第50卷第12期 中国科学技术大学学报 Vol. 50, No. 12
2020年12月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Dec. 2020
Received: 2020-10-18; Revised: 2020-11-02 doi:10.3969/j.issn.0253-2778.2020.12.008

Quantum algorithms for ranking nodes of network

LU Binhan[1,2], HAN Yongjian[1,2], WU Yuchun[1,2], LI Ye[3], AN Ningbo[3], DOU Menghan[3], ZHAO Dongyi[3], GUO Guoping[1,2]*

1. CAS Key Laboratory of Quantum Information, School of Physics, University of Science and Technology of China, Hefei 230026 China;
2. CAS Center For Excellence in Quantum Information and Quantum Physics, University of Science and Technology of China, Hefei 230026, China;
3. Origin Quantum Computing Technology Company Limited, Hefei 230026, China
* Corresponding author. E-mail: gpguo@ustc.edu.cn

Abstract: Ranking the node of a network is one of the central problems in a complex network. Here, An improved SpringRank method was proposed that builds an adaptive physical model with variable-spring connected between nodes and a novel penalty function. By minimizing the penalty function, the method can rank the nodes of a directed and weighted complex network. To decrease the computation complexity which increases too fast with the number of nodes, quantum algorithms were used to speed up the process of minimizing the penalty functions. The convexity enables us to find the minimum by solving a linear system. When the linear system has the properties of sparsity and a small conditional number, we use the HHL algorithm to find the minimum of the penalty function by solving the linear equation. And we use the QITE algorithm to find the minimum by updating the parameters iteratively when the linear system doesn't have those properties. Lastly, using the quantum simulator QPanda, we implemented the algorithms for several networks and gave the right ranking results.

Keywords: network ranking; adaptive physical model; penalty function minimizing; quantum linear system solver; quantum imaginary time evolution

CLC number: TP3-05; TP399; O59 **Document code**: A

1 Introduction

For a complex network with a large number of nodes[1-8,54], one of the most important information is to tell which nodes are more important than others, in other words, ranking nodes is one of the most important tasks for a complex network[9]. It has been applied in many fields, such as epidemic control[10-12], power communication networks[13,14], biological networks[15]. biological networks[16,17], and link prediction[18].

The key point of the algorithms to rank the nodes of complex networks is to determine the importance of a node[19,20]. In Degree Centrality (DC)[21] and Betweenness Centrality (BC)[22] algorithms, the importance of a node is determined by the number of its neighbor nodes. The properties of the adjacent matrix of the complex network, the eigenvalues, also imply the importance of nodes, for example, the spectral method[20,23] and Eigenvector Centrality (EC)[24] which are the core algorithms of some search engines, PageRank algorithm[25] and LeaderRank algorithm[23]. Other algorithms are more flexible, such as Minimum Violation Ranking[26,27], SyncRank[28], and SpringRank method[20], they describe the complex network by an adaptive physical model and introduces penalty function which depends on the specific networks. By minimizing the penalty function, one can obtain the rank of the nodes. However, those methods can not solve the ranking issues for the directed and weighted complex networks.

In this work, we propose an improved SpringRank method by introducing variable-spring in adaptive physical model and construct a novel penalty function. Minimizing the penalty function by classical algorithms needs a lot of computation resources, hence we use quantum optimizers to minimize the penalty function[29,30].

Since the penalty function is convex, the minimum can be found by solving a linear equation. If the linear system is sparse and has a small conditional number, we use the HHL algorithm to solve the linear equation. The HHL algorithm has speedup compared to the classical

Citation: LU Binhan, HAN Yongjian, WU Yuchun, et al. Quantum algorithms for ranking nodes of network. J. Univ. Sci. Tech. China, 2020, 50(12): 1507-1515.

linear equation solver exponentially[31]. In the cases that the linear system is non-sparse or has a large conditional number, which leads to the result that the quantum circuit has qubit number and depth too large for the NISQ devices[35] which is the most feasible form of quantum computer in the near future, the Quantum Imaginary Time Evolution (QITE) algorithm can be used[36]. These two kinds of optimizers are both tested in this paper and results show that the quantum algorithm is capable of ranking the nodes in a complex network.

2 Model

The SpringRank method[20] defines a Hamiltonian, which is also a penalty function, converts the ranking problem to finding the ground state of Hamiltonian, or in other words, the minimum of the penalty function. In the SpringRank method, springs are placed between each pair of connected nodes. In Ref. [20], Bacco set the rest length of each spring to be 1, which is incomplete for weighted and directed networks. In this paper, we generalize the springs with fixed lengths to be the variable-springs whose lengths are determined by the difference of the characters of two connected nodes[37,38]. The characters describe the importance of each node.

Their definitions are as follows. In a weighted directed network, the nodes connected to more high-weight edges are usually more important than the others. So we define the first character strength.

Definition 2.1 Strength

$$S_i = \sum_{j \in V_{\text{in}}} w_{ij} + \sum_{j \in V_{\text{out}}} w_{ji} \tag{1}$$

where w_{ij} is the weight of edge $i \to j$, V_{in} (V_{out}) is the ingoing (outgoing) node set of i.

The second character called irreplaceable number R_i is defined as follows.

Definition 2.2 Local irreplaceable number R_i is the total number of the shortest path between every neighbor pairs of node i that run through node i.

This character describes that if the node is deleted from the network, whether it's neighbors can find another shortest path to keep the connection between them.

However, in general, R_i cannot give a complete description of local connection information. For example, in Eq. (1), node B and node E have the same irreplaceable number. For node B, there are 6 different shortest paths through it which are ABC, CBA, ABE, EBA, ABD, DBA. And similarly, the shortest paths through E are BEH, HEB, CEG, GEC, DEF, FED.

Therefore, one more character should be defined to give a complete description of the local connection information in the network.

Definition 2.3 Local uniqueness

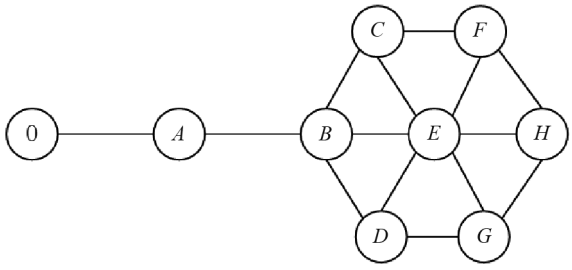

Fig. 1 9-node network.

$$\widetilde{R}_i = \frac{R_i}{\widehat{R}_i} \tag{2}$$

where $\widehat{R}_i = |V_{\text{neighbor of } i}| \times (|V_{\text{neighbor of } i}| - 1)$ is the number of total path through node i.

The character called local uniqueness describes that among all the paths between the neighbors of node i, the ratio of the shortest path. This character also cannot describe the local connection information in the network. In Fig. 1, we have $\widetilde{R}_B = \frac{1}{2}$ and $\widetilde{R}_E = \frac{1}{5}$ which is a reasonable result. While $\widetilde{R}_A = 1$, it is unreasonable to infer that node A is more important than node B.

Based on Definitions 2.2 and 2.3, we define a new character U_i for each node i as

Definition 2.4

$$U_i = R_i \widetilde{R}_i \tag{3}$$

We get $U_A = 2$, $U_B = 3$, $U_E = \frac{6}{5}$ in Fig. 1, which means that in terms of local connection information, their importance order is $B > A > E$. This is a reasonable result.

According to Ref. [20], there is a kind of mutual influence of these characters between pairs of nodes. They are defined as

$$\Delta U_{ji} = \frac{(w_{ij} + w_{ji})^2}{S_i S_j}(U_j - U_i) \tag{4}$$

$$\Delta S_{ji} = \frac{(w_{ij} + w_{ji})^2}{S_i S_j}(S_j - S_i) \tag{5}$$

The mutual influence has the function that larger quantities will increase the corresponding quanties of its neighbors, and vice versa. So the mutual influence between U_i and S_i is defined as

$$\widehat{U}_i = \lambda \sum_{j \in V_{\text{in}} \cup V_{\text{out}}} \Delta U_{ji} + U_i \tag{6}$$

$$\widehat{S}_i = \lambda \sum_{j \in V_{\text{in}} \cup V_{\text{out}}} \Delta S_{ji} + S_i \tag{7}$$

For computation convenience, we renormalize the S_i and U_i to the range $[1, n]$,

$$\widetilde{S}_i = 1 + \frac{n-1}{\widehat{S}_{\max} - \widehat{S}_{\min}}(\widehat{S}_i - \widehat{S}_{\min}) \tag{8}$$

$$\widetilde{U}_i = 1 + \frac{n-1}{\widehat{U}_{\max} - \widehat{U}_{\min}}(\widehat{U}_i - \widehat{U}_{\min}) \tag{9}$$

We divide the nodes into several categories according to their importance. The number of categories n and the number of nodes m should satisfy the condition $n \leqslant m$.

Our model based on the SpringRank method generalizes the rest length from a fixed value to the difference between the linear combination of the characters $\widetilde{U}_i$ and $\widetilde{S}_i$ defined above. The linear combination of these two characters $\lambda_1\widetilde{S}_i + \lambda_2\widetilde{U}_i$ is a measurement of the importance of different nodes, where λ_1, λ_2 are two combination coefficients that satisfy $\lambda_1+\lambda_2=1$. If the difference is large, the ranking of two different nodes should also be large, and vice versa. Based on this idea, the general elastic potential energy is defined as

$$H_{ij} = \frac{1}{2}w_{ij}\{s_i - s_j - [\lambda_1(\widetilde{U}_i - \widetilde{U}_j) + \lambda_2(\widetilde{S}_i - \widetilde{S}_j)]\}^2 \tag{10}$$

s_i is the ranking value of the nodes that should be found by our optimizer. And the Hamiltonian for the whole network is the sum of all elastic potential energy between each pair Eq. (11). The Hamiltonian is also known as the penalty function. The minimum of the penalty function is the ground state energy of the Hamiltonian. Now we have transformed the ranking problem to the problem of finding the minimum of the penalty function,

$$H(\vec{s}) = \sum_{j=0}^{n-1}\sum_{i\in V_j} H_{ji} = \frac{1}{2}\sum_{j=0}^{n-1}\sum_{i\in V_j} w_{ij}\{s_i - s_j - [\lambda_1(\widetilde{U}_i - \widetilde{U}_j) + \lambda_2(\widetilde{S}_i - \widetilde{S}_j)]\}^2 \tag{11}$$

3 Algorithms

3.1 Quantum linear system solver

After getting the Hamiltonian Eq. (11), we have to find the minimum of it. Because it has a quadratic structure, it is convex. We will get $\nabla H(\vec{s}) = 0$ if and only if at the minimum point. So the first algorithm we use to find the minimum is solving the equation $\nabla H(\vec{s}) = 0$, which can also be written as

$$A\vec{s} = \vec{b} \tag{12}$$

where

$$A_{ij} = \begin{cases} \sum_{k=1}(w_{kj} + w_{jk}) - (w_{ij} + w_{ji}), & i = j; \\ -(w_{ij} + w_{ji}), & i \neq j \end{cases} \tag{13}$$

$$\vec{b}_j = \sum_{i=1}^{n}\{w_{ij}[\lambda_1(\widetilde{U}_j - \widetilde{U}_i) + \lambda_2(\widetilde{S}_j - \widetilde{S}_i)] - w_{ji}[\lambda_1(\widetilde{U}_i - \widetilde{U}_j) + \lambda_2(\widetilde{S}_i - \widetilde{S}_j)]\} \tag{14}$$

This is a linear equation, so our problem is converted into solving the linear equation. Our goal is to find a state,

$$|\widetilde{x}\rangle := \frac{\sum_i \widetilde{x}_i |i\rangle}{\| \sum_i \widetilde{x}_i |i\rangle \|} \tag{15}$$

such that $\| |\widetilde{x}\rangle - |x\rangle \| \leqslant \epsilon$. The amplitude of $|x\rangle$ satisfies $\vec{x} = \boldsymbol{A}^{-1}\vec{b}$.

Before solving the equation, we need to encode the coefficients of matrix A and the inhomogeneous term $\vec{b}$ of the linear system into a quantum state that can be manipulatedin a quantum circuit. In Ref. [32], a quantum data structure based on Quantum Random Access Memory (QRAM)[33,34] is proposed.

Theorem 3.1 Let $M \in \mathbb{R}^{m\times n}$, there is a data structure to store M such that the quantum algorithm with access to the data structure can perform the following unitary

$$|0\rangle \rightarrow \sum_{ij} \| M_{ij} \| |i,j\rangle \tag{16}$$

in time $O(\log^2 mn)$.

The proof of Theorem 3.1 is in Refs. [33,34]. This algorithm tells us that the input algorithm depends only logarithmically on the size of the system.

The famous quantum linear system solver was developed by Harrow, Hassidim, and Lloyd (HHL)[31]. The algorithm solves the linear equation $\boldsymbol{A}x = \vec{b}$, which requires $\mathcal{O}(d\log N/\epsilon)$ oracle query and has the success rate with $\mathcal{O}(1/\kappa)$ where N represents the dimension of the equation, d is the sparsity of the matrix, ϵ the target precision, and κ the conditional number of matrix $\boldsymbol{A}$.

The result of the algorithm is stored in the amplitude of the quantum state, and it can be extracted through a sampling algorithm called l_∞-tomography[40].

Theorem 3.2 Given an algorithm U such that $U|0\rangle = |x\rangle$, there is a l_∞-tomography algorithm with time complexity $O(\log d/\delta^2)$ that produces unit vector $\widetilde{X} \in \mathbb{R}^d$ such that $\| \widetilde{X} - x \|_\infty \leqslant \delta$ with probability at least $1-1/\mathrm{poly}(d)$.

The proof of Therem 3.2 is in Ref. [40]. This theorem tells us that we can recover the classical information from the quantum state with an algorithm that depends only logarithmically on the dimension of the state.

In conclusion, the overall complexity of the input, computation, and output process of the quantum linear system solver only depends logarithmically on the dimension of the linear equation, thus our algorithm has exponential speedup compared to the classic algorithm.

3.2 Using QITE

When the linear system is too hard to solve by the HHL

algorithm, such as the matrix $\boldsymbol{A}$ is not sparse or the conditional number κ is too large. The HHL algorithm is not suitable for these situations. A quantum-classical hybrid algorithm, the QITE algorithm is used in these situations[36]. The process can be decomposed into three steps.

① Get the Hamiltonian.

② Design the ansatz for the QITE algorithm based on the Hamiltonian.

③ Use imaginary time τ in the evolution operator and let the initial state evolve to the ground state of the system.

The design of the ansatz for variational quantum imaginary time evolution is an open problem. We give a general ansatz-design scheme to this problem.

① Calculate the value of local information $M_i = \lambda_1 \widetilde{U}_i + \lambda_2 \widetilde{S}_i$ of each node.

② The node with the largest M value is selected as the control node.

③ Add X gate to the front of the control node, control RY gate in the middle, and RX gate to the end of the control node.

Since the QITE algorithm is a quantum-classical hybrid algorithm, so it is suitable for error mitigation and can be implemented by shallow quantum circuits on the NISQ computer[36].

Now we give an example of constructing the ansatz. For simplicity, we only consider the binary ranking, which is the value of s_i only has two values 0, 1 and

$$\frac{I-Z}{2} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{17}$$

Eq. (17) has eigenvalues of 0 and 1. If each s_i is replaced by $I \otimes I \otimes \cdots \left[\frac{I-Z}{2}\right]_i \cdots \otimes I$, we can rewrite the Hamiltonian of the system as

$$H = \sum_{j=0}^{n-1} \sum_{i \in V_j} a_{ji} Z_j Z_i + \sum_{j=0}^{n-1} a_j Z_j + aI \tag{18}$$

Given the Hamiltonian Eq. (18), and an initial state $|\psi(0)\rangle$, the QITE is defined as

$$|\psi(\tau)\rangle = \frac{1}{\sqrt{\langle\psi(0)|\exp(-H\tau)|\psi(0)\rangle}} \exp(-H\tau)|\psi(0)\rangle \tag{19}$$

which is equivalent to the Wick rotating Schrödinger equation[36].

$$\frac{\partial |\psi(\tau)\rangle}{\partial \tau} = -(H - E(\tau))|\psi(\tau)\rangle \tag{20}$$

where $E(\tau) = \langle\psi(\tau)|H|\psi(\tau)\rangle$.

By replacing $|\psi(\tau)\rangle$ with $|\psi(\vec{\theta}(\tau))\rangle$, we can solve the imaginary time evolution by a quantum circuit.

After applying McLachlan's variational principle[45,46], we get

$$\delta \left\| \left(\frac{\partial}{\partial \tau} + H - E_\tau\right)|\psi(\tau)\rangle \right\| = 0 \tag{21}$$

We then get a differential equation,

$$\sum_j A_{ij}\dot{\theta}_j = C_i \tag{22}$$

where

$$A_{ij} = R\left(\frac{\partial\langle\psi(\tau)|}{\partial\theta_i}\frac{\partial|\psi(\tau)\rangle}{\partial\theta_j}\right) \tag{23}$$

$$C_{ij} = R\left(-\sum_j \lambda_j \frac{\partial\langle\psi(\tau)|}{\partial\theta_i} h_j |\psi(\tau)\rangle\right) \tag{24}$$

by measuring the result of the quantum circuit in each iteration.

In a small time interval $\delta\tau$, the differentiation of θ is $\dot{\theta} = A^{-1}(\tau)\vec{C}(\tau)$. We use the update method as

$$\vec{\theta}(\tau + \Delta\tau) \leftarrow \vec{\theta}(\tau) - \delta(k)\vec{r}\,\mathrm{sgn}(A^{-1}(\tau)\vec{C}(\tau)) \tag{25}$$

where $\vec{r}$ is a vector whose elements are random values from $[0, 1)$, $\delta(k) = \delta\tau\ Q^k$, $Q \in [0.9, 0.95]$. The algorithm of ranking nodes based on QITE is as follows Algorithm 3.1.

Algorithm 3.1 QITE algorithm for ranking

Input: Network weight matrix W, number of iteration steps T, time interval $\Delta\tau$, Number of upthrow $upNum$; $u = 0$; $t = 0$;

Output: Ground state $|\varphi\rangle = S_{\text{best}}|\bar{0}\rangle$;

1: Calculate the node strength $\widehat{D}$ and local irreplaceable value $\widehat{U}$ after data preprocessing;

2: Generate the Pauli decomposition form $H = \sum_l \lambda_l h_l$ of the Hamiltonian matrix of the system;

3: Set up ansatz $S(\vec{\theta}(0))$ according to the network topology;

4: **while** $t<T$ **do**

5: Calculate $A_{ij} = R\left(\frac{\partial\langle\psi(\tau)|}{\partial\theta_i}\frac{\partial|\psi(\tau)\rangle}{\partial\theta_j}\right)$ through quantum circuit measurement to get A;

6: Calculate $C_i = R\left(-\sum_j \lambda_j \frac{\partial\langle\psi(\tau)|}{\partial\theta_i} h_j |\psi(\tau)\rangle\right)$ through quantum circuit measurement to get C;

7: Calculate the generalized inverse of A;

8: $\vec{\theta}(\tau + \Delta\tau) \leftarrow \vec{\theta}(\tau) - \delta(k)\vec{r}\,\mathrm{sgn}(A^{-1}(\tau)\vec{C}(\tau))$;

9: Limit parameter $\vec{\theta}(\tau+\Delta\tau)$ to $[0, 2\pi)$;

10: Update the ansatz $S(\vec{\theta}(\tau+\Delta\tau))$ according to the parameter $\vec{\theta}(\tau+\Delta\tau)$;

11: Calculate the expectation $E(\vec{\theta}(\tau+\Delta\tau))$ according to the ansatz $S(\vec{\theta}(\tau+\Delta\tau))$;

12: if $E(\vec{\theta}(\tau+\Delta\tau)) < E(\vec{\theta}(\tau))$ then

13: $S_{\text{best}} \leftarrow S(\vec{\theta}(\tau+\Delta\tau))$;

14: **end if**

15: **end while**

Works show that QITE[36] has great practical value[41-44]. For QITE, the complexity of each iteration

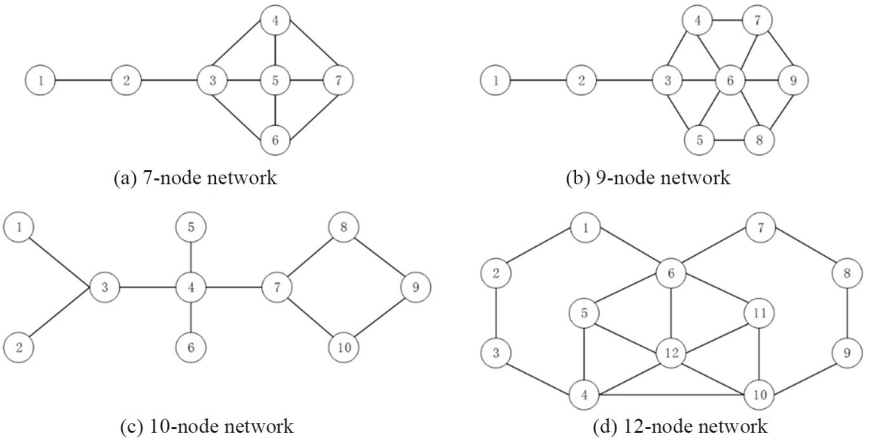

Fig. 2 Network of different nodes.

is $\mathcal{O}(N_P{}^2 N_A + N_P N_H N_C + \log N_P)$, where N_P is the number of parameters used in the ansatz, N_H is the number of items in the Pauli matrix in the Hamiltonian, N_A is the number of measurements required to determine the elements of matrix A with required accuracy, N_C is the number of measurements required to determine the element of C with required accuracy. The complexity of solving the linear equations of $\dot{\theta}$ is the complexity of the HHL algorithm which is $\mathcal{O}(\log N_P)$.

3.3 Advantages over classical algorithms

Given a matrix A and a vector $\vec{b}$, we find a vector $\vec{x}$ such that $\vec{Ax} = \vec{b}$ where A has a size $N \times N$ and has conditional number κ. The fastest known classical algorithms can find $\vec{x}$ in time scaling roughly as $N\sqrt{\kappa}$. While the HHL algorithm has a runtime of $\log N$ and κ[31]. So there is an exponential speed up when the optimization problem can be solved by solving a linear system when the A is sparse and has small κ.

When the matrix A is not sparse or the conditional number κ is too large, the HHL circuit may have qubit number or depth too large for NISQ devices to execute because when the circuit gets longer, we need quantum error correction (QEC) algorithm which requires qubit number that is intractable in the NISQ devices[50,51]. NISQ devices are the most feasible form of the quantum computer under the technical constraint. QITE algorithm can be used in this situation. QITE is a hybrid algorithm with a shallow quantum circuit and leaves the higher level calculation to the classical computer[36]. Even without error correction, noise in the shallow circuit can be suppressed via error mitigation, which indicates that QITE is feasible to the NISQ devices[41]. In Refs. [52, 53], Bravyi gives proof of the existence of quantum advantage with shallow circuits such as QITE. The quantum algorithms running in a constant time are strictly more powerful than their classical counterparts and they are probably better at solving certain linear algebra problems associated with binary quadratic forms which are exactly our optimization problem.

4 Evaluation

We use QPanda[48] (an SDK for building, running, and simulating quantum algorithms) to evaluate our methods.

We use the following networks Fig. 2 to benchmark our algorithms.

4.1 Using the HHL algorithm

Tab. 1 and Fig. 3 shows the numerical calculation results of the 7-node network. The classical results are obtained by using the Moore-Penrose Generalized Inverse (MPGI) method[49]. The more important the node i is, the larger the value s will get. It can be seen that quantum and classical algorithms have the same ranking results.

Tab. 1 Numerical calculation results of 7-node network.

Node number	s(MPGI)	s(HHL)
1	−0.467501	−0.406306
2	−0.037204	−0.031877
3	0.850730	0.739830
4	−0.119542	−0.104146
5	0.051456	0.044585
6	−0.119542	−0.104146
7	−0.158398	−0.137941

4.2 Using QITE

We also test the QITE algorithm with the networks in Algorithm 3.1. The ansatz is implemented according to Eq. (18). After randomly choosing the initial parameters θ_i, we use stochastic gradient descent to update the parameters. We set the $\delta\tau = 1.0$ and the

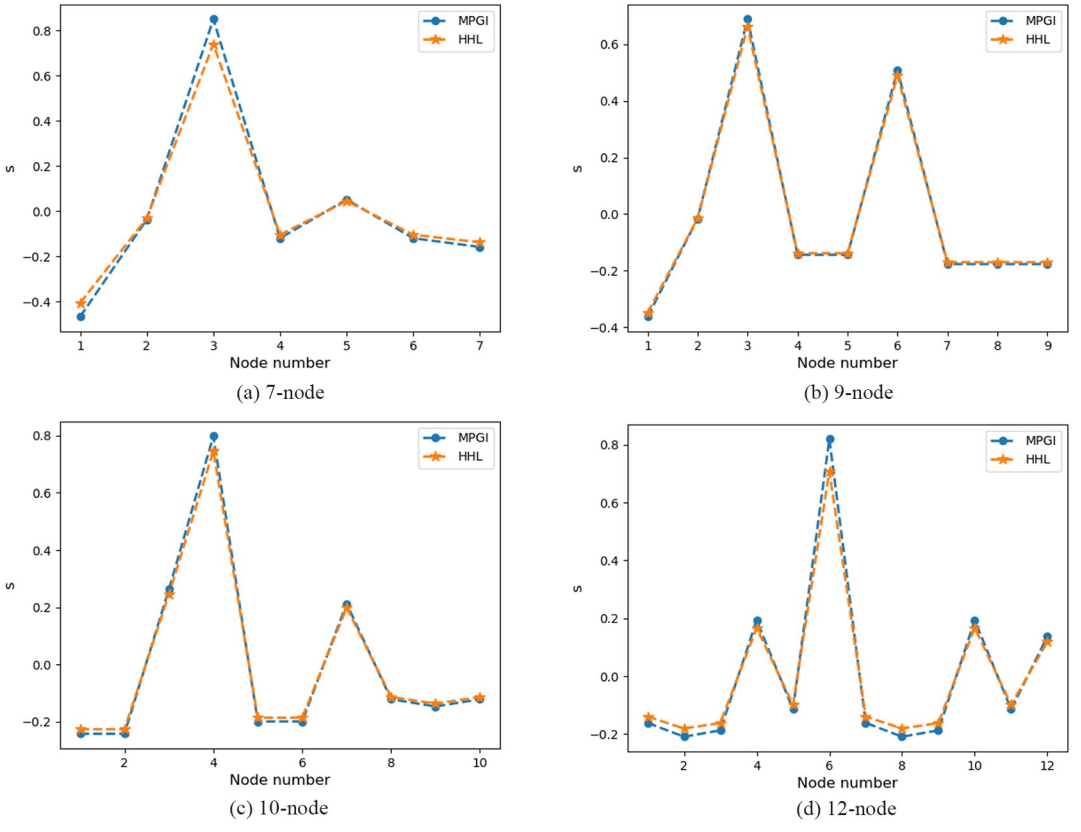

Fig. 3 Comparison of the HHL and MPGI results of four different node networks.

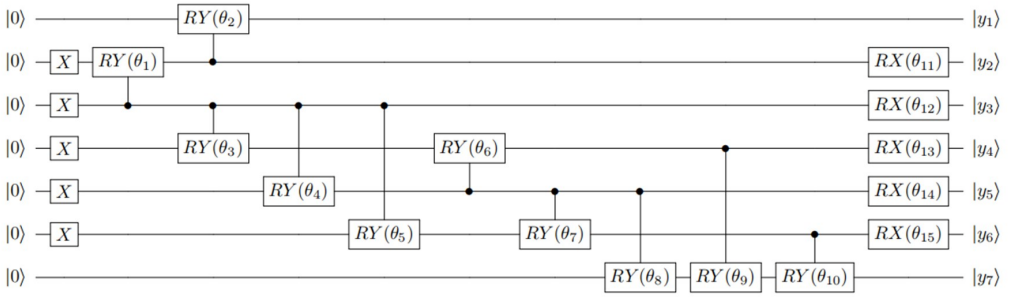

Fig. 4 Ansatz quantum circuit of 7-node network.

number of the steps to be $T=50$.

According to the ansatz constructing rule, the ansatz of the 7-node network is shown in Fig. 4.

The penalty function value descendent curves are shown in Fig. 5. It can be seen that our parameter update policy is not sensitive to the initial parameters $\vec{\theta}_i$. All of the curves converge quickly to the ground state energy value. This indicates the robustness of our method.

The ground states corresponding to Hamiltonian of the 7, 9, 10, 12 node networks are as follows. The results in Tab. 2 are reasonable, for example, in Fig. 2 (c), if any one of the nodes 3, 4, and 7 is deleted from the network, the network will become two sub-network that is not connected.

Tab. 2 The ground state of each Hamiltonian.

Network	Ground state
7-node	$\|0010000\rangle$
9-node	$\|001001000\rangle$
10-node	$\|0011001000\rangle$
12-node	$\|000101000101\rangle$

5 Conclusions

In this paper, we propose an improved SpringRank method by introducing variable-spring which is determined by local connection information of the

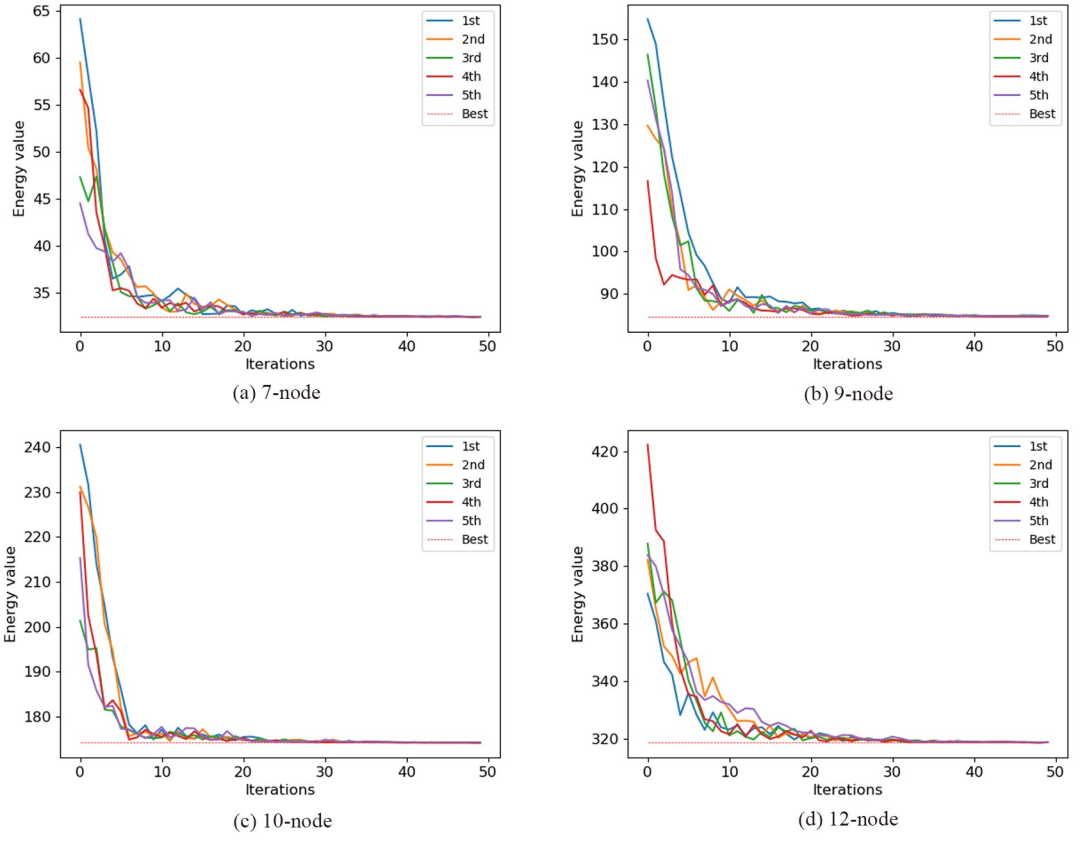

Fig. 5 The penalty function value descendent curve. Different curves in the same subfigure represent the learning processes with different initial parameters.

network, and construct a penalty function. Because the classical algorithms demand unpractical computation resources, quantum algorithms are used to find the minimum of the penalty function. Convexity of the penalty function allows us to find the minimum by solving a linear equation. If the linear system is sparse and has a small conditional number, we use the HHL algorithm which has speedup compared to the classical linear equation solver exponentially to solve the linear equation. If the linear system is non-sparse or has a large conditional number, we use the QITE algorithm which is a kind of shallow circuit feasible to NISQ devices to find the minimum point iteratively. Experiments test both kinds of algorithms and results show that the quantum algorithm can rank the nodes in a complex network.

Only the binary ranking result is obtained in the experiments. What we solve is the snapshot of the current network, that is, the order of importance of nodes corresponding to the topology of the network at this moment. If the network structure is changing, the importance of its nodes has been changing.

In the future, we will further improve and develop our approach in the following three aspects. First, we will try to explore the methods with more accuracy, in other words, with the ability to divide the nodes into more categories i. e. the ranking score can be (0, 1, 2, 3,...). Second, there are other kinds of quantum algorithms for combinational optimization problems such as quantum annealing and Quantum Approximate Optimization Algorithm (QAOA), hence we will compare and find the suitable quantum algorithms for different situations. Third, we will promote our work so that it can solve the dynamic network scheduling problem.

Acknowledgements

This work is supported by the National Key Research and Development Program of China (973 Program) (No. 2016YFA0301700) and the Anhui Initiative in Quantum Information Technologies (No. AHY080000).

Conflict of interest

The authors declare no conflict of interest.

Author information

LU Binhan is a postgraduate student under the supervision of Prof. Wu Yuchun at University of Science and Technology of China(USTC). He receive his B. S. degree in Applied Physics from Jinan University(JNU) in 2019. He mainly focuses on the optimization of quantum algorithm compilation.

GUO Guoping (corresponding author) is professor and doctoral supervisor of University of Science and Technology of China, deputy director of Key Laboratory of Quantum Information of Chinese Academy of Sciences, vice president of Institute of Microelectronics of University of Science and Technology of China, and research leader of semiconductor quantum dot quantum chip of University of Science and Technology of China. Chief scientist of national Key Basic Research and Development Program Class A (super 973), winner of national fund for Distinguished Young Scholars. Founder and chief scientist of Origin Quantum Computing Technology Co., Ltd..

References

[1] Alm E, Arkin A P. Biological networks. Current Opinion in Structural Biology, 2003, 13(2): 193-202.
[2] Mangan N M, Brunton S L, Proctor J L, et al. Inferring biological networks by sparse identification of nonlinear dynamics. IEEE Transactions on Molecular, Biological and Multi-Scale Communications, 2016, 2(1): 52-63.
[3] Wang B, Pourshafeie A, Zitnik M, et al. Network enhancement as a general method to denoise weighted biological networks. Nature Communications, 2018, 9(1): 1-8.
[4] Shi C, Li Y, Zhang J, et al. A survey of heterogeneous information network analysis. IEEE Transactions on Knowledge and Data Engineering, 2016, 29(1): 17-37.
[5] Barucca P, Bardoscia M, Caccioli F, et al. Network valuation in financial systems. Mathematical Finance, 2020,30(4): 1181-1204.
[6] Bardoscia M, Battiston S, Caccioli F, et al. Pathways towards instability in financial networks. Nature Communications, 2017, 8(1): 1-7.
[7] Squartini T, Caldarelli G, Cimini G, et al. Reconstruction methods for networks: The case of economic and financial systems. Physics Reports, 2018, 757: 1-47.
[8] Cheng X, Zhao H. Modeling, analysis and mitigation of contagion in financial systems. Economic Modelling, 2019, 76: 281-292.
[9] Newman M. Networks. 2nd edition. Oxford: Oxford University Press, 2018.
[10] Hindes J, Schwartz I B. Epidemic extinction and control in heterogeneous networks. Physical Review Letters, 2016, 117(2): 028302.
[11] Wang J, Wang X, Wu J. Inferring metapopulation propagation network for intra-city epidemic control and prevention. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. The Association for Computing Machinery, 2018: 830-838.
[12] Jiang J, Zhou T. Resource control of epidemic spreading through a multilayer network[J]. Scientific Reports, 2018, 8(1): 1-10.
[13] Wang Z, Ran Z, Chen X, et al. Critical node identification for electric power communication network based on topology and services characteristics. In: MATEC Web of Conferences. EDP Sciences, 2018, 246: 03023.
[14] Liu H, Luo S, Gan J, et al. Research on node importance of power communication network based on Multi-Attribute Analysis. In: 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2020, 1: 2683-2687.
[15] Das K, Samanta S, Pal M. Study on centrality measures in social networks: A survey. Social Network Analysis and Mining, 2018, 8(1): 13.
[16] Del Ferraro G, Moreno A, Min B, et al. Finding influential nodes for integration in brain networks using optimal percolation theory. Nature Communications, 2018, 9(1): 2274.
[17] Liu X, Hong Z, Liu J, et al. Computational methods for identifying the critical nodes in biological networks[J]. Briefings in Bioinformatics, 2020, 21(2): 486-497.
[18] Wang T, He X S, Zhou M Y, et al. Link prediction in evolving networks based on popularity of nodes. Scientific Reports, 2017, 7(1): 1-10.
[19] Mo H, Deng Y. Identifying node importance based on evidence theory in complex networks. Physica A: Statistical Mechanics and its Applications, 2019, 529: 121538.
[20] De Bacco C, Larremore D B, Moore C. A physical model for efficient ranking in networks. Science Advances, 2018, 4(7): eaar8260.
[21] Albert R, Jeong H, Barabási A L. Diameter of the World-Wide Web. Nature, 1999, 401(6749): 130-131.
[22] Freeman L C. A set of measures of centrality based on betweenness[J]. Sociometry, 1977, 40(1): 35-41.
[23] Lu L Y, Zhang Y C, Chi H Y, et al. Leaders in social networks, the delicious case. Plos One, 2011, 6(6): e21202.
[24] Bonacich P. Power and centrality: A family of measures. American Journal of Sociology, 1987, 92(5): 1170-1182.
[25] Restrepo J G, Ott E, Hunt B R. Characterizing the dynamical importance of network nodes and links. Physical Review Letters, 2006, 97(9):094102.
[26] Ali I, Cook W D, Kress M. On the minimum violations ranking of a tournament. Management Science, 1986, 32(6): 660-672.
[27] Gupte M, Shankar P, Li J, et al. Finding hierarchy in directed online social networks. In: Proceedings of the 20th international conference on World Wide Web. 2011: 557-566.
[28] Cucuringu M. Sync-rank: Robust ranking, constrained ranking and rank aggregation via eigenvector and SDP synchronization. IEEE Transactions on Network Science and Engineering, 2016, 3(1): 58-79.
[29] Izaac J A, Zhan X, Bian Z, et al. Centrality measure based on continuous-time quantum walks and experimental realization. Physical Review A, 2017, 95(3): 032318.
[30] Chawla P, Mangal R, Chandrashekar C M. Discrete-time quantum walk algorithm for ranking nodes on a network. Quantum Information Processing, 2020, 19: 1-21.
[31] Harrow A W, Hassidim A, Lloyd S. Quantum algorithm for linear systems of equations. Physical Review Letters, 2009, 103(15): 150502.
[32] Kerenidis I, Prakash A. Quantum recommendation systems. https://arxiv.org/abs/1603.08675v3.
[33] Giovannetti V, Lloyd S, Maccone L. Architectures for a quantum random access memory. Physical Review A, 2008, 78(5): 052310.
[34] Giovannetti V, Lloyd S, Maccone L. Quantum random access memory. Physical Review Letters, 2008, 100(16): 160501.

[35] Preskill J. Quantum Computing in the NISQ era and beyond. Quantum, 2018, 2: 79.
[36] McArdle S, Jones T, Endo S, et al. Variational ansatz-based quantum simulation of imaginary time evolution. npj Quantum Information, 2019, 5(1): 1-6.
[37] Wittenbaum G M, Hubbell A P, Zuckerman C. Mutual enhancement: Toward an understanding of the collective preference for shared information. Journal of Personality and Social Psychology, 1999, 77(5): 967-978.
[38] Fornito A, Zalesky A, Bullmore E. Fundamentals of Brain Network Analysis. Pittsburgh, PA: Academic Press, 2016.
[39] Dervovic D, Herbster M, Mountney P, et al. Quantum linear systems algorithms: A primer. https://arxiv. org/abs/1802. 08227v1.
[40] Kerenidis I, Landman J, Prakash A. Quantum algorithms for deep convolutional neural networks. https://arxiv. org/abs/1911. 01117.
[41] Xu X, Sun J, Endo S, et al. Variational algorithms for linear algebra. https://arxiv. org/abs/1909. 03898.
[42] Lubasch M, Joo J, Moinier P, et al. Variational quantum algorithms for nonlinear problems. Physical Review A, 2020, 101(1): 010301.
[43] Fontanela F, Jacquier A J, Oumgari M. Aquantum algorithm for linear PDEs arising in Finance. https://ssrn. com/abstract=3499134.
[44] Endo S, Kurata I, Nakagawa Y O. Calculation of the Green's function on near-term quantum computers. https://arxiv. org/abs/1909. 12250.
[45] McLachlan A D. A variational solution of the time-dependent Schrödinger equation[J]. Molecular Physics, 1964, 8(1): 39-44.
[46] Broeckhove J, Lathouwers L, Kesteloot E, et al. On the equivalence of time-dependent variational principles. Chemical Physics Letters, 1988, 149(5-6): 547-550.
[47] Sandvik A W, Vidal G. Variational quantum Monte Carlo simulations with tensor-network states. Physical Review Letters, 2007, 99(22): 220602.
[48] Origin Quantum Computing. QPANDA 2. https://github. com/OriginQ/QPanda-2.
[49] Penrose R. A generalized inverse for matrices. In: Mathematical Proceedings of the Cambridge Philosophical Society. Cambridge: Cambridge University Press, 1955, 51(3): 406-413.
[50] Fowler A G, Mariantoni M, Martinis J M, et al. Surface codes: Towards practical large-scale quantum computation. Physical Review A, 2012, 86(3): 032324.
[51] Li Gushu, Ding Yufei, Xie Yuan. Tackling the qubit mapping problem for NISQ-era quantum devices. In: Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. ACM, 2019.
[52] Bravyi S, Gosset D, König R. Quantum advantage with shallow circuits. Science, 2018, 362: 308-311.
[53] Bravyi S, Sergey D, König R, et al. Quantum advantage with noisy shallow circuits. Nature Physics, 2020, 16: 1040-1045.
[54] Paulin J, Calinescu A, Wooldridge M. Agent-based modeling for complex financial systems. IEEE Intelligent Systems, 2018, 33(2): 74-82.

基于自适应物理模型的量子网络排序算法

卢斌汉[1,2],韩永建[1,2],吴玉椿[1,2],李叶[3],安宁波[3],窦猛汉[3],赵东一[3],郭国平[1,2*]

1. 中国科学技术大学物理学院中科院量子信息重点实验室,安徽合肥 230026;
2. 中国科学技术大学中科院量子信息与量子科技创新研究院,安徽合肥 230026;
3. 合肥本源量子计算科技有限责任公司,安徽合肥 230026

摘要: 对网络节点进行排序是复杂网络分析的核心问题之一. 提出了一种改进的 SpringRank 算法. 该算法基于一个把节点之间连接视为静止长度可变的弹簧的自适应物理模型,并基于此定义一个新型罚函数. 通过最小化罚函数,该算法可以对有向和加权复杂网络的节点进行排序. 为了避免经典算法中计算复杂度随着节点数量的增加而过快增加的情况,使用量子算法加速罚函数最小化过程. 罚函数的凸性使我们能够通过求解线性系统的方式找到最小值. 当线性系统具有稀疏且条件数较小的性质时,使用量子线性求解器 HHL 算法找到罚函数的最小值. 如果线性系统没有这两个性质,则使用量子虚时演化 QITE 算法通过迭代方法找到最小值. 最后,使用量子模拟器 QPanda 对多个网络用所提出的两种求最小值算法进行了节点排序测试,实验结果显示两种算法都能给出正确的排序结果.

关键词: 网络节点排序;自适应物理模型;最小化罚函数;量子线性方程求解器;量子虚时演化