The Quantum Approximate Algorithm for Solving Traveling Salesman Problem

Yue Ruan[1, *], Samuel Marsh[2], Xilin Xue[1], Zhihao Liu[3] and Jingbo Wang[2, *]

Abstract: The Quantum Approximate Optimization Algorithm (QAOA) is an algorithmic framework for finding approximate solutions to combinatorial optimization problems. It consists of interleaved unitary transformations induced by two operators labelled the mixing and problem Hamiltonians. To fit this framework, one needs to transform the original problem into a suitable form and embed it into these two Hamiltonians. In this paper, for the well-known NP-hard Traveling Salesman Problem (TSP), we encode its constraints into the mixing Hamiltonian rather than the conventional approach of adding penalty terms to the problem Hamiltonian. Moreover, we map edges (routes) connecting each pair of cities to qubits, which decreases the search space significantly in comparison to other approaches. As a result, our method can achieve a higher probability for the shortest round-trip route with only half the number of qubits consumed compared to IBM Q's approach. We argue the formalization approach presented in this paper would lead to a generalized framework for finding, in the context of QAOA, high-quality approximate solutions to NP optimization problems.

Keywords: Quantum approximate optimization algorithm, traveling salesman problem, NP optimization problems.

1 Introduction

The Traveling Salesman Problem (TSP), a notorious NP-hard problem, has intrigued computer scientists and mathematicians for over two centuries. It involves a salesman traveling from city to city. The salesman needs to find the shortest possible round-trip route that visits all cities exactly once.

This problem can be abstracted in terms of graphs. The TSP asks for a Hamiltonian cycle (a closed path) that connects every vertex of a weighted graph once with the lowest total sum of edges. Applying brute-force search among all possible permutation of *n* nodes takes time $O(n!)$. The first algorithm to apply dynamic programming solves this problem in $O(n^2 2^n)$ time [Dantzig, Fulkerson and Johnson (1954)]. In the following decades, many approaches, such as Minimum Spanning Trees [Held and Karp (1970)], Branch-and-

[1] School of Computer Science and Technology, Anhui University of Technology, Maanshan, China.

[2] School of Physics, University of Western Australia, Perth, Australia.

[3] School of Computer Science and Engineering, Southeast University, Nanjing, China.

[*] Corresponding Authors: Yue Ruan. Email: yue_ruan@ahut.edu.cn;
Jingbo Wang. Email: jingbo.wang@uwa.edu.au.

Received: 04 February 2020; Accepted: 25 February 2020.

CMC. doi:10.32604/cmc.2020.010001 www.techscience.com/journal/cmc

Bound [Balas and Toth (1983)], Branch-and-Cut [Padberg and Rinaldi (1987, 1991)] have improved on this time-bound. However, since the problem is a NP-hard, these exact algorithms run in exponential time. In practice, various heuristics and approximation algorithms are used to yield sufficiently high-quality approximate solutions. These approaches, including simulated annealing [Geng, Chen, Yang et al. (2011)], genetic algorithms [Grefenstette, Gopal, Rosmaita et al. (1985)], ant colony algorithms [Dorigo and Gambardella (1997)] can find solutions for extremely large problems (millions of cities) within a reasonable time. The solutions are often, with high probability, just 2-3% away from the optimal solution [Rego, Gamboa, Glover et al. (2011)].

In recent years, with the progress of practical quantum computing, such as Quantum Security Communication [Liu, Xu, Yang et al. (2019)] and Quantum Machine Learning [Ruan, Xue, Liu et al. (2017); Liu, Gao, Yu et al. (2018); Liu, Gao, Wang et al. (2019)], the application to approximate combinational optimization has also become tantalizing. In 2014, Farhi et al. [Farhi, Goldstone and Gutmann (2014)] proposed the Quantum Approximate Optimization Algorithm (QAOA) as a candidate to outperform the best classical optimization algorithm in the NISQ (Noisy Intermediate-Scale Quantum) era [Preskill (2018)]. QAOA not only arouses the interest of academia but also intrigues big enterprises. To name a few, IBM [Moll, Barkoutsos, Bishop et al. (2018)], Google [McClean, Boixo, Smelyanskiy et al. (2018)], and Intel [Guerreschi and Matsuura (2019)] have researched this algorithm. In the following sections, we will describe QAOA, then present our QAOA-based scheme for solving the TSP. We finally discuss the advantages and disadvantages of our approach by comparing it with other quantum computing methods.

2 Quantum approximate optimization algorithm

The quantum approximate optimization algorithm (QAOA) [Farhi, Goldstone and Gutmann (2014)] is an algorithmic framework derived from an approximation to the quantum adiabatic algorithm (QAA) [Farhi, Goldstone, Gutmann et al. (2001)]. The QAOA coarsely approximates adiabatic evolution by the application of the Suzuki-Trotter theorem [Nielsen and Chuang (2002)]. The framework breaks the time-dependent QAA Hamiltonian into a sequence of 2p time-independent Hamiltonians. Each of these time-independent operators have the form $e^{-i\beta H_B}$ and $e^{-i\gamma H_C}$ with tunable parameters β and γ. H_C is a diagonal Hamiltonian corresponding to the final QAA Hamiltonian that diagonally encodes the qualities of all possible solutions, and H_B can be considered as a "mixing operator", which serves to 'mix' quantum amplitude between the different solutions. The total adiabatic evolution is then produced by alternating applications of these two operators,

$$\left|\boldsymbol{\beta},\boldsymbol{\gamma}\right\rangle = \boldsymbol{e}^{-i\boldsymbol{\beta}_p H_B}\boldsymbol{e}^{-i\boldsymbol{\gamma}_p H_C}\cdots\boldsymbol{e}^{-i\boldsymbol{\beta}_1 H_B}\boldsymbol{e}^{-i\boldsymbol{\gamma}_1 H_C}\left|\boldsymbol{s}\right\rangle \tag{1}$$

where $\left|\boldsymbol{s}\right\rangle$ is some efficiently-prepared initial state.

When the expectation value of the final QAOA state with respect to H_C is maximized (or minimized), the state $|\beta,\gamma\rangle$ records the optimal solution available to the parameter space. This approach depends on $2p$ parameters of the form β and γ. The values of each of these parameters can be found by a classical optimization algorithm such as Nelder-Mead, with the objective function being the expected quality of the solution. The expected quality of the final QAOA state can be found by sampling. Such a quantum-classical hybrid approach is illustrated in Fig. 1. This approach is regarded as a promising candidate for demonstrating quantum supremacy in the NISQ era [Preskill (2018)]. We note that when $p \to \infty$, the QAOA approaches the quantum adiabatic algorithm (QAA) perfectly and thus guarantees the optimal solution [Farhi, Goldstone and Gutmann (2014)]. Nevertheless we, consistent with prior research, have found that very small p frequently leads to high-quality approximate solutions.

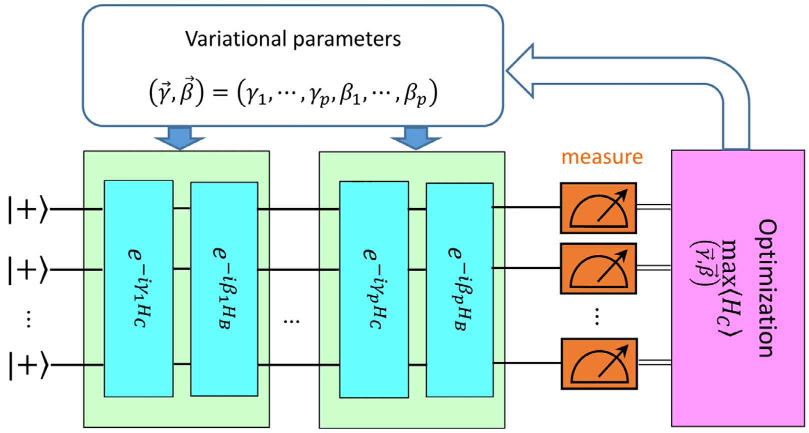


Figure 1: Schematic diagram of hybrid quantum-classical QAOA

3 QAOA-based scheme for solving TSP

3.1 Data structure

We map each city to a vertex i on a graph, with the route between city i and city j corresponding to the edge (i, j). According to the traveling expense between two cities, we assign a weight w_{ij} to each edge. If city i and city j do not have a direct route, set $w_{ij}=\infty$ (in practice, assign a large value). This transforms the problem to that of finding the edge sequence with minimal cost on a complete graph.

The complete graph with n vertices has $n(n-1)/2$ edges. We map each edge to a bit variable, and regard the solution as a binary vector $\vec{x}$, where 1 indicates that the corresponding edge is included in the solution. Thus, the problem Hamiltonian H_C can be defined as:

$$H_C=\sum_{0}^{2^{n(n-1)/2}-1}\vec{x}\cdot\vec{w}|x\rangle\langle x| \tag{2}$$

where vector $\vec{x}=(x_0x_1\cdots x_{n(n-1)/2-1})^T$, and vector $\vec{w}=(w_0w_1\cdots w_{n(n-1)/2-1})^T$ holds the weight of each edge.

We express the Hamiltonian cycle constraint as a polynomial function of $n(n-1)/2$ binary variables $f(x_0,\cdots,x_k,\cdots,x_{n(n-1)/2-1})$. Since each $x_k\in\{0,1\}$, clearly $(x_k)^m=x_k$ and f does not have interaction terms like $x_ix_{i+1}\cdots x_k$ (as each variable/edge can be selected independently). Thus, the polynomial constraint function f can be simplified to

$$f(\vec{x})=\sum_k\overline{c_k}*x_k \tag{3}$$

where $\overline{c_k}$ is the coefficient of x_k . The dimension depends on the specific instance under consideration.

For a graph with n vertices, $\overline{c_k}$ can be defined as a vector with two elements taking the value 1 and the rest 0. The positions where 1 appears indicate the two endpoints of an edge as shown in Fig. 2.

$\vec{x}=x_1x_2...x_k...x_{n(n-1)/2}$ Each x_k = 0/1 represents the

$\downarrow$ non-selection/selection of $n(n-1)/2$ edges

$\vec{c_k}=0...010...010...0$ $\vec{c}_k$ is the coefficient of x_k. The position of '1' in the

$\downarrow \quad \downarrow$ bitstring indicates the two endpoints of an edge.

$\alpha \quad \beta$ indicate the edge (α,β)

Figure 2: the data structure of coefficient $\vec{c_k}$ corresponding to edge x_k in TSP problem

Hence, the constraint that a closed path includes all vertices only once can be described as

$$\sum_{k=0}^{n(n-1)/2-1}\overline{c_k}*x_k=\begin{pmatrix}2\\ \vdots\\ 2\end{pmatrix} \tag{4}$$

We note this equation is a necessary condition, not a sufficient condition, for the constraint of TSP. For example, when the number of vertices is greater than 6, there are independent closed routes (each includes 3 vertices) satisfying Eq. (4), but they do not form a Hamiltonian cycle. We now present an Oracle function *validate*, to check this.

3.2 Oracle function validate

Associated with every NP optimization problem is an efficient algorithm *validate*, which checks if a solution meets the problem constraints [Kolaitis and Thakur (1994)]. The below algorithm is the specific *validate* for TSP, checking if the edge sequence represents a Hamiltonian cycle.

Algorithm 1 *validate*

```
Input: one solution $\vec{x}$ of a city graph with $n$ vertices
Output: True if and only if $\vec{x}$ is a Hamiltonian cycle
$\Gamma=\{\vec{c_i}\}$        /* $\Gamma$ contains all $\vec{c_i}$ having $x_i$ =1 */
If ($|\Gamma| \neq n$) then return "infeasible solution";/* indicate a solution not having $n$ edges */
$\vec{S} = \vec{c_k}$        /* an arbitrary coefficient from $\Gamma$ */
Loop: according to the positions of 1 in vector $\vec{S}$, select $\vec{c_p}$ and $\vec{c_q}$ whose corresponding position also have 1 from $\Gamma$.
$\vec{S} = \vec{S} + \vec{c_p} + \vec{c_q}$
$\Gamma=\Gamma-\{\vec{c_p}\}-\{\vec{c_q}\}$
If ($\vec{S}$ has components with value beyond 0, 1 and 2) then
    return "infeasible solution";
Else If ($\vec{S}$ has components with value 1) then
    Goto Loop;
Else If ($\vec{S}$ has components with value 0) then
    return "infeasible solution";    /*exist disconnected closed paths*/
Else If (all components of $\vec{S}$ have value 2)
    return "feasible solution";
End If
```

The possible number of elements in Γ is no more than $n(n-1)/2$, so the time performance of the above algorithm is bounded by $O(n^2)$ and is thus efficient.

3.3 Construct constraint-encoded H_B

Since the oracle function *validate* determines whether a given solution satisfies the constraints of TSP efficiently, we can use it to obtain a superposition over all feasible solutions.

Prepare a superposition state containing all combinatorial solutions and add an auxiliary qubit with initial value $|0\rangle$. Then, by applying the quantum oracle that maps $|x\rangle|0\rangle \rightarrow |x\rangle|validate(x)\rangle$, we can select all feasible solutions in Ω with a controlled operation on the ancilla:

$$\sum_{x=0}^{2^{\frac{n(n-1)}{2}}-1} |x\rangle \otimes |0\rangle \underrightarrow{validate} \sum_{x\notin\Omega} |x\rangle \otimes |0\rangle + \sum_{x\in\Omega} |x\rangle \otimes |1\rangle \tag{5}$$

In our scheme, we have transformed the city traveling graph into a complete graph, so any vertex sequence is a valid sequence. Using a 6-city example, {A, B, C, D, E, F} represents the edge set {A-B, B-C, C-D, D-E, E-F, F-A}. Changing the order of any two vertices e.g., {B, A, C, D, E, F} means reselecting two edges {B-A, A-C, C-D, D-E, E-F, F-B}. That means, from one feasible solution $\vec{x}$, one can determine another feasible solution $\vec{x}'$ by two 0-1 swaps. Therefore $\vec{x}'$ must lie in the set $\left\{\vec{x}' \mid d(\vec{x},\vec{x}')=4\right\}$ where $d(\cdot,\cdot)$ is the Hamming distance between two binary vectors.

Treating Hamiltonian H_B as a graph, then solutions $\vec{x}$ and $\vec{x}'$ are vertices on this graph connected by an edge. As per [Childs (2004)], the unitary operator e^{-iH_Bt} describes a continuous quantum walk on H_B. The walker can reach all the connected nodes and the walking is restricted in the range defined by the connected subgraph to which the starting node belongs. Thus, encoding constraint in H_B is equivalent to find a way connecting all feasible solutions (vertices) and excluding all infeasible solutions (vertices).

Based on this understanding, one can use the following pseudo-code to generate the rows of H_B:

Algorithm 2 H_B - generation

Input: every feasible solution $\vec{x}$ in superposition state

Output: the constraint-encoded H_B

```
For each $\vec{x} \in \Omega$                /* i.e., $\sum_{x\in\Omega}|x\rangle$ */
    For $\vec{x}' \in \left\{\vec{x}' \mid d(\vec{x}',\vec{x})=4\right\}$
        $\vec{x}' \leftarrow flip\ 4\ bits\ of\ \vec{x}$
    If validate ( $\vec{x}'$ ) then
            generate $|\mathrm{x}\rangle\langle x'| + |\mathrm{x}'\rangle\langle x|$  /* i.e. $H_B(x,x') = H_B(x',x) = 1$ */
    End For
End For
```

The number of flips in the inner For loop is bounded by $O(C_n^4)$, i.e., $O(n^4)$. And since the *validate* function is also efficient, each row of H_B can be efficiently generated. Thus, applying Aharonov's decomposition lemma, H_B can be decomposed as *poly*(*n*) 2×2 combinatorially block diagonal matrices so that an efficient quantum circuit can simulate $e^{-i\beta H_B}$ to the desired accuracy [Aharonov and Ta-Shma (2003)].

3.4 Scheme steps

Step 1 Prepare the superposition state including all feasible solutions.
Since we have oracle function *validate* at hand, prepare the system as per Eq.(5), then do a further projection $I\otimes|1\rangle\langle 1|$, to provide the initial state $|s\rangle = \sum_{x\in\Omega}|x\rangle$.[*]

Step 2 Prepare H_B and H_C as per Eqs. (6) and (2)

Step 3 Generate the circuit to simulate $e^{-i\beta H_B}$ and $e^{-i\gamma H_C}$.
Since H_C is diagonal, it is straightforward to simulate $e^{-i\gamma H_C}$. Applying the scheme from Aharonov et al. [Aharonov and Ta-Shma (2003)], we can also simulate $e^{-i\beta H_B}$ efficiently.

Step 4 Fix *p*, i.e., determine the number of $\{\beta_1,\cdots,\beta_p\}$ and $\{\gamma_1,\cdots,\gamma_p\}$.
As *p* is chosen to be higher, the average solution quality increases at the cost of a deeper quantum circuit.

Step 5 Perform the QAOA state evolution:

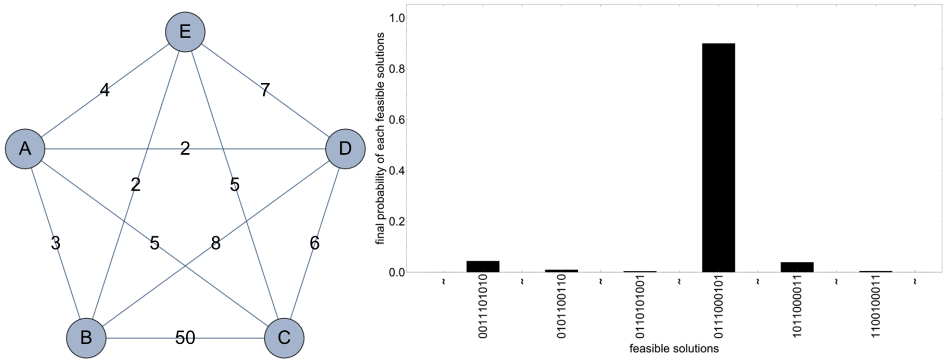
$$|\beta,\gamma\rangle = e^{-i\beta_p H_B}e^{-i\gamma_p H_C}\cdots e^{-i\beta_1 H_B}e^{-i\gamma_1 H_C}|s\rangle$$

Step 6 Measure $\langle\beta,\gamma|H_C|\beta,\gamma\rangle$. Use a classical optimization method to vary the value of $\{\beta_1,\cdots,\beta_p\}$ and $\{\lambda_1,\cdots,\gamma_p\}$, and repeat from ***Step 5***, until convergence to the minimum $\langle\beta,\gamma|H_C|\beta,\gamma\rangle$.

3.5 Numerical simulation

Here we give a concrete 5-city example. The costs of traveling between each city are shown in Fig. 3(a), and the result returned by *p=4* QAOA is shown in Fig. 3(b). According to the edge mapping sequence (A-B, A-C, A-D, A-E, B-C, B-D, B-E, C-D, C-E, D-E), the solution "1010001110" with highest probability represents the edge-set {A-B, A-D, B-E, C-D, C-E}. We can easily verify that this edge-selection is the optimal solution to this specific problem.

[*] An alternative is to prepare an initial state as one of the feasible solutions.



(a) the complete graph of cities

(b) the results of applying QAOA to this TSP instance

Figure 3: a TSP problem solved by QAOA with p=4

4 Discussion

The numerical simulations were conducted on a computer with an i7 CPU and 8 GB main memory. For numbers of cities greater 6, 15+ qubits are required to represent the solution space, which quickly becomes infeasible for classical simulation.

The classical optimization method used to obtain the above results is the Nelder Mead simplex algorithm. We replace with other classical optimization algorithms provided by Mathematica© to observe the influence of different classical optimization methods for obtaining the final probability. We find that the QAOA ends up with almost identical results by exploiting varied classical optimization algorithms. This observation manifests the classical optimization method is not a vital factor in the performance of QAOA. Rather, p determines the depth of the quantum circuit, which determines the complexity of QAOA.

IBM Q also presented a method to solve TSP [IBM Q (2018)]. They take the approach named variational quantum Eigensolver (VQE) [Peruzzo, McClean, Shadbolt et al. (2014), Wang, Higgott and Brierley (2019)], a quantum-classical algorithm proposed earlier than QAOA. In comparison, our method has two advantages.

First, we map each edge to a qubit so the total qubits we used are $n(n-1)/2$. IBM Q uses two-dimensional qubits-mapping, one is for each vertex, and the other is for the order in the vertex sequence. Hence, the qubits needed are n^2. Clearly, our method consumes only half the number of qubits, a constant-factor improvement.

Second, we improve on the handling of constraints. IBM Q takes the conventional method for encoding constraints, adding "penalty" terms to the problem Hamiltonian whenever a constraint is violated [Lucas (2014)]. These terms add extra energy scales to the problem Hamiltonian, which in practice increases the difficulty of getting the final solution. In addition, there is the distinct possibility of the algorithm returning an invalid solution. Instead, our approach does not use penalty terms. We instead encode the constraint directly in the mixer H_B. This idea stemmed from the work of Hen et al. [Hen and Spedalieri (2016), Hen and Sarandy (2016)] and discussed in the context of QAOA by Marsh et al. [Marsh and Wang (2019)] and Hadfield et al. [Hadfield, Wang, O'Gorman et al. (2019)]. In this paper, we present a unique H_B -encoding approach, and demonstrate that encoding constraint in H_B (mixing Hamiltonian) indeed excludes the infeasible solutions. More details are provided in Ruan et al. [Ruan, Marsh, Xue et al. (2020)].

5 Conclusion

In this paper, we have proposed a QAOA-based scheme to solve the well-known NP-hard Traveling Salesman Problem. We use a unique approach in comparison to previous methods, in mapping edges to qubits and encode the closed tour constraint in the mixing operator H_B . The results and comparison with other schemes demonstrated our approach's effectiveness and efficiency for solving TSP, albeit only for the small problem instances that can be analyzed with classical computers.

We consider QAOA as a high-performing algorithm for resolving NP-optimization problems. However, QAOA, the hybrid quantum-classical variational optimization scheme, can only be considered a heuristic. Theoretically, QAOA is the approximation of QAA only produces the optimal result with certainty when $p \to \infty$ [Farhi, Goldstone and Gutmann (2014)]. However, the presented approach for solving TSP reinforces the observation made in Farhi et al. [Farhi, Goldstone and Gutmann (2014)], that a small value p appears adequate to obtain the optimal (or at least, a high-quality) solution. If this is indeed the case for large problem instances, QAOA is a powerful utility that can be applied to a wide range of real-world optimization problems to obtain high-quality approximate solutions efficiently.

Funding Statement: This work is supported by the Natural Science Foundation, China (Grant No. 61802002) and Natural Science Foundation of Anhui Province, China (Grant No. 1708085MF162).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Aharonov, D.; Ta-Shma, A. (2003): Adiabatic quantum state generation and statistical zero knowledge. *Proceedings of The Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pp. 20-29.

Balas, E.; Toth, P. (1983): *Branch and Bound Methods for the Traveling Salesman Problem*. Carnegie-Mellon University Pittsburgh PA, Management Sciences Research Group.

Childs, A. M. (2004): Quantum information processing in continuous time. (*Ph.D. Thesis) Massachusetts Institute of Technology*.

Dantzig, G.; Fulkerson, R.; Johnson, S. (1954): Solution of a large-scale traveling-salesman problem. *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393-410.

Dorigo, M.; Gambardella, L. M. (1997): Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53-66.

Farhi, E.; Goldstone, J.; Gutmann, S.; Lapan, J.; Lundgren, A. et al. (2001): A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, vol. 292, no. 5516, pp. 472-475.

Farhi, E.; Goldstone, J.; Gutmann, S. (2014): A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028.

Geng, X.; Chen, Z.; Yang, W.; Shi, D.; Zhao, K. (2011): Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search. *Applied Soft Computing*, vol. 11, no. 4, pp. 3680-3689.

Grefenstette, J.; Gopal, R.; Rosmaita, B.; Van Gucht, D. (1985): Genetic algorithms for the traveling salesman problem. *Proceedings of the First International Conference on Genetic Algorithms and Their Applications*, pp. 160-168.

Guerreschi, G. G.; Matsuura, A. Y. (2019): QAOA for max-cut requires hundreds of qubits for quantum speed-up. *Scientific Reports*, vol. 9, pp. 6903.

Hadfield, S.; Wang, Z.; O'Gorman, B.; Rieffel, E.; Venturelli, D. et al. (2019): From the quantum approximate optimization algorithm to a quantum alternating operator Ansatz. *Algorithms*, vol. 12, no. 2.

Held, M.; Karp, R. M. (1970): The traveling-salesman problem and minimum spanning trees. *Operations Research*, vol. 18, no. 6, pp. 1138-1162.

Hen, I.; Spedalieri, F. M. (2016): Quantum annealing for constrained optimization. *Physical Review Applied*, vol. 5, no. 3, pp. 412-416.

Hen, I.; Sarandy, M. S. (2016): Driver Hamiltonians for constrained optimization in quantum annealing. *Physical Review A*, vol. 93, no. 6.

IBM Q. (2018): https://quantum-computing.ibm.com/jupyter/tutorial/advanced/aqua/optimization/max_cut_and_tsp.ipynb.

Kolaitis, P. G.; Thakur, M. N. (1994): Logical definability of NP optimization problems. *Information and Computation*, vol. 115, no. 2, pp. 321-353.

Lucas, A. (2014) Ising formulations of many NP problems. *Frontiers in Physics*, no. 2.

Liu, W.; Xu, Y.; Yang, J.; Yu, W.; Chi, L. (2019): Privacy-preserving quantum two-party geometric intersection. *Computers, Materials & Continua*, vol. 60, no. 3, pp. 1237-1250.

Liu, W.; Gao, P.; Wang, Y.; Yu, W.; Zhang, M. (2019): A unitary weights based one-iteration quantum perceptron algorithm for non-ideal training sets. *IEEE Access*, vol. 7, pp. 36854-36865.

Liu, W.; Gao, P.; Yu, W. B.; Qu, Z. G.; Yang, C. N. (2018): Quantum relief algorithm. *Quantum Information Processing*, vol. 17, no. 10, pp. 280.

Marsh, S.; Wang, J. B. (2019): A quantum walk-assisted approximate algorithm for bounded NP optimization problems. *Quantum Information Processing*, vol. 18, no. 3, pp. 61.

McClean, J. R.; Boixo, S.; Smelyanskiy, V. N.; Babbush, R.; Neven, H. (2018): Barren plateaus in quantum neural network training landscapes. *Nature Communications*, vol. 9, no. 1, pp. 1-6.

Moll, N.; Barkoutsos, P.; Bishop, L. S.; Chow, J. M.; Cross, A. et al. (2018): Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, vol. 3, no. 3.

Padberg, M.; Rinaldi, G. (1987): Optimization of a 532-city symmetric traveling salesman problem by branch and cut. *Operations Research Letters*, vol. 6, no. 1, pp. 1-7.

Padberg, M.; Rinaldi, G. (1991): A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, vol. 33, no. 1, pp. 60-100.

Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.; Zhou, X. et al. (2014): A variational Eigenvalue solver on a photonic quantum processor. *Nature Communications*, no. 5, pp. 4213.

Preskill, J. (2018): Quantum computing in the NISQ era and beyond. *Quantum*, vol. 2, pp. 79.

Nielsen, M. A.; Chuang, I. (2002): *Quantum Computation and Quantum Information*. Cambridge University Press.

Rego, C.; Gamboa, D.; Glover, F.; Osterman, C. (2011): Traveling salesman problem heuristics: leading methods, implementations and latest advances. *European Journal of Operational Research*, vol. 211, no. 3, pp. 427-441.

Ruan, Y.; Xue, X.; Liu, H.; Tan, J.; Li, X. (2017): Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance. *International Journal of Theoretical Physics*, vol. 56, no. 11, pp. 3496-3507.

Ruan, Y.; Marsh, S.; Xue, X.; Li, X.; Liu, Z. et al. (2020): Quantum approximate algorithm for NP optimization problems with constraints. arXiv:2002.00943.

Wang, D.; Higgott, O.; Brierley, S. (2019): Accelerated variational quantum Eigensolver. *Physical Review Letters*, vol. 122, no. 14.