b] Worst of three results in the row

(D) The maximum number of cities is 9. This limitation is not due to the software. The D-Wave 2000 solver cannot process or embed some TSPs with ten or more cities.

(E) Less than maximum number of cities? 4 to 8 cities can be processed.

(F) Limitations? The demo is designed for up to 9 of 28 European capitals and cannot handle other input.

(G) Suggested improvements? The software could be enhanced to collect data to display the length of the quantum tour and compare it to an optimal tour and its length.

(H) The software for this project is in GitHub.

(I) Statistics? This demo does not show data.

6 TSPs solved and reported in [18]

Report [18] describes the quantum results of solving five TSPs that are too large to be processed on a current D-Wave machine, and must be solved in pieces that are combined to solve the TSP. Software QBSolv was called to break a TSP into quantum solvable pieces and combine the solutions of the pieces to solve the TSP. Usually this is needed to solve a TSP with more than 9 cities on a current D-Wave 2000 processor. Please refer to Sect. 1 for the following properties A–I.

(A) Optimal? An optimal tour was found for two TSPs. One has 14 cities and the other 16 cities. The study used 25,000 samples (100 executions, each with 250 repeats) for each TSP to discover an optimal tour. Three TSPs with 22, 29 and 38 cities were solved approximately on a D-Wave machine. A study of each of these TSPs also used 25,000 samples.

Table 1 of [18] indicates outstanding solution quality for a 14 city, symmetric TSP. It shows that 100% of the 100 executions found an optimal tour, even though QBSolv decomposition software was used. That means the quantum effects can consistently distinguish between the length of an optimal tour and the next smallest length of a tour for a specific 14 city TSP. This leads to asking, what size is the gap between these two tour lengths, and how were the settings of the parameters for chain strength and penalty functions determined? Also, to what extent was post processing used and anneal timing adjusted? The corresponding author for [18] has not responded to a request for information.

(B) Percent? Distances are not random.

(C) ATSP? Apparently, symmetric because the formulation is for undirected edges of a graph according to

Table 2 Evaluation of the useable software programs

Section of paper	References	Number of TSPs and number of cities	Strength	Weakness
4	[17]	1 TPS with 48 cities	Code for the modules	Insufficient comments in code
5	[20]	$\frac{28!}{4!24!}$ TPSs with 4 cities, …, $\frac{28!}{9!19!}$ with 9 cities	Software designed for optimal tour	Does not list an optimal tour and its length to compare to quantum results Website often down
6	[18]	5 TPSs; 14, 16, 22, 29, 38 cities	Statistics and analysis of TSP results. TSPs in the TSP library	No code for the reader to execute to verify results

paragraph 3 of Section 1 in [18]. Also, this appears at the end of first paragraph of Section 4.3 in [18].

(D) The maximum number of cities that can be processed is not clear. Probably, a maximum number is limited by the decomposition software QBSolv that D-Wave provides. It is noted in Figure 5 of [18] that the quality of the best solution decreases as the number of cities increases.

(E) An answer is not clear for the number of cities that can be handled. Paper [18] reports that TPSs with 14, 16, 22, 29 and 38 cities were processed on a quantum annealer and optimal solutions were found for the smaller two. Apparently, the best case maybe if $14 \leq x \leq 38$, then symmetric TSPs with x-cities can be processed by the software described in [18].

(F) Limitations? The software processed five TSPs with differing number of cities. It is not clear if the software can handle other number of cities.

(G) Suggested improvements? Not applicable because the software is not made available.

(H) The listings in the Appendix in [18] indicate that many python modules are included in the software.

(I) Statistics? Paper [18] has excellent statistics about lengths of tours from solving five symmetric TSPs on a quantum annealer. See Table 1 and Figure 5 in [18].

Considering the 38-city TSP, according to Table 1 in [18], the average of the best solutions for 100 executions deviates from the optimal solution by 25.91%. It is more important how much the best solution varies from the optimal solution. The length of the best solution is reported to be 7396. Compared to the optimal length of 6656, this is a variance of 11.12%. We note that there appears to be a wide range of best solutions (lowest energy) for 100 executions of the 38-city TSP.

7 Conclusions about the software programs

We conclude in Table 2 with a list of strengths and weaknesses of the software programs that we reviewed. This is followed by a discussion of our skepticism of the optimal results about the 14-city TSP Burma in [18].

Table 1 of [18] indicates for the 14-city TSP Burma that the average deviation of 100 quantum runs from the best known solution is 0.00%. This seems to mean that 100% of the 100 quantum executions found the best known solution, even though the D-Wave processor is highly analogue and the partitioning/solving/recombining software QBSolv was used. Apparently, an independent verification of this claim has not been made. A verification has been stymied by lack of the code that was used. Additional details are in Sect. 6 Property A.

The software programs described in our Table 2 are based on actual cities and physical distances between them, and all are symmetric. The TSP has many applications that are not symmetric and are more general than cities and distances. Furthermore, we do not think the evidence in [18] about a 14-city TPS and a 16-city TPS is adequate for the conclusion in [18] Section 5.2.1 that "the TSP can be solved comparatively well for smaller sized problem instances on the quantum annealer." Therefore, in Sect. 8 we recommend a new demonstration that is more universal than those described in Table 2 and incorporates their best attributes.

8 Outlook

The D-Wave 2000 qubit computer is highly analogue and nondeterministic. The difficulties seem to include noise that hinders the hardware from distinguishing small numeric differences, sensitive parameters that need adjusting for the distances of TSPs, and imprecise qubit biases and coupling strengths that represent the TSP formulation. These obstructions are hardware snags that impede the best software for solving the TSP.

Therefore, a new demonstration that we are recommending will help to determine the extent to which the D-Wave hardware can find optimal tours for a variety of TSPs.

We recommend that software be designed, implemented and made available to solve ATSPs on the next generation D-Wave processor [21, 22]. The software shall be designed to find an optimal tour for ATSPs with random distances. The software shall have the following capabilities.

Design features: N is a fixed, positive integer. It is an upper bound for the number of cities.
n is an integer such that $4 \le n \le N$. It is the number of cities to be processed.
B is a fixed, positive integer. It is an upper bound for the distances.
N and B can be used for sizing the software or replaced with a library.

User inputs: Integer n.
An $n \times n$ matrix whose ij entry is the integer distance from city i to city j.
Parameters for the D-Wave processor.

Outputs from a classical processor: When feasible, all optimal tours [12] for the n-city ATSP and length of an optimal tour for the n-city ATSP.

Outputs from a D-Wave processor: Tours and their length for the lowest energy solution, the number of samples that have the lowest energy, and the lowest energy.

Statistics: (shortest tour length for the lowest energy solution) minus (length of an optimal tour). Percent of variance of (the length of a lowest energy solution) from (the length of an optimal tour). Percent of 100 different TSPs, all with the same number of cities and random distances, that are solved optimally [12].

Acknowledgements Michał Stęchły, Sebastian Feld, a reviewer and the Section Editor contributed suggestions that improved this paper. The Lord Jesus provided ideas, motivation and financial resources.

Compliance with ethical standards

Conflict of interest The author declares that he has no conflict of interest.

References

1. Lawler EL, Lenstra JK, Rinnooy Kan AHG, Shmoys DB (eds) (1985) The traveling salesman problem. A guided tour of combinatorial optimization. Wiley, Hoboken
2. Reinelt G (1994) The traveling salesman. Computational solutions for TSP applications, Lecture notes in computer science, vol 840. Springer, Berlin
3. Fischetti M et al (2007) Exact methods for the asymmetric traveling salesman problem, Chapter 4. In: Gutin G, Punnen A (eds) The traveling salesman problem and its variations. Springer, Berlin
4. Zakharov AO, Kovalenko YV (2018) Reduction of the Pareto set in bicriteria asymmetric traveling salesman problem. arXiv:1805.10606
5. Warren RH (2013) Adapting the traveling salesman problem to an adiabatic quantum computer. Quantum Inf Process 12:1781–1785. https://doi.org/10.1007/s11128-012-0490-8
6. Lucas A (2014) Ising formulations of many NP problems. Front Phys 2:5
7. Ranjbar M, Macready WG, Clark L, Gaitan F (2016) Generalized Ramsey numbers through adiabatic quantum optimization. Quantum Inf Process 15:3519–3542
8. D-Wave Systems Inc. https://www.dwavesys.com/ > TECHNOLOGY > D-WAVE 2000Q. Accessed 8 Dec 2019
9. Murata Y et al (2017) Solving asymmetric traveling salesman problems by coherent Ising machine. Inst Electron Inf Commun Eng IEICE Tech Rep 116:7–12 **(in Japanese)**
10. Roberti R, Toth P (2012) Models and algorithms for the asymmetric traveling salesman problem: an experimental comparison. EURO J Transp Logist 1:113–133
11. Öncan T, Altinel IK, Laporte G (2009) A comparative analysis of several asymmetric traveling salesman problem formulations. Comput Oper Res 36:637–654
12. Warren RH (2017) Small traveling salesman problems. J Adv Appl Math 2:101–107. https://doi.org/10.22606/jaam.2017.22003
13. TSPLIB. http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/. Accessed 8 Dec 2019
14. Heim B, Brown EW, Wecker D, Troyer M (2017) Designing adiabatic quantum optimization: a case study for the traveling salesman problem. arXiv:1702.06248v1
15. Moylett DJ et al (2017) Quantum speedup of the traveling-salesman problem for bounded-degree graphs. Phys Rev A 95:032323. https://doi.org/10.1103/PhysRevA.95.032323
16. Kieu TD (2019) The travelling salesman problem and adiabatic quantum computation: an algorithm. Process, Quantum Inf. https://doi.org/10.1007/s11128-019-2206-9
17. D-Wave TSPs 7 Cities and 48 Cities. https://cloud.dwavesys.com/leap/login/ log in or register as nondeveloper > Resources > Learning and Docs > scroll down and click on jupyter – LEARN – Interactive Examples > scroll down to Traveling Salesman and click. Accessed 8 Dec 2019
18. Feld S et al (2019) A hybrid solution method for the capacitated vehicle routing problem using a quantum annealer. Front ICT. https://doi.org/10.3389/fict.2019.00013. arXiv:1811.07403v2
19. Cook WJ (2017) National traveling salesman problems. http://www.math.uwaterloo.ca/tsp/world/countries.html. Accessed 8 Dec 2019
20. Stęchły M Traveling salesman problem quantum solver, http://www.solvequantum.com/ > scroll down and click on "Try it out!". https://github.com/BOHRTECHNOLOGY/quantum_tsp. Accessed 8 Dec 2019
21. D-Wave Previews Next-Generation Quantum Computing Platform (2019). https://www.dwavesys.com/press-releases/d-wave-previews-next-generation-quantum-computing-platform. Accessed 8 Dec 2019
22. Dattani N, Szalay S, Chancellor N (2019) Pegasus: the second connectivity graph for large-scale quantum annealing hardware. arXiv:1901.07636