

SCIENTIFIC REPORTS

OPEN

Link Prediction based on Quantum-Inspired Ant Colony Optimization

Zhiwei Cao^{1,2,3}, Yichao Zhang^{1,2}, Jihong Guan^{1,2} & Shuigeng Zhou^{4,5}

Received: 18 August 2017

Accepted: 8 August 2018

Published online: 06 September 2018

Incomplete or partial observations of network structures pose a serious challenge to theoretical and engineering studies of real networks. To remedy the missing links in real datasets, topology-based link prediction is introduced into the studies of various networks. Due to the complexity of network structures, the accuracy and robustness of most link prediction algorithms are not satisfying enough. In this paper, we propose a quantum-inspired ant colony optimization algorithm that integrates ant colony optimization and quantum computing to predict links in networks. Extensive experiments on both synthetic and real networks show that the accuracy and robustness of the new algorithm is competitive in respect to most of the state of the art algorithms. This result suggests that the application of intelligent optimization to link prediction is promising for boosting its accuracy and robustness.

Many complex systems, including social, biological, information and technological systems can be well described as networks where nodes represent individuals or organizations and edges represent the relationships or interactions between them. But real-world networks are typically incomplete or inaccurate, which makes it hard to completely and precisely depict the properties of these networks. To handle this problem, researchers turn to link prediction to refine the networks before their specific studies on them^{1,2}.

Link prediction in complex networks concerns the probability of establishing a link between two disconnected nodes. In the past decade, there are many applications of link prediction for various networks^{3–16}. For instance, in online social networks, such as Facebook, Weibo and Twitter, link prediction is used to recommend registered users to connect with someone they know but not recognized in the network^{5–7}. As a personalized service, an accurate recommendation is likely to promote its users' loyalty⁵. In biological networks, such as protein-protein interaction networks and metabolic networks, the links are typically neither complete (high false positives) nor highly reliable (high false negative)^{8–10}. Accurate link prediction based on known network structure and some specific biological information is helpful for designing targeted experiments, which may substantially reduce experimental time and cost^{11–14}. In the application to monitoring the network of criminals, link prediction is used to discover the possible connections among criminals (including potential criminals), which is useful for locating some specific criminals and thus detecting and disrupting terror attacks^{15,16}.

So far, topology-based link prediction algorithms can be roughly divided into three categories. The first category is based on various generating mechanisms, for instance the common neighbours (CN) algorithm^{17–19} and the cannistraci-resource-allocation algorithm (CRA, which can be also called Cannistraci-Hebb network automata model (CH) based on local community paradigm)^{20–22}. Their computational complexity is relatively low, while precision is not so satisfying. The second category is based on probabilistic models, for instance the stochastic block model (SBM)^{23,24} and the fast probability block model (FBM)²⁵. Relatively, their precision is generally higher, while computational complexity is likewise higher. Thus, they are not suitable for large-scale networks. The third category is model-free, such as the structural perturbation method (SPM)^{26,27}. Among them, the SBM and FBM usually perform poorly because of the issues of inference in the maximum likelihood procedure, which may fall in local optimum or in the configuration of maximum likelihood that are unrealistic. The CRA (or CH) algorithm performs the best specifically on the networks that are generated by a hyperbolic geometry because its generative mechanism (which is a local community paradigm) is associated to that geometry. The SPM algorithm is the most

¹Department of Computer Science and Technology, Tongji University, 4800 Cao'an Road, Shanghai, 201804, China.

²Key Laboratory of Embedded System and Service Computing (Tongji University), Ministry of Education, Shanghai, 200092, China. ³Information Security Technology Division, The Third Research Institute of Ministry of Public Security, 339 Bi Sheng Road, Shanghai, 201204, China. ⁴Shanghai Key Laboratory of Intelligent Information Processing, Shanghai, 200433, China. ⁵School of Computer Science, Fudan University, 220 Handan Road, Shanghai, 200433, China. Zhiwei Cao and Yichao Zhang contributed equally. Correspondence and requests for materials should be addressed to J.G. (email: jhguan@tongji.edu.cn)

robust and well-performed in general regardless of the hidden generative model of the network. This may originate from its model free property, which is adaptive since it does not assume any underlying mechanism.

In this paper, we propose a quantum-inspired ant colony optimization algorithm (QACO) to predict missing links in networks, which integrates ant colony optimization and quantum computing²⁸. To sum up, our contributions are three-folds. First, based on the assumption that the pheromone of artificial ants can reflect the importance of a path, we introduce a biased ant colony optimization algorithm to remedy incomplete datasets. In our algorithm, the visibility integrates the quasi-local information of nodes. Second, quantum bits and quantum logic gates are introduced into the ant colony optimization algorithm, which can effectively prevent the optimization process from falling into a local optimum. Third, extensive experiments are conducted on many benchmark networks. Our experimental results show that the accuracy and robustness of the QACO algorithm is competitive in respect to most of the state of the art algorithms.

Results

The QACO algorithm. *Overview.* Consider an undirected network $G = (V, E)$, the corresponding complete network G' is constructed by connecting all the disconnected node pairs in G . Let all artificial ants randomly walk in G' , where links and nodes are allocated some amount of pheromone. The probability that a link or node is visited by ants is proportional to its pheromone.

Assume an ant visits n nodes, which results in a walking path. The probability that an ant travels from node v_i to node v_j is p_{ij} . The value of p_{ij} depends on the pheromone of the path, the visibility of link (v_i, v_j) and the quantum pheromone of v_j . After an ant reaches its destination (say v_j), the pheromone on the links and nodes in the path will be updated according to a certain rule. In turn, the updated pheromone on the links and nodes will affect the paths of the ants in the next iteration. Generally, the pheromone and visibility of the links will lead the ants to the globally optimal paths, since following the quantum pheromone is an effective way to avoid local optimums. Finally, the pheromone τ_{ij} and visibility η_{ij} on link (v_i, v_j) can, to some extent, reflect the similarity between v_i and v_j .



Quantum Pheromone.

- Quantum computation: The basic principle of quantum computation is that the relative phase and the probability amplitude of each ground state of the superposition state are not constant. The occurrence probability of each ground state varies with time, leading to the corresponding variations of the superposition state. Quantum computation is based on a device called quantum gate, which can realize a logical transformation in a certain time interval. The properties of quantum computation include interference, superposition and parallelism. In this paper, we apply quantum bits to representing the quantum pheromone of nodes and quantum rotation gate to updating quantum pheromone of ants^{29,30}.
- Quantum pheromone representation: In quantum computation, $|0\rangle$ and $|1\rangle$ denote two basic states of microscopic particles. An arbitrary state of a single quantum bit can be represented by a linear combination of the two basic states. The sign $| \rangle$ is called Dirac mark, which represents an eigenstate in quantum mechanics. Each quantum bit is typically in a superposition state, which is a combination of the two eigenstates. Thus, it can be represented by $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are a pair of complex numbers, which denote the probability amplitudes of quantum states. For the quantum state $|\varphi\rangle$, the probabilities of collapsing to $|0\rangle$ and $|1\rangle$ are $|\alpha|^2$ and $|\beta|^2$ respectively, where $|\alpha|^2 + |\beta|^2 = 1$.
A quantum bit represents two states $|0\rangle$ and $|1\rangle$. Naturally, a quantum bit of length m can represent 2^m different states. The probability amplitude of individual j with m quantum bits is thus defined as $P_j = [\alpha_1 | \alpha_2 | \dots | \alpha_m]$, where $|\alpha_i|^2 + |\beta_i|^2 = 1$ for $i = 1, 2, \dots, m$.
In the QACO, the quantum pheromone is represented by the quantum bit. Let the size of population be n . The quantum pheromone is defined as $p = (p_1, p_2, p_3, \dots, p_n)$, where p_j ($j = 1, 2, \dots, n$) is the quantum pheromone of individual j ^{31,32}.
- Quantum pheromone updating: The core of ant colony optimization algorithm is that the ants select their paths according to the density of pheromone, which can be applied to link prediction. For each ant, the probability of choosing a node is proportional to the density of pheromone left on the node. The update of the quantum pheromone intensity of a node can be implemented by updating the rotation angle of the quantum rotating gate in QACO. Concretely, the quantum bit of a node is updated by tuning the probability amplitude of quantum bit in the quantum rotating gate. Here, the quantum rotating gate is defined as:

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix}, \quad (1)$$

with $i = 1, 2, \dots, m$. Here, m is the number of ants used, $[\alpha_i^{t+1}, \beta_i^{t+1}]^T$ is the probability amplitude of the i -th quantum bit at the t -th iteration. θ_i is the rotation angle of the i -th quantum bit. Its size and orientation can be determined according to the formula $\theta_i = \Delta\theta * \text{sign}(\alpha_i * \beta_i)$. $\text{sign}(\alpha_i * \beta_i)$ is a sign function. For $\alpha_i * \beta_i > 0$, $\text{sign}(\alpha_i * \beta_i) = 1$; for $\alpha_i * \beta_i < 0$, $\text{sign}(\alpha_i * \beta_i) = -1$; and for $\alpha_i * \beta_i = 0$, $\text{sign}(\alpha_i * \beta_i) = 0$. $\Delta\theta$ usually falls in the range of $[0.01\pi, 0.08\pi]$ ³³.

Parameters setting.

- Parameter initialization: At the beginning, we uniformly set the pheromone on all links to

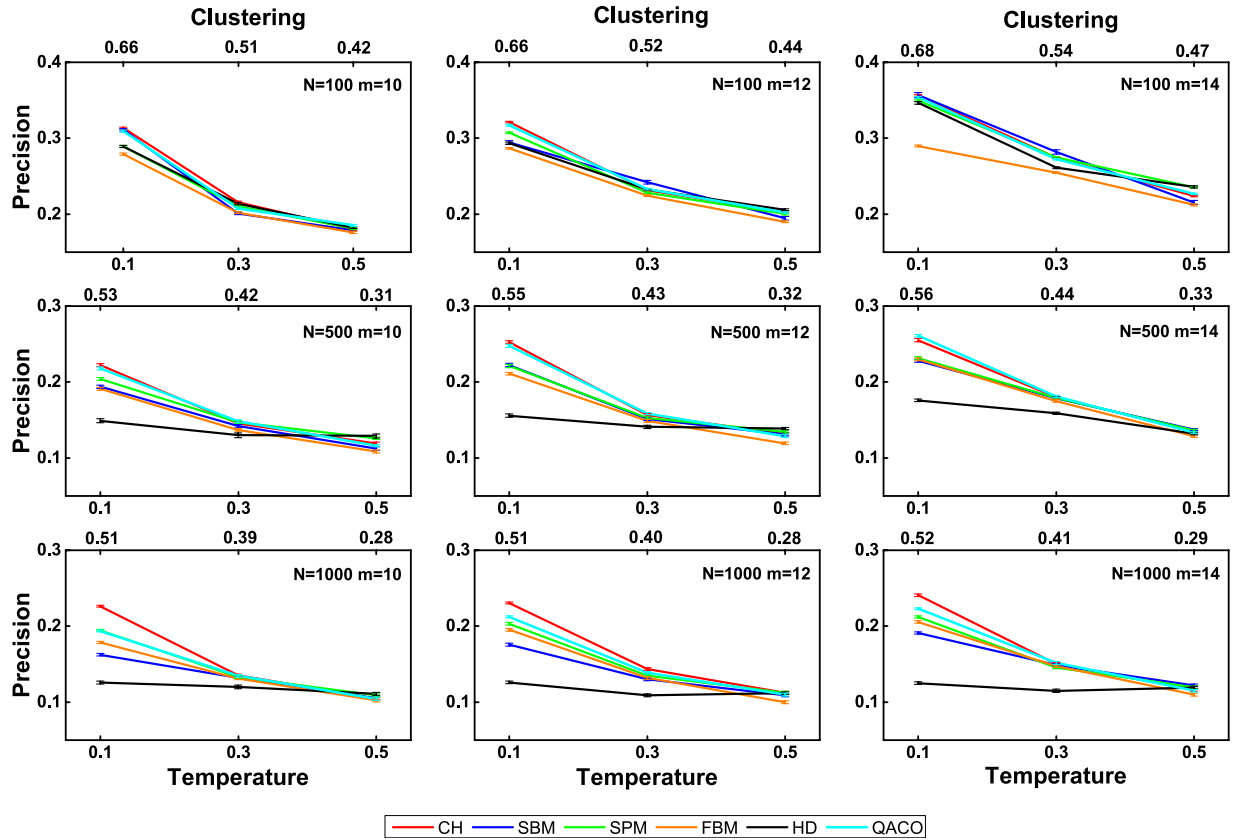


Figure 1. The precision evaluation of six algorithms with 90% of the links used as the training set on the nPSO networks with 8 communities. Synthetic networks are generated by the nonuniform PSO model with parameters $\gamma = 3$ (power-law degree distribution exponent), $m = [10, 12, 14]$ (half of average degree), $T = [0.1, 0.3, 0.5]$ (temperature, inversely related to the clustering coefficient), $N = [100, 500, 1000]$ (network size) and 8 communities. For each combination of parameters, 100 networks are generated. For each parameter combination, the plots report the mean precision and standard error over the random iterations. Note that for SBM only 10 networks are considered due to the high time complexity. In addition, HD is the hyperbolic distances between the nodes in the original network.

$$\tau_{ij} = \delta, \quad (2)$$

where δ is a constant.

- On the other hand, we set the initial value of visibility of link (v_i, v_j) to

$$\eta_{ij} = \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{[k(z)]^\iota} + \omega \sum_{x, y \in l_{i \rightarrow j}} \frac{1}{[k(x)k(y)]^\iota}, \quad (3)$$

where ι is a constant, varying with the topology of network. Following the settings of previous studies³⁴, we set $\omega = 0.01$. $\Gamma(i)$ and $\Gamma(j)$ denote the sets of i and j 's neighbors, respectively. k_x, k_y and k_z denote the degrees of x, y and z , respectively. $l_{i \rightarrow j}$ denotes the node set in the paths from node i to node j , the length of which is 3. x and y are intermediate nodes in the set $l_{i \rightarrow j}$.

Finally, we set the intensity of quantum pheromone at node v_j to

$$\mu_j = \frac{1}{|\alpha_j|^2}, \quad (4)$$

where $|\alpha_j|^2$ denotes the probability that the quantum state of the j -th quantum bit collapses to $|0\rangle$, while $1 - |\alpha_j|^2$ denotes the probability collapsing to $|1\rangle$.

- State transition rule: In each iteration, the transition probability that an ant k moves from v_i to v_j is defined as

$$p_{ij}^k = \frac{[\tau_{ij}]^\lambda [\eta_{ij}]^\kappa [\mu_j]^\nu}{\sum_l [\tau_{il}]^\lambda [\eta_{il}]^\kappa [\mu_l]^\nu}, \quad (5)$$

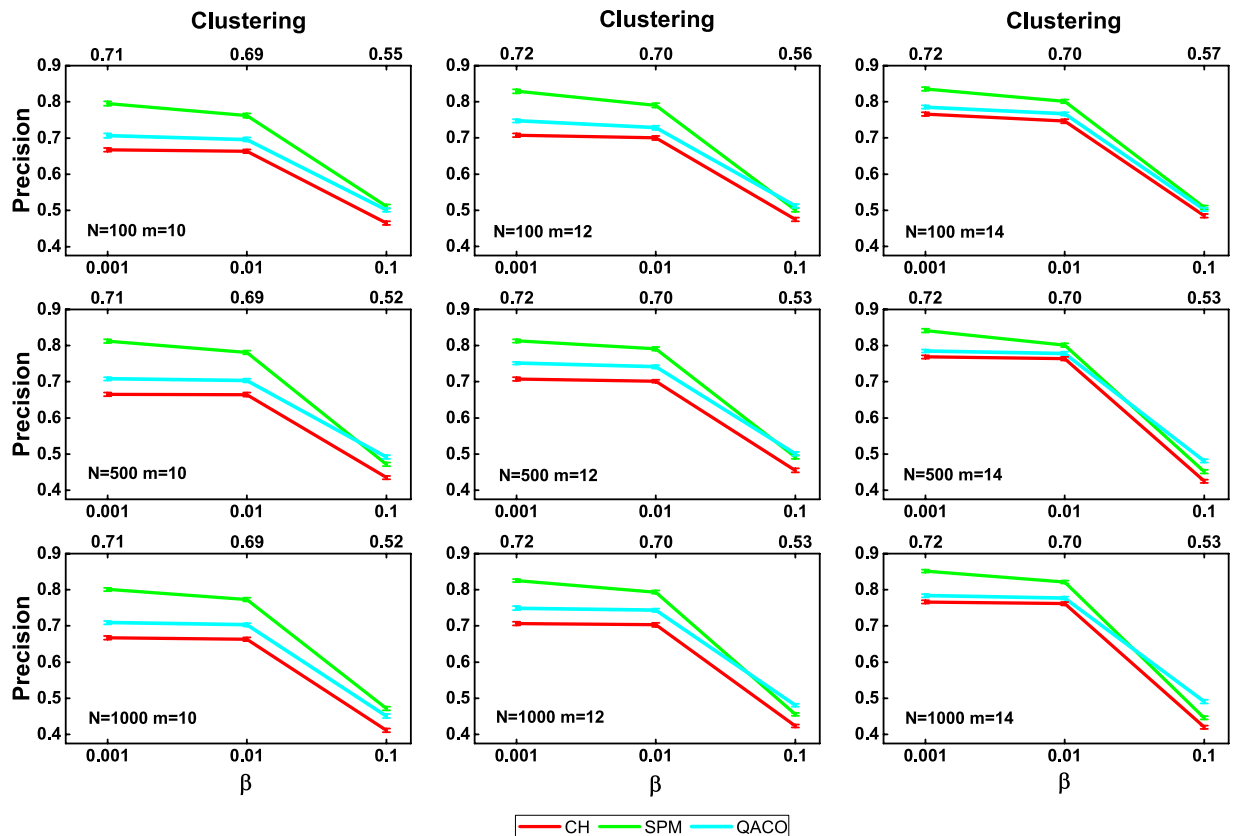


Figure 2. The precision evaluation of three algorithms with 90% of the links used as the training set on the Watts-Strogatz networks. Synthetic networks are generated by the Watts-Strogatz model with parameters $N = [100, 500, 1000]$ (network size), $m = [10, 12, 14]$ (half of average degree) and $\beta = [0.001, 0.01, 0.1]$ (rewiring probability). For each combination of parameters, 100 networks are generated. For each parameter combination, the plots report the mean precision and standard error over the random iterations.

where τ_{ij} and η_{ij} are the pheromone and visibility of link (v_i, v_j) , respectively. μ_j is the quantum pheromone intensity of node v_j . The parameters λ , κ and ν are used to control the impact of the link pheromone, visibility and node pheromone intensity, respectively.

- Fitness function: In each iteration, an ant walks through a path of N nodes. The path of the k -th ant is denoted by $S(k) = (v_1, v_2, v_3, \dots, v_n)$, where $v_i \in V$ is the i -th node in the path. The fitness function is defined to evaluate each path and update the pheromone. Here, the fitness function is defined as $Q(S) = C * \frac{1}{n} \sum_{i=1}^n d(v_i)$, where C is a positive constant and $d(v_i)$ is the degree of node v_i . Generally, a path with more densely-connected nodes gets a higher score.
- Pheromone updating: After each iteration, the QACO algorithm updates the pheromone of each link according to the following formula:

$$\tau_{ij}^{new} = \rho \cdot \tau_{ij}^{old} + \Delta\tau(i, j), \quad (6)$$

where $\Delta\tau(i, j)$ is the variation of pheromone on link (v_i, v_j) , which is defined as $\sum_{k=1}^m \Delta\tau_k(i, j)$. Here, $\Delta\tau_k(i, j)$ is the pheromone released by ant k on link (v_i, v_j) , ρ is the trajectory persistence, which is in the range of $[0, 1]$. Obviously, $\Delta\tau(i, j)$ is proportional to the frequency that the link is visited by ants. If we consider only the fitness Q in the update of pheromone, the gaps of pheromone among links may grow. In this case, the paths of ants are likely to fall into local optimums. To remedy this drawback, β_k is introduced to Eq. (6). In Eq. (6), $\Delta\tau_k(i, j)$ is defined as $Q \cdot (1 - |\beta_k|^2)$, where the probability that ant k selects node v_j is proportional to $|\beta_k|^2$. In this way, we can effectively prevent that the gap of the pheromone among links drastically grows. In other words, it can keep the optimizing process from falling into local optimums.

- Termination conditions and evaluations: Once the number of iterations exceeds a prespecified threshold N_c , the optimizing process will be terminated. Finally, we can obtain a matrix $\tau + \varepsilon \cdot \eta$, which is normally referred to as the score matrix in some ref.³⁵. To evaluate the performance of the QACO algorithm for link prediction, we will compare the precision of our algorithm to that of a number of existing algorithms in the performance evaluation section.

Algorithm 1. QACO for link prediction.

Require: A : the adjacency matrix of the network;
 N_c : the number of iterations;
 δ : a nonzero constant;
 ι : a constant;
 ε : a constant between 0 and 1;

Ensure: Score: the score matrix;

```

1: /* Initialize Parameter */
2:  $t \leftarrow 1$ ;  $\tau_{ij} \leftarrow \delta$ ;  $\eta_{ij} \leftarrow \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{1}{[k(z)]^\iota} + \omega \sum_{x, y \in l_{i \rightarrow j}} \frac{1}{[k(x)k(y)]^\iota}$ ;  $\alpha_i \leftarrow \frac{1}{\sqrt{2}}$ ;  $\beta_i \leftarrow \frac{1}{\sqrt{2}}$ 
3: while  $t < N_c$  do
4:   /* Randomly Selection */
5:   for  $k = 1$  to  $m$  do
6:     the  $k_{th}$  ant is randomly assigned to a node  $v_1$ ;
7:   end for
8:   /* Probability Selection */
9:   for  $k = 1$  to  $m$  do
10:    for  $i = 2$  to  $n$  do
11:       $nowVertex \leftarrow Tabu(k, i - 1)$ ;
12:       $randValue \leftarrow rand(1)$ ;
13:      for  $j = 1$  to  $n$  do
14:        if  $randValue \leq p(nowVertex, j)$  then
15:           $Tabu(k, i) \leftarrow j$ ;
16:        end if
17:      end for
18:    end for
19:  end for
20:  /* Calculate the fitness */
21:  for  $k = 1$  to  $m$  do
22:    Calculate the fitness of the path left by the  $k_{th}$  ant;
23:  end for
24:  /* Update the pheromone of links */
25:  for  $i = 1$  to  $n$  do
26:    for  $j = 1$  to  $n$  do
27:       $\tau_{ij}^{new} \leftarrow \rho \cdot \tau_{ij}^{old} + \Delta\tau(i, j)$ ;
28:       $\Delta\tau(i, j) \leftarrow \sum_{k=1}^m \Delta\tau_k(i, j)$ ;
29:    end for
30:  end for
31:  /* Update the pheromone of nodes */
32:  for  $i = 1$  to  $n$  do
33:    Update the pheromone of nodes;
34:  end for
35: end while
36:  $Score \leftarrow \tau + \varepsilon \cdot \eta$ ;
37: Return Score;

```

The Algorithm. The proposed algorithm is outline in Algorithm 1, where the major steps are explained as follows:

- Step 1 (Line 1–2): In the network G of n nodes, m ants are used. For node j , its status in the t -th iteration is $p_j^t = \begin{bmatrix} \alpha_1^t & \alpha_2^t & \dots & \alpha_m^t \\ \beta_1^t & \beta_2^t & \dots & \beta_m^t \end{bmatrix}$, $t = 1, 2, \dots, N_c$ (the number of iterations). In the first iteration, $\alpha_i = \beta_i = \frac{1}{\sqrt{2}}$ for all i ($i = 1, 2, \dots, m$). The initial settings of the pheromone matrix τ and the visibility matrix η follow Eq. (2) and Eq. (3), respectively. In the experiments, we set $\delta, \lambda, \kappa, \nu, \rho$ and ε to 1, 1, 2, 1, 0.9 and 0.2, respectively. ι is determined by the topology of networks, which is set to 0 or 1.
- Step 2 (Line 4–7): The m ants are randomly assigned to m nodes after shuffling the order of nodes. For example, to assign 3 ants to 3 of 5 nodes. We first shuffle the order of 5 nodes, say the result is 3, 1, 5, 2, 4. Then the 3 ants are assigned to the nodes of 3, 1 and 5, respectively.
- Step 3 (Line 8–19): The transition probability of ant k from v_i to v_j , $p_{ij}^k = \frac{p_{ij}^k}{\sum_i p_{ii}^k}$. In each iteration, m ants simultaneously walk for n steps without interaction.
- Step 4 (Line 20–34): After each iteration, the pheromone matrix τ of links will be updated according to Eq. (6). Similarly, the pheromone of nodes will be updated too, according to Eq. (1).
- Step 5 (Line 36–37): Let $t = t + 1$. If $t < N_c$, then go to Step 2. Otherwise, output the matrix $\tau + \varepsilon \cdot \eta$, which will be used to calculate the precision.

- **Computational Complexity:** Let n be the number of nodes in the network, and $\langle k \rangle$ be the average degree of nodes. Firstly, in order to calculate the visibility matrix η , all the neighbours of the second-order neighbours of each node are required to loop over. Therefore, the time complexity of calculating visibility matrix η is $O(n\langle k \rangle^3)$. Secondly, the number of qubits on each node is the same as the number of ants m . Therefore, the time complexity of initializing quantum pheromone μ is $O(nm)$. Thirdly, the paths are the traces that the ants traverse in the network. The time complexity for forming a path in each iteration is $O(n^2)$. Then the time complexity for recording m paths in N_c iterations is $O(N_c mn^2)$. In conclusion, the overall time complexity of the QACO algorithm can be estimated as $O(n\langle k \rangle^3) + O(nm) + O(N_c mn^2)$. Since $\langle k \rangle$, m and N_c are constants, the time complexity of the algorithm is $O(n^2)$.

Experimental results and analysis. In our experiments, we provide the performance comparison of the proposed algorithm against four state-of-the-art algorithms mentioned in Section Methods, including the algorithms based on the indices of SPM, SBM, FBM and CH. Among them, the SPM, SBM and FBM algorithms are three global algorithms for topological link prediction, while the CH algorithm is a local algorithm for topological link prediction. The recent studies have confirmed that the SPM and CH algorithms are actually one of the best-performing state-of-the-art global and local algorithms, respectively²⁰. In order to verify the robustness of our algorithm, many different types of networks (the artificial, small-size, large-size and time-evolving networks) introduced in Table SI of the Supplementary Information (SI) are used. In addition, considering the high time complexity of the SBM and FBM algorithms, we only compare the SPM and CH algorithms on the large-size and time-evolving networks. Their performance will be shown as the following.

Evaluation on artificial networks. Figures 1 and 2 show the precision of the tested algorithms on the nPSO and WS networks, respectively. Figure 1 shows the average precision on the nPSO networks with 8 communities. We select the parameters (r , m , T and N) of the nPSO model, based on the topological features of the small-size real networks mentioned above. In Fig. 1, one can observe that the algorithms achieve fluctuating performances for low temperature (high clustering), relatively stable performances for medium temperature (medium clustering) and pretty stable performance for high temperature (low clustering). Furthermore, the performance of these algorithms generally decays with temperature. In general, the CH algorithm outperforms the other algorithms for most of the parameter domain. Meanwhile, as the network size N expands, the advantage of CH algorithm becomes more pronounced. Whereas, our algorithm is ranked the second. This result also confirms the robustness of our algorithm.

Figure 2 shows the average precision on the Watts-Strogatz networks. The values of N and m are the same as what they are in the nPSO networks. The values of the parameter β are adjusted to build a network sharing a similar clustering coefficient and average shortest path length with the nPSO networks. In Fig. 2, one can observe that the performance of the SPM algorithm is ranked first, while that of our algorithm is ranked second again. Generally speaking, the robustness of our algorithm is better than that of the SPM and CH algorithms on both artificial networks.

Evaluation on small-size complex networks. In order to test the robustness of our algorithm, 20 small-size network datasets are collected, which covers social networks, biological networks and technology networks, etc.

Table 1 shows the precision of the tested algorithms on the 20 small-size networks. One can observe that the SPM algorithm achieves the highest performance in 12 out of 20 networks, tying for the first place in two of them, whereas our algorithm receives the best result in four networks, tying for the first place in one of them. In Table 1, the mean precision of our algorithm over the tested datasets is tied with the SBM algorithm for second place, just following the SPM algorithm. The last row in Table 1 also shows the mean ranking of the tested algorithms on the networks. The precision-ranking metric is a more robust and reliable metric for assessing the overall performance. The mean ranking of the algorithms over all the networks represents the final evaluation score. One can observe that the SPM algorithm is still the best performing approach, as already deducible from mean precision, with an average ranking of 2.15. Our algorithm is ranked second with 2.78. The third approach is CH with 3.20. The SBM and FBM algorithms are ranked the fourth and fifth, respectively (The result of ranking score for each network can refer to Table SII in the SI). In short, compared with these state-of-the-art algorithms, the performance of our algorithm basically exceeds the CH, SBM and FBM algorithm, while lags behind the global algorithm, SPM. In addition, the performance of the CH algorithm (a local approach) exceeds that of the SBM and FBM algorithm (two global algorithms), which demonstrates its capability of prediction.

In order to check the statistical significance of the difference in performance between the algorithms, we perform the pairwise permutation tests (10,000 iterations), based on the precision-ranking values of each algorithm over the networks (columns of Table SII in the SI). Table SIII in the SI presents the pairwise p-values, adjusted for multiple hypothesis comparison by the Benjamini-Hochberg correction. The p-values lower than the significance level 0.05 are highlighted in bold. We find that the mean performances of the SPM-CH, SPM-SBM, SPM-FBM and QACO-FBM pairs are significantly different. In the sense, the result confirms the advantage of the SPM in performance, on the other hand, confirms that our algorithm is competitive in respect to the four state of the art algorithms. Note that the SPM is a global algorithm with time complexity, N^3 , while our algorithm is a quasi-local algorithm with time complexity, N^2 .

Comparing the performance of our algorithm in a variety of networks, we find that our algorithm is suitable for the networks with disassortative mixing³⁶. For example, for the four networks, mouse neural, ACM2009 contacts, physicians innovation and haggle contacts networks, our algorithm is all ranked first in terms of precision-ranking. Their Pearson coefficients are -0.52 , -0.12 , -0.08 and -0.47 , respectively. Conversely, when the Pearson coefficient is positive, our algorithm typically performs poorly. For example, the precision-ranking

| | <i>SPM</i> | <i>QACO</i> | <i>CH</i> | <i>SBM</i> | <i>FBM</i> |
|------------------------|-------------|-------------|-------------|-------------|-------------|
| mouse neural | 0.02 | 0.15 | 0.11 | 0.10 | 0.01 |
| karate | 0.17 | 0.23 | 0.20 | 0.28 | 0.27 |
| dolphins | 0.13 | 0.18 | 0.14 | 0.16 | 0.19 |
| macaque neural | 0.72 | 0.64 | 0.56 | 0.68 | 0.55 |
| polbooks | 0.17 | 0.17 | 0.17 | 0.15 | 0.18 |
| ACM2009 contacts | 0.26 | 0.28 | 0.27 | 0.25 | 0.26 |
| football | 0.31 | 0.30 | 0.36 | 0.34 | 0.25 |
| physicians innovation | 0.07 | 0.09 | 0.07 | 0.06 | 0.08 |
| FWFW | 0.56 | 0.30 | 0.08 | 0.18 | 0.14 |
| manufacturing email | 0.51 | 0.41 | 0.42 | 0.47 | 0.39 |
| littlerock foodweb | 0.84 | 0.44 | 0.15 | 0.73 | 0.17 |
| jazz | 0.65 | 0.48 | 0.56 | 0.47 | 0.45 |
| residence hall friends | 0.28 | 0.21 | 0.24 | 0.18 | 0.24 |
| haggle contacts | 0.62 | 0.62 | 0.57 | 0.62 | 0.57 |
| worm nervoussys | 0.16 | 0.13 | 0.12 | 0.15 | 0.11 |
| netsci | 0.41 | 0.37 | 0.50 | 0.13 | 0.33 |
| infectious contacts | 0.37 | 0.30 | 0.34 | 0.30 | 0.33 |
| flightmap | 0.75 | 0.59 | 0.54 | 0.64 | 0.56 |
| email | 0.16 | 0.15 | 0.16 | 0.09 | 0.16 |
| polblog | 0.23 | 0.20 | 0.17 | 0.19 | 0.17 |
| mean precision | 0.37 | 0.31 | 0.29 | 0.31 | 0.27 |
| mean ranking | 2.15 | 2.78 | 3.20 | 3.28 | 3.60 |

Table 1. The precision evaluation of five algorithms with 90% of the links used as the training set on the small-size real networks. For each network, the table reports the mean precision over the random iterations and the mean precision over the entire datasets. Moreover, the mean ranking of the algorithms over all the networks is shown in the last row. In addition to 10 iterations for SBM due to its high computational time, the other algorithms are 100 iterations. For each network, the best algorithm (or algorithms) is highlighted in bold. The networks are sorted by N in ascending order. The algorithms are ranked from left to right according to the mean ranking (the results of ranking score for each network can refer to Table SII in the SI).

| | <i>CH</i> | <i>QACO</i> | <i>SPM</i> |
|----------------|-------------|-------------|-------------|
| yeast | 0.25 | 0.26 | 0.44 |
| odlis | 0.12 | 0.11 | 0.08 |
| router | 0.11 | 0.33 | 0.30 |
| advogato | 0.16 | 0.15 | 0.15 |
| wikipedia | 0.14 | 0.11 | 0.16 |
| oregon | 0.08 | 0.08 | 0.07 |
| P2P | 0.03 | 0.04 | 0.03 |
| arxiv astroph | 0.53 | 0.58 | 0.67 |
| thesaurus | 0.08 | 0.08 | 0.07 |
| arxiv hepht | 0.22 | 0.21 | 0.27 |
| ARK201012 | 0.16 | 0.14 | 0.11 |
| facebook | 0.11 | 0.10 | 0.10 |
| mean precision | 0.17 | 0.18 | 0.20 |
| mean ranking | 1.88 | 2.00 | 2.13 |

Table 2. The precision evaluation of three algorithms with 90% of the links used as the training set on the large-size real networks. For each network, the table reports the mean precision over the random iterations and the mean precision over the entire datasets. Moreover, the mean ranking of the algorithms over all the networks is shown in the last row. In addition, all algorithms are 10 iterations. For each network, the best algorithm (or algorithms) is highlighted in bold. The networks are sorted by N in ascending order. The algorithms are ranked from left to right according to the mean ranking (the results of ranking score for each network can refer to Table SIV in the SI).

values of our algorithm are 4, 3, 4, 4.5 and 4 on the football, jazz, residence hall friends, infectious contacts and email networks, respectively. All of the networks are assortative mixing (The details can refer to Table SI and Table SII in the SI.).

| QACO | | | | | CH | | | | | SPM | | | | | | mean precision | mean ranking |
|-------------|-------------|-------------|-------------|-------------|------|------|------|------|------|------|------|------|------|------|-------------|----------------|--------------|
| 0.13 | 0.14 | 0.15 | 0.16 | 0.16 | 0.11 | 0.12 | 0.13 | 0.14 | 0.14 | 0.08 | 0.09 | 0.09 | 0.10 | 0.11 | QACO | 0.15 | 1 |
| | 0.13 | 0.14 | 0.16 | 0.16 | | 0.12 | 0.13 | 0.14 | 0.14 | | 0.07 | 0.08 | 0.09 | 0.10 | CH | 0.13 | 2 |
| | | 0.13 | 0.15 | 0.16 | | | 0.12 | 0.13 | 0.14 | | | 0.08 | 0.09 | 0.10 | SPM | 0.09 | 3 |
| | | | 0.13 | 0.15 | | | | 0.12 | 0.13 | | | | 0.08 | 0.09 | | | |
| | | | | 0.14 | | | | | 0.12 | | | | | 0.09 | | | |

Table 3. The precision evaluation of three algorithms in time on the AS Internet networks. From September 2009 to December 2010, six AS Internet network snapshots are considered at time steps of 3 months. For every snapshot at times $i = [1, 5]$, the non-observed links are assigned likelihood scores based on the algorithms. Meanwhile, the link-prediction performance is evaluated with respect to every future time point $j = [i + 1, 6]$. Considering a pair of time points (i, j) , the non-observed links at time i are ranked by likelihood scores in descending order. And the precision is computed as the percentage of links that appear at time j among the $top - r$ links, where $top - r$ is the total number of non-observed links at time i that appear at time j . Non-observed links at time i involving nodes that disappear at time j are not considered in the ranking. For each algorithm, a 5-dimensional upper triangular matrix is shown, where entry (i, j) denotes the precision of the link prediction from time i to time $j + 1$. On the right side, the algorithms are ranked by the mean precision computed over all the time combinations. For each comparison, the best algorithm is highlighted in bold.

Evaluation on large-size real complex networks. Aiming to test the scalability of our algorithm, 12 large-size network datasets of different type are selected, which covers social networks, internet networks and technology networks, etc. Their topological features of common interest are likewise shown in Table SI. Table 2 shows the precision of the tested algorithms on the networks. It is evident that the CH algorithm obtains the highest performance in 6 out of 12 networks, tying the first position in two of them. The overall performance can be measured by the mean ranking. The mean ranking of the CH algorithm is 1.88, which is higher than its counterparts. With respect to the value of precision, the CH algorithm performs very poorly on the router network, which drags its mean precision. This result may be induced by the sparsity of the router network (its average clustering coefficient is 0.01), which restricts the formation of local community and ultimately leads to poor performance of CH algorithm on this network. Instead, the SPM algorithm performs very well on the yeast and arxiv astroph network. The remarkable advance leads to its mean precision is ranked first. However, its mean ranking is at the bottom, since its performance on the other networks is not desirable. The performance of the QACO algorithm is ranked second in both rankings.

We also perform the pairwise permutation tests (10,000 iterations), based on the precision-ranking values of each algorithm over the large-size real networks (columns of Table SIV). From Table SV, one can find that the mean performances of the three algorithms are not significantly different on these networks. Relatively speaking, the CH algorithm performs better on most of the large-size real networks.

Evaluation on time-evolving real networks. In order to maintain the diversity of evaluation framework, we adopt the evaluation framework mentioned in previous study²⁰, which considers the link-growth evolution of a real network over time. Six Autonomous systems (AS) Internet topologies collected by CAIDA are selected, which is from September 2009 to December 2010, spanning 3 months in total. Their topological features of common concern are shown in Table SI.

In Table 3, one can observe that the QACO algorithm outperforms the CH and SPM algorithm, with a mean precision of 0.15. One can see that the precision grows with time, going from 0.13 to 0.16 for the QACO, from 0.11 to 0.14 for the CH and from 0.07 to 0.11 for the SPM, respectively.

In conclusion, different from the removal and re-prediction framework in which the set of missing links is artificially generated by a random procedure, here the set of links that will appear after two consecutive time points is given by ground-truth information, which makes the result even more truthful and significant, confirming the effectiveness of the QACO algorithm.

In summary, our algorithm is competitive in respect to most of the state-of-the-art algorithms in balancing precision and robustness on the tested networks.

Discussions

Inspired by ant colony optimization and quantum computing, we propose a quantum-inspired ant colony optimization algorithm for link prediction in networks. By utilizing visibility, the algorithm integrates the quasi-local structural information of individuals. By using quantum bits and quantum logic gates, it can effectively keep the optimization process from being trapped in local optimums. Compared with a series of the state-of-the-art algorithms on the artificial, small-size, large-size and time-evolving networks, our algorithm exhibits a satisfying robustness and scalability. Especially on the time-evolving real networks, our algorithm outperforms all the tested algorithms. We believe that the quantum-inspired ant colony optimization algorithm provides a new paradigm for the future studies of link prediction.

Admittedly, there may be other definitions of pheromone and visibility that can improve the performance of our algorithm. The pheromone updating strategy and the parameter selecting procedure can also be further optimized. Apart from the ant colony optimization algorithm, other intelligent optimization algorithms may be more effective on some networks. All these will be explored in our future work.

Methods

Link prediction problem. Consider an undirected network $G(V, E)$ where V is a set of nodes and E is a set of links. Here, self-connections and multiple links are not considered. For each pair of disconnected nodes $x, y \in V$ in the network, a link prediction algorithm assigns a score S_{xy} , which indicates the probability of x and y connecting with each other. By sorting the scores of all disconnected node pairs, those pairs at the top of the list are more likely to be connected.

In order to test the performance of the algorithm, the existing links in the network, E , are randomly divided into two sets: the training set E^T and the probe set E^P . Here, $E = E^T \cup E^P$ and $E^T \cap E^P = \emptyset$. The algorithm estimates the scores of disconnected node pairs in G based on the information of the training set, and E^P is used as the benchmark for evaluating the prediction result.

Evaluation metrics. In order to measure the accuracy of link prediction, we use the following metrics: Precision and Precision-ranking^{37–40}.

- Precision refers to the fraction of correctly predicted links in the predicted links. It is defined as:

$$Precision = \frac{m}{L}, \quad (7)$$

where m denotes the number of correctly predicted links, L denotes the number of predicted links. For a given L , the greater precision is, the better the performance of an algorithm is.

- Precision-ranking is a more robust and reliable metric for assessing the overall performance. For each network, all the algorithms are ranked by precision in descending order. The mean ranking of the algorithms over all the networks represents the final score.

Benchmark algorithms. For a comprehensive comparison, three global algorithms (SPM, SBM and FBM) and a local algorithm (CH) are considered.

Let x and y be two randomly selected nodes in a network. $\Gamma(X)$ and $\Gamma(y)$ denote the sets of x and y 's neighbors, respectively. In the following, we will briefly introduce the definitions of the algorithms mentioned above.

The SPM algorithm is a structural perturbation method that relies on a theory similar to the first-order perturbation in quantum mechanics. The idea behind this algorithm is that a missing part of the network is predictable if it does not significantly change the structural features of the observable part^{26,41}. It is thus defined as:

$$S_{xy}^{SPM} = \sum_{k=1}^N (\lambda_k + \Delta\lambda_k) x_k x_k^T, \quad (8)$$

where λ_k , x_k and $\Delta\lambda_k$ are the eigenvalue of the observed matrix, the corresponding orthogonal normalized eigenvector and the eigenvalue of a perturbation set, respectively.

The SBM algorithm is based on the assumption that the probability that two nodes are connected depends only on the groups to which they belong, and it is one of the most general network models²³.

The FBM algorithm is a global algorithm based on the same network partitioning theory as the SBM algorithm, but it introduces a greedy strategy for an efficient sampling over the space of the possible partitions, which leads to high improvements in the computational time²⁵.

The CH index is based on the assumption that two nodes are more likely to be connected if their common neighbours are densely connected^{20,42}. It is thus defined as:

$$S_{xy}^{CH} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{|\gamma(z)|}{|\Gamma(z)|}, \quad (9)$$

where $\gamma(z)$ refers to the sub-set of z 's neighbours which are also common neighbours of x and y . $|\gamma(z)|$ is defined as the local community degree of z .

Data sets. In order to validate the proposed algorithm, we provide the performance comparison over two groups of network datasets. The first group composes of two synthetic networks that are Watts-Strogatz (WS)'s small-world network⁴³ and Cannistraci's nonuniform popularity-similarity optimization (nPSO) network⁴⁴. The WS small-world network is generated by rewiring $r * |E|$ links on a regular lattice with n nodes, where r denotes the randomized rewiring probability and $|E|$ denotes the total number of links^{45,46}. The nPSO model generates synthetic networks in the hyperbolic space where heterogeneous angular node attractiveness is forced by sampling the angular coordinates from a tailored nonuniform probability distribution, and the nPSO model allows to explicitly control the size, the mixing property and the number of communities of the generated network⁴⁷.

The second group composes of three types of real-world networks: 20 small-size, 12 large-size and 6 time-evolving real networks, which are described in detail in the SI. In addition, all real networks have been transformed into undirected, unweighted and no self-loops. Moreover, we consider only the largest component of each of the all real-world networks. The basic topological properties of the largest component in each tested network are shown in Table SI.

References

- Li, F. H. *et al.* Node-coupling clustering approaches for link prediction. *Knowledge-Based Systems* **89**, 669–680 (2015).
- Tan, S. Y., Wu, J. & Lü, L. Y. Efficient network disintegration under incomplete information: the comic effect of link prediction. *Sci. Rep.* **6**, 22916 (2016).
- Zhang, Q. M., Shang, M. S. & Lü, L. Y. Similarity-based classification in partially labeled networks. *International Journal of Modern Physics C* **21**, 813–824 (2010).
- Vinayagam, A., Stelzl, U. & Foulle, R. A directed protein interaction network for investigating intracellular signal transduction. *Science Signaling* **4**, rs8 (2011).
- Zhang, Y., Zheng, Z. & Lyu, M. R. An online performance prediction framework for service-oriented systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **44**, 1169–1181 (2014).
- Aiello, L. M. *et al.* Friendship prediction and homophily in social media. *ACM Transactions on the Web* **6**, 1–33 (2012).
- Zhang, Z. K., Zhou, T. & Zhang, Y. C. Tag-aware recommender systems: a state-of-the-art survey. *Journal of Computer Science and Technology* **26**, 767–777 (2011).
- Maslov, S. & Sneppen, K. Specificity and stability in topology of protein networks. *Science* **296**, 910–913 (2002).
- Yu, H. Y. *et al.* High-quality binary protein interaction map of the yeast interactome network. *Science* **322**, 104–110 (2008).
- Jeong, H., Mason, S. P. & Barabási, A. L. Lethality and centrality in protein networks. *Nature* **411**, 41–42 (2001).
- Bu, D. B., Zhao, Y. & Cai, L. Topological structure analysis of the protein-protein interaction network in budding yeast. *Nucleic Acids Research* **31**, 2443–2450 (2003).
- Stumpf, M. P. H. *et al.* Estimating the size of the human interactome. *Proceedings of the National Academy of Sciences of the United States of America* **105**, 6959–6964 (2008).
- Cannistraci, C. V., Alanis-Lobato, G. & Ravasi, T. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Sci. Rep.* **3**, 1613 (2013).
- Cannistraci, C. V., Alanis-Lobato, G. & Ravasi, T. Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. *Bioinformatics* **29**, 199–209 (2013).
- Knocke, D. Emerging Trends in Social Network Analysis of Terrorism and Counterterrorism. *John Wiley & Sons* (2015).
- Li, F. *et al.* A Clustering-based Link Prediction Method in Social Networks. *Procedia Computer Science* **29**, 432–442 (2014).
- Tan, F., Xia, Y. X. & Zhu, B. Y. Link Prediction in Complex Networks: A Mutual Information Perspective. *PLoS One* **9**, e107056 (2014).
- Lü, L. Y. & Zhou, T. Link Prediction. *Beijing: Higher education press* (2013).
- Zhou, T., Lü, L. Y. & Zhang, Y. C. Predicting missing links via local information. *Eur Phys J B* **71**, 623–630 (2009).
- Muscoloni, A. & Cannistraci, C. V. Local-ring network automata and the impact of hyperbolic geometry in complex network link-prediction. *arXiv preprint arXiv:1707.09496* (2017).
- Daminelli, S., Thomas, J. M., Durán, C. & Cannistraci, C. V. Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. *New Journal of Physics* **17**, 113037 (2015).
- Liu, W. P. & Lü, L. Y. Link prediction based on local random walk. *Europhysics Letters* **89**, 58007 (2010).
- Airolidi, E. M. *et al.* Mixed-membership stochastic block models. *Journal of Machine Learning Research Jmlr* **9**, 1981–2014 (2007).
- Pan, L. M., Zhou, T. & Lü, L. Y. Predicting missing links and identifying spurious links via likelihood analysis. *Sci. Rep.* **6**, 22955 (2016).
- Liu, Z. *et al.* Correlations between Community Structure and Link Formation in Complex Networks. *PLoS One* **8**, e72908 (2013).
- Lü, L. Y., Pan, L. M. & Zhou, T. Toward link predictability of complex networks. *PNAS* **112**, 2325–2330 (2015).
- Lü, L. Y. & Zhou, T. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications* **390**, 1150–1170 (2011).
- He, X. F. & Ma, L. A quantum-inspired ant colony algorithm for graph coloring problem. *Operations Research Transactions* **17**, 19–26 (2013).
- Yang, J., Xu, Q. & Zhang, J. R. A novel quantum ant colony optimizing algorithm. *Acta Scientiarum Naturalium Universitatis Sunyatseni* **48**, 22–27 (2009).
- Chen, W. N. & Zhang, J. An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem with Various QoS Requirements. *IEEE Transactions on Systems, Man, and Cybernetics, Part C* **39**, 29–43 (2009).
- He, X. F. & Ma, L. Quantum-inspired ant algorithm for solving 0–1 knapsack problem. *Computer Engineering and Applications* **47**, 29–31 (2011).
- Olawale, T. & Alan, C. Quantum annealing of the graph coloring problem. *Discrete Optimization* **8**, 376–384 (2011).
- Chen, X. F., Xia, X. Y. & Yu, R. Y. Improved quantum ant colony algorithm based on bloch coordinates. *Journal of Computers* **8**, 1536–1543 (2013).
- Lü, L. Y., Jin, C. H. & Zhou, T. Similarity index based on local paths for link prediction of complex networks. *Physical Review E* **80**, 046122 (2009).
- Chen, B. L. & Chen, L. A link prediction algorithm based on ant colony optimization. *Applied Intelligence* **41**, 694–708 (2014).
- Newman, M. E. J. Assortative mixing in networks. *Physical review letters* **89**, 208701 (2002).
- Davis, J. & Goadrich, M. The relationship between Precision-Recall and ROC curves. *ICML '06 Proceedings of the 23rd international conference on Machine learning*, 233–240 (2006).
- Yang, Y., Lichtenwalter, R. N. & Chawla, N. V. Evaluating link prediction methods. *Knowledge and Information Systems* **45**, 751–782 (2015).
- Bai, M., Hu, K. & Tang, Y. Link prediction based on a semi-local similarity index. *Chin. Phys. B* **20**, 498–504 (2011).
- Zhu, B. Y. & Xia, Y. X. An information-theoretic model for link prediction in complex networks. *Sci. Rep.* **5**, 13707 (2015).
- Zhao, J. *et al.* Prediction of Links and Weights in Networks by Reliable Routes. *Sci. Rep.* **5**, 12261 (2015).
- Liu, J., Xu, B. M. & Xu, X. A link prediction algorithm based on label propagation. *Journal of Computational Science* **16**, 43–50 (2016).
- Watts, D. J. & Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature* **393**, 440–442 (1998).
- Muscoloni, A. & Cannistraci, C. V. A nonuniform popularity-similarity optimization (nPSO) model to efficiently generate realistic complex networks with communities. *New Journal of Physics* **20**, 052002 (2018).
- Ahn, M. W. & Jung, W. S. Accuracy test for link prediction in terms of similarity index: The case of WS and BA models. *Physica A* **429**, 177–183 (2015).
- Zhu, B. Y., Xia, Y. X. & Zhang, X. J. Weight prediction in complex networks based on neighbor set. *Sci. Rep.* **6**, 38080 (2016).
- Muscoloni, A. & Cannistraci, C. V. Leveraging the nonuniform PSO network model as a benchmark for performance evaluation in community detection and link prediction. *New Journal of Physics* **20**, 063022 (2018).

Acknowledgements

The authors sincerely thank the anonymous reviewers for their valuable comments that have led to the present improved version of the original manuscript. Y. Z. was supported by the National Natural Science Foundation of China (No. 61503285), Municipal Natural Science Foundation of Shanghai (No. 17ZR1446000). J. G. was supported by the National Natural Science Foundation of China (No. 61772367 and 61373036), Program of

Shanghai Subject Chief Scientist (15XD1503600), and Program of Shanghai Science and Technology Committee (16511105200). S. Z. was supported by the Key Projects of Fundamental Research Program of Shanghai Municipal Commission of Science and Technology under Grant No. 14JC1400300 and the Program of Science and Technology Innovation Action of Science and Technology Commission of Shanghai Municipality (STCSM) (No. 17511105204).

Author Contributions

All authors designed the experiments together. Z.C. implement the experiments and prepared all the figures. Z.C. and Y.Z. wrote the main manuscript text together. J.G. and S.Z. reviewed the manuscript.

Additional Information

Supplementary information accompanies this paper at <https://doi.org/10.1038/s41598-018-31254-3>.

Competing Interests: The authors declare no competing interests.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2018