CHAOS **28**, 121104 (2018)

Optimal disintegration strategy in multiplex networks

Mingze Qi, Ye Deng,[a] Hongzhong Deng, and Jun Wu[b]
College of Systems Engineering, National University of Defense Technology, Changsha, Hunan 410073, People's Republic of China

(Received 25 October 2018; accepted 28 November 2018; published online 26 December 2018)

Network disintegration comprises the problem of identifying the critical nodes or edges whose removal will lead to a network collapse. The solution of this problem is significant for strategies for dismantling terrorist organizations and for immunization in disease spreading. Network disintegration has received considerable attention in isolated networks. Here, we consider the generalization of optimal disintegration strategy problems to multiplex networks and propose a disintegration strategy based on tabu search. Experiments show that the disintegration effect of our strategy is clearly superior to those of typical disintegration strategies. Moreover, our approach sheds light on the properties of the nodes within the optimal disintegration strategies. *Published by AIP Publishing.* https://doi.org/10.1063/1.5078449

Many systems can be represented by networks of nodes connected by edges. In particular, many essential natural, technological, and social systems consist of fully or partially interdependent subsystems, allowing them to be represented by such interdependent or multiplex networks. In addition to enhancing the reliability of some crucial networks, it is also desirable to dismantle networks that are harmful to us, such as disease spreading networks and terrorist organizations. The essential purpose of this work is to seek an optimal disintegration strategy in multiplex networks.

I. INTRODUCTION

Networks are ubiquitous.[1,2] These may consist of tangible objects in the Euclidean space, such as certain infrastructure networks,[3] or entities defined in an abstract space, such as in social networks.[4] The majority of these networks are beneficial to people, and many studies concentrate on how to enhance their robustness.[5–7] However, in other situations, we would like to disintegrate networks that are harmful. Examples include epidemic spreading networks,[8,9] terrorist networks,[10,11] and rumor spreading networks.[12,13] Hence, research regarding disintegration strategies for networks has received increasing attention.[14,15]

In previous research, disintegration strategies for complex networks have focused on certain properties of the nodes. Most traditional strategies are degree-based strategies, in which nodes are sequentially removed in decreasing order of their degrees.[16] On the other hand, some global metrics have also been employed to identify the vital nodes for network disintegration, including coreness,[17] subgraph centrality,[18] and others.[19] Recently, some algorithms have been presented in various contexts, such as network dismantling,[20,21] targets,[22] immunization,[23] and percolation.[24,25] The solutions for these problems are equivalent, with the aim of identifying the minimal set of removals required to destroy a giant connected component in a single network.[26] These algorithms can also be employed to obtain efficient disintegration strategies under limited conditions.

There have been some previous studies on disintegration strategies in single-layer networks,[14,27] which do not interact with other networks. However, real-world networks rarely exist independently, and most are coupled with or interact with other networks, such as the same actors appearing in different social networks,[28] multi-modal transportation systems sharing common geographical locations,[29] and diverse types of connections among proteins.[30,31] A better description of such systems is in terms of multiplex networks, where each layer typically contains the same type of nodes but different links among them.[32,33] Recent research has demonstrated that ignoring the co-existence of various interactions in the study of a multiplex network may have dramatic consequences in the ability and properties of the system.[34,35] Therefore, studying the problem of disintegration in multiplex networks is of considerable significance.

There have also been many studies in recent years concerning attacks or robustness in multiplex networks.[36,37] In particular, Osat *et al.*[38] considered various existing algorithms for optimal percolation in multiplex networks, finding that ignoring the co-existence of interactions may lead to overestimating the robustness of a system. However, there have been few studies concentrating on the problem of disintegration strategies in multiplex networks, especially with different sizes of node sets removed.

In this paper, we focus on an optimal disintegration strategy in multiplex networks. We construct a general model for finding the optimal disintegration strategy. Then, we solve this using a tabu search (TS) algorithm, which is a flexible and efficient global search algorithm, by simulating the process of human memory.[39] Moreover, we test the algorithm on both artificial networks and real datasets, and introduce some existing algorithms for comparison. The performance of the optimal strategy and the properties of nodes in different optimal strategies are revealed.

[a]Electronic mail: dengye13a@nudt.edu.cn
[b]Electronic mail: junwu@nudt.edu.cn

The remainder of this paper is organized as follows. The optimization model for multiplex network disintegration problems is presented in Sec. II. In Sec. III, we introduce a solution based on tabu search. Experiments on three types of model network and real-world networks are presented in Sec. IV. Finally, we present conclusion and discussions in Sec. V.

II. OPTIMIZATION MODEL FOR DISINTEGRATION STRATEGY IN A MULTIPLEX NETWORK

A multiplex network can be thought of as a set of nodes connected by various types of edges, and we depict an example of a two-layer network in Fig. 1. However, a multiplex network is not simply a stack of many network layers but incorporates functional coupling between layers.[40] For instance, the different layers in a multiplex network may be coupled either in a dependent manner, as in the critical infrastructure,[41] or in a connective manner, as in a transportation system.[30] Depending on the type of multiplexity, the same multiplex structures can behave quite differently. As shown in Fig. 1(a), we employ two disintegration strategies to attack node 4 or node 6. From the perspective of a single layer, each of these has four edges of different types, and they do not affect the connectivity of other nodes. However, from the perspective of a multi-layer network, their performances are quite different in two different coupling modes [the dependent manner in (c) and (e) and connective manner in (d) and (f)].

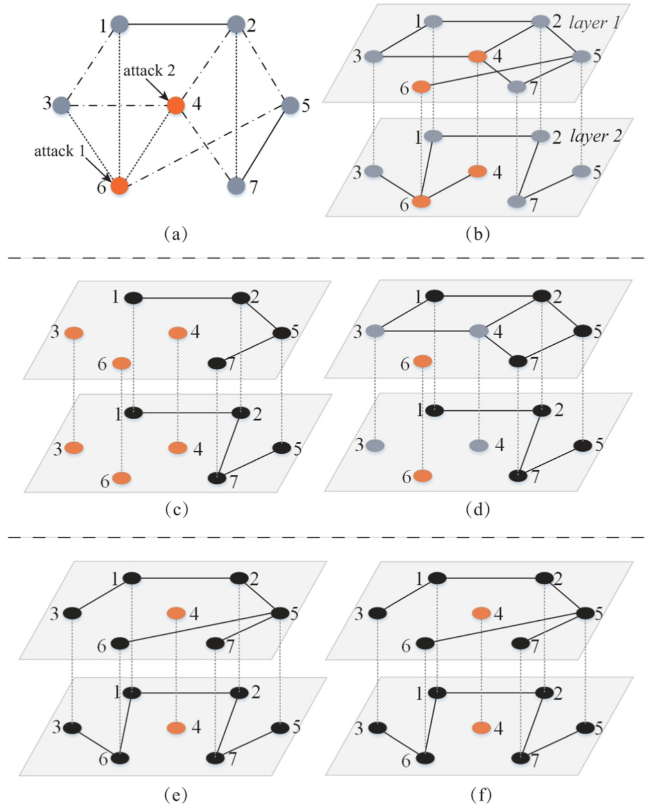

FIG. 1. Schematic illustration of the disintegration strategy in multiplex networks. A multiplex network with two different types of edges is shown in (a) and (b). In (a), the dashed-dotted line represents the edges in layer 1, and the dotted line represents those in layer 2. The edges that appear twice are shown as solid lines. We separate the two layers in (b). In (c) and (d), we show the disintegration effect of attack 1 in two different coupling modes (dependent and connective, respectively). In the same manner, the effect of attack 2 is shown in (e) and (f). In each scheme, we use orange to mark the nodes that are attacked or failed and black to denote the nodes belonging to the GMCC. From the perspective of multiplex networks, the effects of different attacks may be entirely different, owing to the varying properties of edges between layers or within layers.

A complex network (single-layer network) can be formed as a tuple $G = (V, E)$, where $V(G) = V$ is the set of nodes and $E(G) = E \subseteq V \times V$ is the set of edges. By $A(G) = (a_{ij})_{N\times N}$, denote the adjacency matrix of G, in which $a_{ij} = a_{ji} = 1$ if v_i and v_j are connected, and $a_{ij} = 0$ otherwise. We consider a multiplex network with M layers and an adjacency matrix $A^{[\alpha]}$ in each layer $\alpha = 1, 2, \ldots, M$. Without loss of generality, we assume that a multiplex network G composed of N nodes is given. Every node $i \in G$ appears in both layers so that the failure of a node in one layer implies the simultaneous failure of its copy in the other layer. In addition, let $k_i^{[\alpha]}$ be the degree of node v_i in layer α. The value of $k_i^{[\alpha]}$ is the number of adjacent edges of node v_i in layer α.

In this study, the disintegration approach focuses on node removal, and we assume that the attached edges are removed if one node is removed. Denote by $\widehat{V} \subseteq V$ the set of nodes that are removed, and denote by $\widehat{G} = (V - \widehat{V}, \widehat{E})$ the network after these nodes are removed. By $n = \left|\widehat{V}\right|$, denote the disintegration strength parameter. We define the disintegration strategy as $\widehat{X} = [x_1, x_2, \ldots, x_n]$, where $x_i = 0$ if $v_i \in \widehat{V}$ and $x_i = 1$ otherwise. It is easy to obtain that

$$n = N - \sum_{i=1}^{N} x_i. \tag{1}$$

Let Γ be the measure function for the network performance. This could be given by the connectivity or robustness indices of the multiplex network. It is worth noting that the measure function must be monotonous, i.e., $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$, then $\Gamma(G_1) \le \Gamma(G_2)$, where $G_1 = (V_1, E_1)$ is a subgraph of $G_2 = (V_2, E_2)$. The monotonicity guarantees that the measure function declines after node removal. The goal of the optimization model is to identify the optimal solution $\hat{X}^*$ that can maximize the disintegration effect. Therefore, the optimization model for the disintegration strategy in a multiplex network can be described as follows:

$$\max \Phi(X = [x_1, x_2, \ldots, x_n]) \\ \text{s.t.} \begin{cases} n = N - \sum_{i=1}^{N} x_i, \\ x_i = 0 \text{ or } 1, i = 1, 2, \ldots, N. \end{cases} \tag{2}$$

In single-layer networks, connected components are used to measure how well a network is connected. For the coupled networks described above, in this study, we generalize the concept of connected components to mutually connected clusters (MCC), which are defined as follows:[41]

Mutually connected cluster (MCC): A set of nodes $S_A \subseteq A$ with corresponding set $S_B \subseteq B$ is called a mutually connected set if any two nodes in S_A are connected by a path of A-edges in S_A and any two nodes in S_B are connected by a path of B-edges in S_B. Furthermore, a mutually connected set is called a mutually connected cluster (MCC) if it is impossible to add another node to the set to form a mutually connected set.

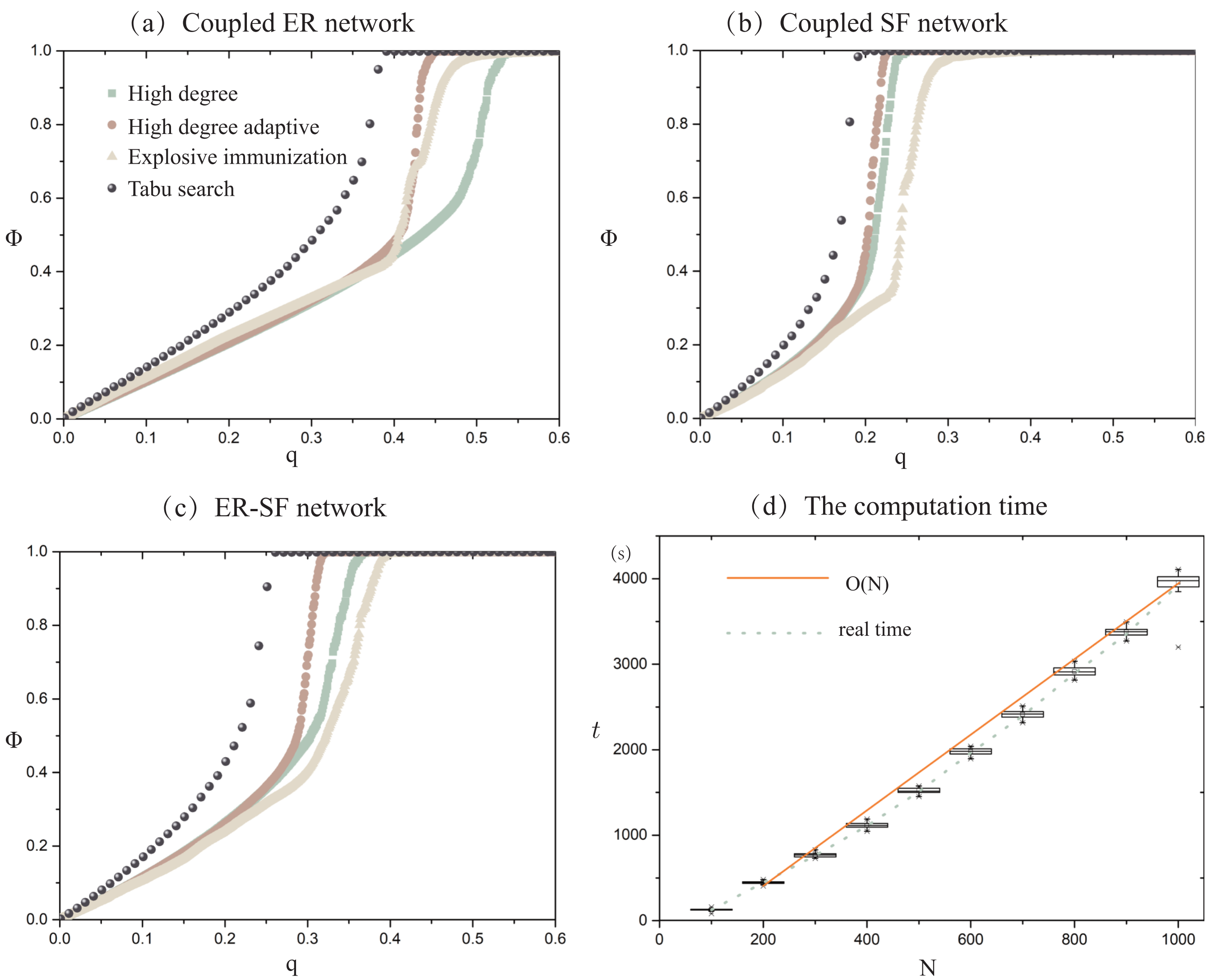

FIG. 2. The optimal disintegration strategy in three synthetic networks. We consider the disintegration effect Φ versus the fraction q of nodes removed in a coupled Erd'os–Rényi (ER) network (a), coupled scale-free (SF) network (b), and ER-SF network (c) and compare the results with various disintegration strategies. The computation time of the tabu search algorithm versus the network size N is shown in (d), where $q = 0.2$. The figures show that the disintegration effects of the proposed disintegration strategies are considerably better than those of the others, and the time complexity is approximately $O(N)$.

In particular, we focus our attention on the largest among these clusters, usually referred to as the giant mutually connected cluster (GMCC). In this study, we use the size of the GMCC, which is given as a fraction μ_∞ of the size of the full network, as the measure function Γ. We assume that each layer of the multiplex network is connected, and so the size of the full network is 1, i.e., $\Phi(X) = 1 - \mu_\infty$.

III. SOLUTION BASED ON TABU SEARCH

Clearly, the objective function Φ yields no explicit form, and so we cannot solve the optimization model in Eq. (2) using traditional integer programing methods. It can instead be viewed as a combinatorial optimization problem. For a network with N nodes, there are C_N^n ways to pick n nodes for removal, which increases sharply with N and n. Therefore, it is almost impossible to look through all possible solutions if the network size is large. Hence, we consider solving this problem using the tabu search (TS) algorithm. The tabu search algorithm has been employed for the optimal disintegration strategies in single-layer networks.[14,42] In this study, the multiplex network G is coded with M adjacency matrixes. The state of the network after the node set removed and the method of calculating the GMCC are determined by the properties of the edges between or within the layers.

The four primary parts of the algorithm are described as follows:

Move mechanism: The move mechanism represents the process of pointing the current solution state toward another one. The basic principle of the move mechanism is to obtain an initial potential solution and check its immediate neighbors to iteratively choose the best one for the next step. In the tabu search algorithm, we consider the operation swap as a move operation. Each neighbor of the current solution is obtained by swapping the states (0 or 1) of two nodes randomly, which is denoted as a swap S. Because the number of possible neighbors is large, we randomly choose n_{can} neighbors as candidate solutions.

Tabu list: The tabu list, denoted as T_{list}, records the previously encountered "*best*" swap. The key observation here is that when the swap operation occurs on nodes that are in the tabu list, we need to abandon this swap to prevent the search from tracing back. The tabu list is initially empty, and the "*best*" swap at each step is recorded in the tabu list once conducted. Each time we swap the two corresponding nodes so that the tabu nodes will not be allowed to change their state within a certain number of iterations. The length of the tabu list L determines the number of iterations for which the best swaps cannot be chosen. A swap in tabu list for more than L iterations can then be released. The length of the tabu list L will affect the time complexity and performance of the algorithm. The larger the L is, the more accurate the results will be, but the corresponding storage and computational complexity will also increase. The most appropriate L can be confirmed by testing in advance according to the performance of the computer. Notably, the tabu list should be emptied if $\hat{X}_{opt}$ is updated to a better one, because a better $\hat{X}_{opt}$ indicates that the algorithm has not fallen into a sub-optimal solution, and the previous good swaps can be accepted.

Aspiration criterion: An aspiration criterion is set to avoid the loss of good solutions and encourage global optimization. An aspiration criterion is a condition that allows a swap to be released from the tabu list when its corresponding solution yields $\Phi(\hat{X}_{can})$ that is better than the current $\Phi(\hat{X}_{opt})$.

Termination criterion: The termination criterion, denoted by n_{max}, is set to stop a round of process. The current number of iterations is denoted by n_{iter} before $\hat{X}_{opt}$ is updated. The value of n_{iter} will be set to 0 once $\hat{X}_{opt}$ is updated. When the current number of iterations n_{iter} reaches the previously set n_{max} without an improved solution, the process ends. This process can be viewed as a round of searching, where the number of iterations is uncertain. We denote the maximum total number of iterations by T_{max} and the total number of iterations in all past rounds by T_{iter}. When a round of iterations is completed before T_{iter} exceeds T_{max}, the program will start a new round of searching from an initial solution.

Each step of the algorithm is described as follows (the source code can be obtained from the authors):

Procedure 1: Initialization of the algorithm. Set the maximum iteration number n_{max}, the maximum total iteration number T_{max}, the number of candidate solutions n_{can}, and the length of the tabu list L. Initialize $T_{iter} = 0$.

Procedure 2: Generate the initial solution $\hat{X}_0$ and initialize the tabu list. Initialize $n_{iter} = 0$ and $T_{list} =$ NULL. Generate a vector $\hat{X}_0 = (x_1, x_2, \ldots, x_n)$. Set $x_i = 0$ when node v_i is removed and $x_i = 1$ otherwise. $\hat{X}_0$ can either be given randomly or by another strategy with a better performance. Determine whether $T_{iter} > 0$. If satisfied, then continue to the next step. If not satisfied, then let the current best solution $\hat{X}_{opt} = \hat{X}_0$. Calculate $\Phi(\hat{X}_{opt})$.

Procedure 3: $T_{iter} = T_{iter} + 1$. Determine whether $T_{iter} > T_{max}$. If satisfied, output the result. Otherwise, continue to the next step.

Procedure 4: Generate candidate solution. Generate n_{can} candidate solutions $\hat{X}_i \in N(X_{cur})$ by the respective swaps S_i. Determine $\hat{X}_k$, where $\hat{X}_k = \max \Phi(\hat{X}_i)$.

Procedure 5: Determine whether $S_K \notin T_{list}$ or $\Phi(\hat{X}_k) > \Phi(\hat{X}_{opt})$ (aspiration criterion). If satisfied, continue to the next step. If not satisfied, find another $\hat{X}_k$ s.t. $\hat{X}_k = \max \Phi(\hat{X}_{opt})$ and $S_k \notin T_{list}$.

Procedure 6: Update the tabu list. Remove the element added into T_{list} L iterations ago, and then add the swap S_k to T_{list}.

Procedure 7: Determine whether $\Phi(\hat{X}_{cur}) > \Phi(\hat{X}_{opt})$. If satisfied, then $\Phi(\hat{X}_{opt}) = \Phi(\hat{X}_{cur})$, $T_{list} =$ NULL, and $n_{iter} = 0$. If not satisfied, then $n_{iter} = n_{iter} + 1$.

Procedure 8: Determine whether $n_{iter} < n_{max}$. If satisfied, return to step 3. If not satisfied, return to step 2.

IV. EXPERIMENTAL ANALYSIS

A. Experiments in synthetic networks

In this section, we test the performance of the algorithm on multiplex networks consisting of a pair of interdependent networks, as discussed by Buldyrev *et al.* in 2010.[41] In the model, we consider two networks A and B with degree distributions $P_A(k)$ and $P_B(k)$ and equal sizes $|A| = |B| = N$. We say that a node i_A in A depends on a node i_B in B if the failure of i_B causes the failure of i_A. In this model, we assume that each node i_A of A depends only on one node i_B of B and vice versa. Hence, the removal of nodes may cause an iterative process of a cascade of failures.

The parameters for solving the optimization model are set as $L = 10$, $T_{max} = 1000$, $n_{max} = 100$, and $n_{can} = 100$. These experiments are conducted on a PC (Intel Core i7-7700U CPU at 3.6 GHz, 16 GB RAM) with MATLAB version R2017b. Here, we focus on three typical networks: a coupled ER network, a coupled SF network, and a ER-SF network, which are described as follows:

Coupled ER network: A multiplex network composed of two layers generated independently according to the Erd'os–Rényi model,[43] with N nodes and an average degree of $\langle k \rangle$. The Erd'os–Rényi model generates a network by connecting nodes randomly, where each edge is included in the network with a probability p, independently from every other edge. The parameter p determines the average degree $\langle k \rangle$ of the network.

Coupled SF network: A multiplex network composed of two layers generated independently according to a scale-free model with N nodes and an average degree $\langle k \rangle$. The scale-free model[44,45] (i.e., Barabasi-Albert preferential attachment network) is created from a few isolated nodes and expanded by adding new nodes and links. The new nodes have a preference to attach to heavily linked nodes (hubs).

ER-SF networks: A multiplex network composed of two layers generated independently, according to the above Erd'os–Rényi and scale–free models, respectively.

To demonstrate the efficiency and effectiveness of our algorithm, we compare it with three typical score-based disintegration strategies: high degree (HD), high degree adaptive (HDA), and explosive immunization (EI).

In a multiplex network, a degree-based attack is the easiest method of dismantling a network. We extend the HD strategy to multiplex networks, and the score of a node is defined as the product of its degrees across all layers, which has been proved to be the best combination.[38] In the HD algorithm, if we recalculate the degrees of the nodes after node

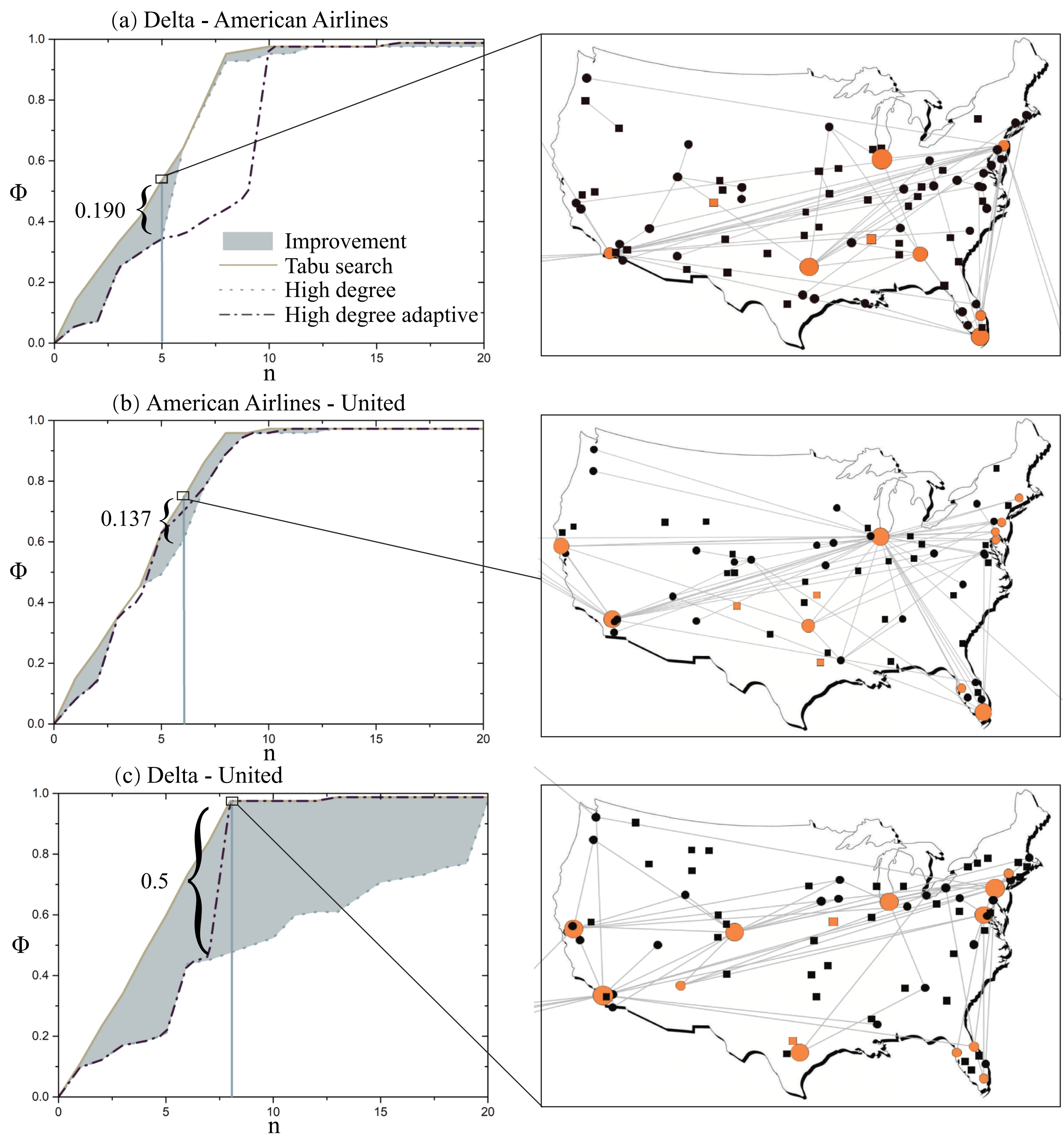

FIG. 3. Optimal disintegration strategy in multiplex transportation networks, which consist of Delta-American Airlines (a), American Airlines-United (b), and Delta-United (c). We show the relative size of the disintegration effect Φ versus the number of deleted nodes with the high degree strategy (dotted lines), high degree adaptive strategy (dashed-dotted lines), and optimal disintegration strategy based on tabu search (solid lines) in the left side of each graph. The filled area illustrates the improvement of the network disintegration effect in the optimal disintegration relative to the high degree strategy. The right side of each graph shows a visualization of the optimal strategy on the multiplex network of US domestic flights. We depict the intersection graphs for the networks, where nodes belonging to the largest connected component are represented with circles and other nodes are represented with squares. Moreover, the size of each node is proportional to the probability of finding that nodes in different optimal disintegration strategies with the same disintegration effect. The nodes with higher probability are shown in red, and other nodes are shown in black. The result shows that as the number of nodes removed increases, the effect of the optimal collapse strategy steadily increases, and most of the high-frequency nodes appear in the largest connected cluster of the intersection graph.

deletion at each step, this is referred to as an HDA algorithm. As in the HD case, we define the method using the product of the degree.

The EI algorithm is based on a method referred to as explosive percolation in a single-layer network, which was introduced by Achlioptas *et al.*[46] Clusella *et al.*[23] proposed a modified version to solve the optimal immunization problem related to explosive percolation. Osat *et al.*[38] generalized this method to multiplex networks. In this experiment, we refer to this method as the explosive immunization (EI) algorithm and utilize it to create disintegration strategies (see the above reference for details).

The results on the different coupled networks are depicted in Figs. 2(a)–2(c), respectively. It is worth mentioning that in the tabu search algorithm, we employ the solution given by the HD method as the initial solution to improve the search efficiency. Both layers are generated according to the BA model or the ER model, with $N = 1000$ nodes and an average degree of approximately $\langle k \rangle = 4$ (i.e., $p = 0.008$ in the ER model, and we add four nodes at each step in the BA model). We determine the mean value of Φ over 20 realizations of the tabu algorithm and 100 realizations of the other strategies.

We find that given the fraction q of nodes, the disintegration effect of the tabu search algorithm (called the optimal disintegration strategy) is considerably better than those for the other disintegration strategies, which is especially true on the coupled ER network. On the coupled SF network, the effects of the degree strategies are incredibly close to the optimal strategy. In addition, in the above results, as q increases, the transition of the disintegration effect becomes more smooth in our strategy than the others, and the catastrophic cascade is less obvious.

Moreover, we illustrate the computation time of the tabu algorithm as a function of the network size N in Fig. 2(d), where the box plots are drawn based on over 100 independent realizations. To ensure that the initial size of GMCC is equal to 1, we choose the coupled SF network to calculate the time. We find that the time complexity of the optimal disintegration strategy based on tabu search is approximately $O(N)$.

B. Experiments on real networks

In order to test the disintegration effect of the optimal strategy on real systems and observe the characteristics of nodes in optimal strategies, we consider the multiplex network of US domestic flights operating in January 2014. In the networks, airports are nodes and connections on each layer are determined by the existence of at least one flight between two locations.[30] The basic data for the multiplex transportation networks are shown in Table I. The value of the parameters in the algorithm and the experimental environment are the same as those in Sec. IV A.

The intersection between the layers: Radicchi[30] decomposed a multiplex network into two uncoupled graphs: the intersection between the layers and the remainders of each layer. The former is a network consisting of overlapping edges in different layers (i.e., the intersection graph with the adjacency matrix is given by the Hadamard product of first and second layers). It was demonstrated that when the intersection dominates the remainders, an interconnected network undergoes a smooth percolation transition.

TABLE I. The basic data for the multiplex transportation networks. The first column identifies the names of the different pairs of layers used to construct coupled networks. For each of these, we report the following columns: number of nodes N, number of nodes N_{inter} in the giant connected component (GCC) of the intersection graph, the total number of edges E_{inter} in the intersection graph, the number of nodes in the GCC of the remainder of the first or the second layer (N_1 and N_2), and the number of edges present only in the first or the second layer (E_1 and E_2).

Layers	N	N_{inter}	E_{inter}	N_1	E_1	N_2	E_2
Delta-American Airline	84	49	68	76	190	79	374
American Airline-United	73	42	68	68	161	69	202
Delta-United	82	39	56	78	348	78	226

To observe the improvement of the disintegration effect of the optimal strategy compared with the other general strategies, we choose the degree strategies (HD and HDA) for comparison. It is worth mentioning that transportation networks are coupled in a connective manner. Therefore, when a node is removed, we only need to remove the adjacent edges in each layer. Moreover, to achieve a better performance, in HDA strategy, we calculate the degrees of nodes through the number of adjacent edges that belong to MCCs. The results are shown in the left part of Fig. 3. We can see the improvement of the optimal strategy effect compared with the HD strategy through the shadow area. Moreover, we choose the optimal strategies with the greatest improvement. In the 100 independent experiments, we obtained 20, 20, and 19 different sets of nodes, all of which have the optimal disintegration effect. In these optimal disintegration strategies, we count the number of occurrences of different nodes and consider those that appear more than twice as those with less substitutability. On the right side of Fig. 3, we visualize these nodes in the intersection graphs of different networks. We observe that most nodes with less substitutability are on the largest connected component of the intersection between different layers. There are a small number of nodes in other locations, and the overall distribution is geographically uniform.

V. CONCLUSION AND DISCUSSION

Network disintegration is a key issue in complex network fields, which can guide effective attacks on harmful networks. Conversely, network disintegration methods can be utilized to determine the weak parts of a network, helping us to protect vital networks.

In this study, we generalized the optimization model for a disintegration strategy to multiplex networks and solved it using a tabu search to obtain the optimal disintegration strategy. The disintegration effects of the optimal disintegration strategies have been verified by comparison with other strategies on various models and real networks. We can approximate the best choice of the set of nodes through a global search. Here, we only discussed some coupled interdependent or interconnected networks with two layers. The performance of the disintegration strategy on other multiplex network models or networks with different multi-layer types requires further study.

Moreover, the algorithm we utilized in this study is a serial global search algorithm, which is more suitable for dealing with medium-sized networks. When facing certain large-scale networks or demanding faster results, we require further optimization of the algorithm or to sacrifice part of the optimization effect to improve the algorithm efficiency.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (NNSFC) (Grant Nos. 71871217, 71690233, and 71771214). We thank Professor Filippo Radicchi and Dr. Saeed Osat for their assistance with datasets of real network and code of the EI algorithm used for comparison.

[1] M. E. J. Newman, "Networks: An Introduction," Astron. Nachri. **327**(8), 741–743 (2010).
[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," Phys. Rep. **424**, 175–308 (2006).
[3] J. W. Simpson-Porco, F. Drfler, and F. Bullo, "Voltage collapse in complex power grids," Nat. Commun. **7**, 10790–10790 (2016).
[4] D. Helbing, D. Brockmann, T. Chadefaux, K. Donnay, U. Blanke, O. Woolley-Meza, M. Moussaid, A. F. Johansson, J. Krause, S. Schutte, and M. Perc, "Saving human lives: What complexity science and information systems can contribute," J. Stat. Phys. **158**, 735–781 (2015).
[5] B. Bollobs and O. Riordan, "Robustness and vulnerability of scale-free random graphs," Internet Math. **1**, 1–35 (2004).
[6] J. Wu, M. Barahona, Y.-J. Tan, and H.-Z. Deng, "Spectral measure of structural robustness in complex networks," Syst. Man Cybern. **41**, 1244–1252 (2011).
[7] Z.-X. Wu and P. Holme, "Onion structure and network robustness," Phys. Rev. E **84**, 26106 (2011).
[8] A. Guggiola and G. Semerjian, "Minimal contagious sets in random regular graphs," J. Stat. Phys. **158**, 300–358 (2015).
[9] Z. Wang, C. T. Bauch, S. Bhattacharyya, A. d'Onofrio, P. Manfredi, M. Perc, N. Perra, M. Salath, and D. Zhao, "Statistical physics of vaccination," Phys. Rep. **664**, 1–113 (2016).
[10] J. Qin, J. J. Xu, D. Hu, M. Sageman, and H. Chen, "Analyzing terrorist networks: A case study of the global Salafi Jihad network," in *Intelligence and Security Informatics* (Springer, 2005), pp. 287–304. 10.1007/11427995_24
[11] M. Perc, J. J. Jordan, D. G. Rand, Z. Wang, S. Boccaletti, and A. Szolnoki, "Statistical physics of human cooperation," Phys. Rep. **687**, 1–51 (2017).
[12] Y. Moreno, M. Nekovee, and A. F. Pacheco, "Dynamics of rumor spreading in complex networks," Phys. Rev. E **69**, 66130 (2004).
[13] V. Capraro and M. Perc, "Grand challenges in social physics: In pursuit of moral behavior," Front. Phys. **6**, 107 (2018).
[14] Y. Deng, J. Wu, and Y.-J Tan, "Optimal attack strategy of complex networks based on tabu search," Phys. A Stat. Mech. Appl. **442**, 74–81 (2016).
[15] D. Ye and W. Jun, "Optimal attack strategy with heterogeneous costs in complex networks," in *2016 Annual IEEE Systems Conference (SysCon)* (IEEE, 2016), pp. 1–5.
[16] R. Albert, H. Jeong, and A.-L. Barabsi, "Error and attack tolerance of complex networks," Nature **406**, 378–382 (2000).
[17] M. Kitsak, L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley, and H. A. Makse, "Identification of influential spreaders in complex networks," Nat. Phys. **6**, 888–893 (2010).
[18] E. Estrada and J. A. Rodriguez-Velazquez, "Subgraph centrality in complex networks," Phys. Rev. E **71**, 56103 (2005).
[19] L. L'u, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, "Vital nodes identification in complex networks," Phys. Rep. **650**, 1–63 (2016).
[20] L. Zdeborov, P. Zhang, and H.-J. Zhou, "Fast and simple decycling and dismantling of networks," Sci. Rep. **6**, 37954 (2016).
[21] A. Braunstein, L. DallAsta, G. Semerjian, and L. Zdeborov, "Network dismantling," Proc. Natl. Acad. Sci. U.S.A. **113**, 12368–12373 (2016).
[22] S. Mugisha and H.-J. Zhou, "Identifying optimal targets of network attack by belief propagation," Phys. Rev. E **94**, 12305 (2016).
[23] P. Clusella, P. Grassberger, F. J. Perez-Reche, and A. Politi, "Immunization and targeted destruction of networks using explosive percolation," Phys. Rev. Lett. **117**, 208301 (2016).
[24] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," Nature **524**, 65–68 (2015).
[25] A. Szolnoki and M. Perc, "Collective influence in evolutionary social dilemmas," Europhys. Lett. **113**, 58004 (2016).
[26] F. Morone, B. Min, L. Bo, R. Mari, and H. A. Makse, "Collective influence algorithm to find influencers via optimal percolation in massively large social media," Sci. Rep. **6**, 30062 (2016).
[27] X. Zhang, J. Wu, H. Wang, J. Xiong, and K. Yang, "Optimization of disintegration strategy for multi-edges complex networks," in *2016 IEEE Congress on Evolutionary Computation (CEC)* (IEEE, 2016), pp. 522–528.
[28] M. Szell, R. Lambiotte, and S. Thurner, "Multirelational organization of large-scale social networks in an online world," Proc. Natl. Acad. Sci. U.S.A. **107**, 13636–13641 (2010).
[29] M. D. Domenico, A. Sol-Ribalta, S. Gmez, and A. Arenas, "Navigability of interconnected networks under random failures," Proc. Natl. Acad. Sci. U.S.A. **111**, 8351–8356 (2014).
[30] F. Radicchi, "Percolation in real interdependent networks," Nat. Phys. **11**, 597 (2015).
[31] S. Majhi, M. Perc, and D. Ghosh, "Chimera states in a multilayer network of coupled and uncoupled neurons," Chaos **27**, 73109 (2017).
[32] F. Battiston, V. Nicosia, V. Nicosia, V. Latora, V. Latora, and V. Latora, "Structural measures for multiplex networks," Phys. Rev. E **89**, 32804 (2014).
[33] Z. Wang, L. Wang, A. Szolnoki, and M. Perc, "Evolutionary games on multilayer networks: A colloquium," Eur. Phys. J. B **88**, 124 (2015).
[34] M. Dickison, S. Havlin, and H. E. Stanley, "Epidemics on interconnected networks," Phys. Rev. E **85**, 66109 (2012).
[35] S. Gomez, A. Diaz-Guilera, A. Diaz-Guilera, J. Gomez-Gardees, C. J. Perez-Vicente, Y. Moreno, and A. Arenas, "Diffusion dynamics on multiplex networks," Phys. Rev. Lett. **110**, 28701 (2013).
[36] X. Huang, J. Gao, S. V. Buldyrev, S. Havlin, and H. E. Stanley, "Robustness of interdependent networks under targeted attack," Phys. Rev. E **83**, 65101 (2011).
[37] L. M. Shekhtman, M. M. Danziger, and S. Havlin, "Recent advances on failure and recovery in networks of networks," Chaos Solitons Fractals **90**, 28–36 (2016).
[38] S. Osat, A. Faqeeh, and F. Radicchi, "Optimal percolation on multiplex networks," Nat. Commun. **8**, 1540 (2017).
[39] F. Glover, "Future paths for integer programming and links to artificial intelligence," Comput. Oper. Res. **13**, 533–549 (1986).
[40] K.-M. Lee, B. Min, and K.-I. Goh, "Towards real-world complexity: An introduction to multiplex networks," Eur. Phys. J. B **88**, 48 (2015).
[41] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," Nature **464**, 1025–1028 (2010).
[42] Y. Yu, Y. Deng, S.-Y. Tan, and J. Wu, "Efficient disintegration strategy in directed networks based on tabu search," Phys. A Stat. Mech. Appl. **507**, 435–442 (2018).
[43] P. Erd'os and A. Rényi, "On the evolution of random graphs," Trans. Am. Math. Soc. **286**(1), 257–274 (2011).
[44] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," Science **286**, 509–512 (1999).
[45] M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: Uses and interpretations," Neuroimage **52**, 1059–1069 (2010).
[46] D. Achlioptas, R. M. D'souza, and J. Spencer, "Explosive percolation in random networks," Science **323**, 1453–1455 (2009).