An Improved Differential Evolution Framework Using Network Topology Information for Critical Nodes Detection

Shanqing Yu, Yongqi Wang, Jiaxiang Li, Xu Fang, Jinyin Chen, Ziwan Zheng, and Chenbo Fu, *Member, IEEE*

***Abstract*— Critical nodes detection (CND) focuses on identifying the nodes that significantly impact the network's robustness and is applied in various fields such as power grids, communication networks, and disease spreading. However, detecting the critical nodes is a challenging nondeterministic polynomial time complete (NP-complete) problem. One possible solution is using the evolutionary algorithm which has a high global search capability. However, the existing evolutionary algorithms for CND only focus on independent nodes, ignoring the underlying relationship among the nodes. Thus, in this work, we proposed a new topology-combined differential evolution framework called TDE to explore the possibility of improving the performance by fusing topology information, which designs individual genotypes through node degree, and new mutation and decoding-based selection operators are designed for these genotypes to use topology information effectively. The experiments on synthetic and real networks show that it is feasible to improve the search capability of the algorithm by fusing node degree information.**

***Index Terms*— Complex network, critical nodes detection (CND), differential evolution (DE), evolutionary computation.**

I. Introduction

THE critical nodes detection (CND) in network science aims to identify a set of nodes; removing these nodes will maximize or minimize a predefined graph connectivity metric [19], as Fig. 1 shows. The important role of CND in network theory has continuously attracted wide attention from various fields such as social [5], infrastructure [6] and transmission networks [36] and is it well-studied [12], [14], [21]. For instance, in infrastructure networks such as power grids and road networks, targeted prevention and protection of critical nodes will effectively reduce losses caused by malicious attacks or unexpected failures. In contrast, in transmission networks for viruses or forest fires, timely blocking of critical nodes facilitates propagation control and it can be used in the biomedical field to control the spread of diseases and the development of anticancer drugs.

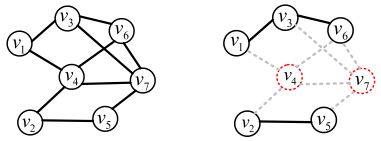


Fig. 1. Simple illustration of critical nodes. Removal of critical nodes will cause largest impact on the connectivity of the original network. (a) Original network. (b) Residual network after removing the critical nodes.

There are lots of approaches for the CND problem, which can be simply summarized into two categories, i.e., exact and approximate approaches. The exact approaches focus on the theoretical optimal solution in the whole solution space. For example, Shen and Smith [29] proposed an optimal polynomial-time dynamic programming algorithm for several types of CND on tree structures and series–parallel networks. Summa et al. [9] proposed an integer linear programming model with a nonpolynomial number of constraints to solve CND in polynomial time. Veremyev et al. [35] developed more compact linear 0–1 formulations for the considered types of CND with $O(n^2)$ entities. Rezaei et al. [27] proposed the exact iterative algorithm for CNDP (EIA-CNDP) algorithm to improve the mixed integer linear programming model for solving the CND problem more efficiently. However, existing studies show that the CND problem is a nondeterministic polynomial time complete (NP-complete) problem [19], [20]. Although new approaches [35] have constantly been intended to alleviate this problem, they are still computationally expensive when dealing with large networks. Therefore, high computational complexity caused by the large solution space of CND limits the application of exact approaches.

The approximate approaches balance the tradeoff between the computational cost and performance, making it possible to solve CND in a larger solution space. The greedy-strategy-based algorithms are efficient and acceptable approaches to the approximate solution. For CND, the greedy

Manuscript received 8 June 2022; revised 15 September 2022; accepted 19 October 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62103374, in part by the Basic Public Welfare Research Project of Zhejiang Province under Grant LGF20F020016, in part by the National Key Research and Development Program of China under Grant 2018AAA0100800, and in part by the Open Project of Key Laboratory of Ministry of Public Security Information Application Based on Big Data Architecture under Grant 2020DSJSYS003. *(Corresponding author: Chenbo Fu.)*

Shanqing Yu, Yongqi Wang, Jiaxiang Li, Xu Fang, Jinyin Chen, and Chenbo Fu are with the Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 310023, China (e-mail: yushanqing@zjut.edu.cn; 2111903062@zjut.edu.cn; jiaxiangli.zjut@gmail.com; henryfangx@gmail.com; chenjinyin@zjut.edu.cn; cbfu@zjut.edu.cn).

Ziwan Zheng is with the Zhejiang Police College, Hangzhou 310000, China (e-mail: zhengziwan@zjjcxy.cn).

Digital Object Identifier 10.1109/TCSS.2022.3217071

strategy is based on the traditional network indicators such as degree [1], k-shell [40], and centrality [17], which can approximate critical nodes but are accompanied by a large loss. Except for the traditional indicators, some new indicators and strategies have been developed. For instance, Ventresca and Aleman [33] proposed a depth-first search solution with the greedy strategy for quickly identifying critical nodes in large networks. Khomami et al. [18] proposed the community finding influential node (CFIN) algorithm for detecting influential nodes in the network combining the community structure and using a greedy algorithm to optimize the objective function in local search. Pullan [25] proposed a multistart greedy algorithm for better performance in several types of CND. Chen et al. [7] designed a novel metric and a nonconvex mixed-integer quadratic programming model, and on this basis, the greedy strategy is used to solve CND in undirected weighted networks. These approaches achieve acceptable solutions within a feasible time. However, losses caused by the greedy-based algorithms are still depressing when the solution space becomes larger and more complex.

Randomness is an important element for global searching in the approximate algorithms, such as the simulated annealing optimization algorithm, the swarm intelligence optimization Algorithm [23], [31], the multiobjective evolutionary algorithm [24], the population-based incremental learning algorithm [32], the randomized rounding algorithm [34], and the random walk algorithm [11], [15]. The evolutionary algorithm is an effective way to tackle such complex discrete problems [41], and many evolution-based algorithms have been proposed for CND. For instance, Qiu et al. [26] proposed the local-influence-descending differential evolution (LIDDE) algorithm for solving the influence maximization problem based on the differential evolutionary algorithm. Aringhieri et al. [2] proposed a flexible evolutionary framework for solving several variants of CND by adopting the standard genetic algorithm (GA). Zhou et al. [44] combined the evolutionary algorithms with local search operators and proposed an effective memetic algorithm for solving CND. Furthermore, Zhou et al. [43] proposed a variable population memetic algorithm with better performance on this problem. Meanwhile, the metaheuristic algorithms, such as variable neighborhood search proposed by Hansen et al. [16] and iterated local search proposed by Lourenço et al. [22], combine the random element with the local search algorithms to achieve better performance.

However, these algorithms treat the nodes as the independent elements, ignoring the topology relationship between the nodes, e.g., propinquity in the distance and the similarity in structure. These underlying correlations may help improve the performance of the evolutionary algorithms. For instance, when a critical node is identified, another node with similar topological characteristics is likely to become another critical node.

In this work, we proposed the topology-combined differential evolution (TDE) framework for CND, which can improve CND performance using network topology information. The proposed method consists of two parts, i.e., the topology-combined genotype design module for fusing topology information and the information-combined differential evolution (DE) module for identifying critical nodes. Since node degree is a simple but essential topological feature in a network, nodes with a high degree usually have a more significant impact on the robustness of the network [4]. In addition, the similarity between node degrees can be mapped into similarity on node topological information, so we extract the degrees as genotypes in the topological combination genotype design module. Furthermore, the DE algorithm is simple in principle, with few parameters and high robustness. Compared with the traditional evolutionary algorithms (e.g., GA), DE converges faster when individuals are represented as real-valued vectors due to its adaptive nature in the variation process [8]. After using normalized degree values instead of node symbols, the resulting individuals are encoded to be represented exactly in the form of real-valued vectors and contain network topology information. For these reasons, DE is used in this article to construct the DE module. The experiments on degree-based TDE show it is feasible to improve the search capability of the algorithm by fusing network topology information. This work may provide a new path for improving the performance of algorithms on complex networks [13], [39], [42]. The main contributions of the article as described as follows.

1) We propose the TDE framework, in which nodes' degree is used for genotype designing. Moreover, we also design the mutation and decoding-based selection operators to make TDE efficient and feasible. To our knowledge, this is the first attempt to improve the performance of the evolutionary algorithm by designing the genotype with network topology information.
2) Experiments on both the synthetic and real datasets verify the effectiveness of improving the algorithm's efficiency by fusing topology information.

The rest of this article is summarized as follows. In Section II, we define the CND used in the article. In Section III, we introduce the proposed TDE framework. In Section IV, we verify the influence of network topology information fusion on algorithm efficiency through experiments. Finally, we give a summary in Section V.

II. Definition

CND identifies nodes that have a significant impact on network connectivity, and therefore, connectivity metrics such as maximizing the number of connected components, minimizing pairwise connectivity, and minimizing the largest component size are frequently used in CND. However, these connectivity metrics cannot reflect the dynamical robustness of the network when it suffers continuous malicious attacks. Thus, we adopt the network robustness metric R [28] based on continuous attacks as the predefined connectivity metric R, which can be defined as

$$R = \frac{1}{n}\sum_{q=1}^{N} S(q) \tag{1}$$

where n is the number of nodes in the network, and $S(q)$ is the fraction of nodes in the largest connected subgraph after removing q nodes. In calculating $S(q)$, the q nodes are

selected individually, that is, choosing the node with the highest degree in the remaining network each time. Furthermore, CND is defined as follows.

Definition: Given an integer k and a undirected unweighted network $G(V)$, where V denotes the set of nodes in G. The CND aims to find a set of nodes $V^- \subseteq V$ within the network satisfying the following equation:

$$\begin{aligned} &\max \quad \text{Fitness}\left(V^-\right) = R(G(V)) - R\left(\hat{G}\left(V - V^-\right)\right) \\ &\text{s.t.} \quad |V^-| <= k \end{aligned} \tag{2}$$

where k is the number of critical nodes, and Fitness(V^-) is the fitness value of nodes' set V^-. The result of maximizing Fitness can be expressed as $V^{-^{\text{critical}}} = \{\text{node}_1^{\text{critical}}, \text{node}_2^{\text{critical}}, \ldots, \text{node}_k^{\text{critical}}\}$.

III. Method

In this section, the details of the TDE framework are introduced first, which includes the topology-combined genotype design module and the DE module as shown in Fig. 2. Furthermore, the degree-based TDE is realized and the visualization solution space of CND is also given to explain and analyze the effectiveness of the proposed method.

A. Topology-Combined Genotype Design

Individuals in the evolutionary algorithm are usually set as the candidate solutions of the target problem, and in CND, the search target is a set of nodes. However, using the nodes' collection as an individual's gene directly cannot use the topology information of the network during the process of evolution search. Therefore, this method incorporates topological information and uses it to improve search efficiency. The individuals are set into two forms, phenotype, and genotype. As shown in Fig. 3, the phenotype as indi_pheno$_i$ is set to the collection of nodes, and the genotype as indi_geno$_i$ is set to the collection of node representation containing topology information. The **mapping-based encoding** and **similarity-based decoding** are also designed to transform between the phenotype and the genotype, and both of them depend on the node representation dictionary NR obtained from the process of **node representation construction** as shown in Fig. 3. The following will introduce the node representation construction, the mapping-based encoding, and the similarity-based decoding in detail.

1) Node Representation Construction: Node representation construction aims to transform the node into the topology-combined representation, which can be expressed as the node representation dictionary NR $= \{\text{node}_i : \text{rep}_i\}$, where $i = 1, 2, 3, \ldots, |V|$, node$_i$ is the ith node in G, rep$_i$ is the topology-combined representation of node$_i$, and node$_i$: rep$_i$ is one of the key–value pairs in collection NR. The node representation construction strategy can be different when the topology information to be fused is different. In this article, node degree is adopted to construct the node representation dictionary NR $= \{\text{node}_1 : \text{degree}_1, \text{node}_2 : \text{degree}_2, \ldots, \text{node}_n : \text{degree}_n\}$, where degree$_i$ is the normalized degree of node$_i$.

2) Mapping-Based Encoding: Mapping-based encoding aims to not only transform the individuals' phenotype into the genotype but also make the genotype contain network topology information. The phenotype of an individual is a set of nodes, and it can also be regarded as a set of keys in the node representation dictionary NR. Considering that the values in NR contain topology information, the key-to-value mapping strategy is used to achieve encoding, which is shown in Fig. 3. After key-to-value mapping, the genotype of the individual satisfies both crossover-mutation executable and topology containment. The meaning of encoding will be explained in the follow-up content.

3) Similarity-Based Decoding: Fitness calculation for selection is based on the individual phenotype, and thus, it is necessary to transform the genotype into the phenotype by decoding. However, after crossover and mutation, the node representations are changed and cannot be transformed into phenotype by reverse value-to-key mapping. Therefore, this article designs similarity-based decoding to select nodes most similar to the genotype denoted as Decoding(). Take the genotype of one individual indi_geno$_m$ for example, the main process of decoding can be expressed by the following equation:

$$\begin{aligned} &\text{indi\_pheno}_m[j] \\ &\quad = \text{Decoding}\left(\text{indi\_geno}_m[j]\right) \\ &\quad = \arg\min_{\text{node}_i \in V} \quad \text{Distance}\left(\text{indi\_geno}_m[j], \text{NR}[\text{node}_i]\right) \end{aligned} \tag{3}$$

where indi_pheno$_m$ is the phenotype of indi_geno$_m$ after decoding, indi_pheno$_m[j]$ is the jth node in indi_pheno$_m$, and indi_geno$_m[j]$ is the jth gene in indi_geno$_m$, where $j = 1, 2, \ldots, k$ is the gene locus as shown in Fig. 3. NR[node$_i$] is the node representation of node$_i$, where $i = 1, 2, \ldots, |V|$ is used to traverse every element in NR. Distance() is the function to measure the distance between indi_geno$_m[j]$ and NR[node$_i$], and the more similar the two are, the smaller the function value is. When there are multiple nodes whose node representation have the same distance as indi_geno$_m[j]$ and are the minimum value, one of these nodes will be selected randomly as the decoded result. By the formula above, the nodes that have the closest node representation to the gene in indi_geno$_m$ can be obtained one by one, and after k operations, the phenotype of indi_geno$_m$ can be obtained as indi_pheno$_m$. In the degree-based method, the normalized node degree is a real value between 0 and 1, and thus, we adopt the Euclidean distance to measure the similarity in the similarity-based decoding. And Distance() in (3) can be stated as follows:

$$\begin{aligned} &\text{Distance}\left(\text{indi\_geno}_m[j], \text{NR}[\text{node}_i]\right) \\ &\qquad = \left|\text{indi\_geno}_m[j] - \text{NR}[\text{node}_i]\right|. \end{aligned} \tag{4}$$

B. DE Module

The DE module realizes the global optimization of the target problem by the iterative update of the individuals in the population, and it is the core component for CND in this work.

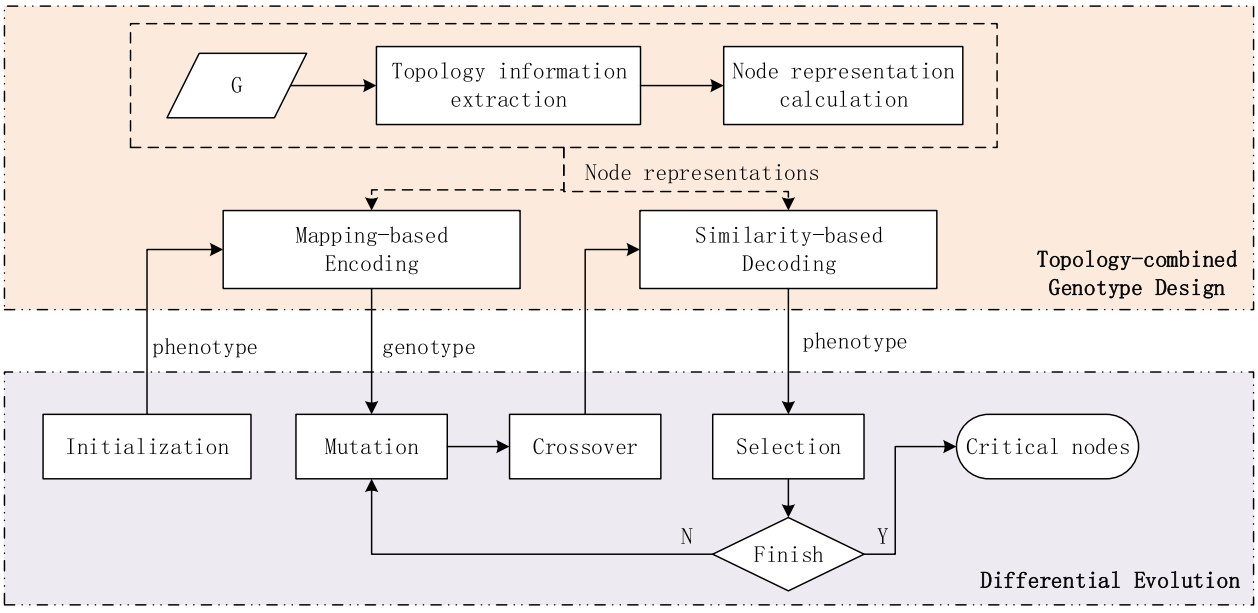


Fig. 2. Framework of TDE, which consists of the topology-combined genotype design module and DE module.

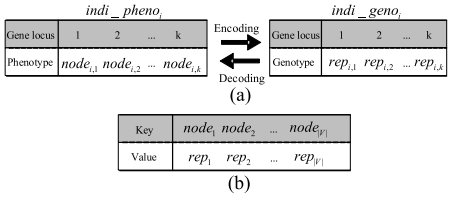


Fig. 3. Process of mutual conversion between phenotype and genotype by the node representation dictionary. (a) Encoding and decoding between phenotype and genotype. (b) Node representation dictionary NR.

The overall process of the DE module is shown in Algorithm 1. In Algorithm 1, indi_geno$_i^g$ represents the ith individual's genotype in the population of the gth generation. x_{mutate} is the temporary individual in genotype after mutation. $x_{\text{crossover}}$ is the temporary individual in genotype after crossover, and it also represents the new individual generated by indi_geno$_i^g$. Details of initialization, crossover, mutation, and selection are introduced as follows.

1) Initialization: Initialization aims to generate initial individuals as initial candidate solutions. CND aims to find the set of nodes that have the largest influence on the connectivity of the original network. Thus, initialization in TDE is to generate a set of initial individuals, which can be stated as Init_population $=$ {indi_pheno$_1$, indi_*ph*eno$_2$, ..., indi_pheno$_{\text{pop_size}}$}, where pop_size is the presett population size, and indi_pheno$_i$ is the ith individual of phenotype in Init_population, which can be stated as indi_pheno$_i$ $=$ {indi_pheno$_i$[1], indi_pheno$_i$[2], ..., indi_pheno$_i$[k]}, where k is the preset number of critical nodes, and indi_pheno$_i$[j] is the jth nodes in indi_pheno$_i$.

2) Mutation: Mutation is the operator used for global search in the evolutionary algorithms. In TDE, each individual in the population mutates according to the following equation:

$$x_{\text{mutate}} = \text{indi\_geno}_i + F \cdot \left(x_{\text{superior}} - x_{\text{inferior}}\right) \tag{5}$$

where indi_geno$_i$ denotes the ith individual in the population, and x_{mutate} is the temporary individual in genotype after mutate. According to Storn and Price [30], $F \in [0, 2]$ is a natural and constant factor, which controls the amplification of the differential variation $(x_{\text{superior}} - x_{\text{inferior}})$. x_{superior} and x_{inferior} are the superior individual and inferior individual in population selected by roulette, respectively. The probability of individual being selected as x_{superior} can be obtained according

Algorithm 1 TDE

Input: The network $G(V)$, node representations' dictionary NR, max generation max_g, population size pop_size
Output: Critical nodes $critical\_nodes$

```
Init_population = Initialization(G)
Population_0 = Encoding(Init_population)
for g = 0; g < max_g; g++ do
  for i = 0; i < pop_size; i++ do
    fitness = Fitness(Decoding(indi_geno_i^g))
    x_mutate=Mutate(indi_geno_i^g)
    x_crossover=Crossover(x_mutate)
    new_fitness=Fitness(Decoding(x_crossover))
    if new_fitness > fitness then
      indi_geno_i^{g+1} = x_crossover
    else
      indi_geno_i^{g+1} = indi_geno_i^g
    end if
  end for
end for
critical_nodes=Decoding(Max_fitness
  (Population^{max_g}))
Return critical_nodes
```

to the following equation:

$$P_{\text{superior}}(\text{indi\_geno}_i) = \frac{\text{Fitness}(\text{Decoding}(\text{indi\_geno}_i))}{\sum_{j=1}^{n} \text{Fitness}(\text{Decoding}(\text{indi\_geno}_j))}. \quad (6)$$

In the equation, $P_{\text{superior}}(\text{indi\_geno}_i)$ is the probability indi_geno_i is selected as x_{superior}. $\text{Decoding}(\text{indi\_geno}_i)$ is the similarity-based decoding operation described above for obtaining the representation indi_pheno_i of indi_geno_i, i.e., (3), which is used to find the node with the most similar genotype to indi_geno_i. $\text{Fitness}(\text{Decoding}(\text{indi\_geno}_i))$ is the fitness of indi_geno_i, where $i = 1, 2, \ldots, |V|$, which can be used to calculate the selected probability of each individual in population. It is obvious that individual with larger fitness has a higher probability to be selected as x_{superior} in this way. Similarly, the probability of individual being selected as x_{inferior} can be obtained according to the following equation:

$$P_{\text{inferior}}(\text{indi\_geno}_i) = \frac{1 - \text{Fitness}(\text{Decoding}(\text{indi\_geno}_i))}{\sum_{j=1}^{n} 1 - \text{Fitness}(\text{Decoding}(\text{indi\_geno}_j))}. \quad (7)$$

In addition, we define the function Roulete_superior(Population) to select one individual in Population as x_{superior} based on the probability obtained by (6). Furthermore, Roulete_inferior(Population) can be defined in a similar way. Algorithm 2 shows the process of mutation.

Algorithm 2 Mutation in TDE

Input: The individual for mutation $Population_i^g$, the gth generation population $Population^g$
Output: Mutated individual x_{mutate}

```
x_superior = Roulete_superior(Population^g)
x_inferior = Roulete_inferior(Population^g)
x_mutate = Population_i^g + F · (x_superior − x_inferior)
Return x_mutate
```

3) Crossover: Crossover is the operator used for local search in evolutionary algorithms. The multipoint crossover is adopted in TDE. It selects two individuals randomly and decides whether to execute a gene exchange operation based on the crossover rate. Algorithm 3 shows the process of crossover.

Algorithm 3 Crossover in TDE

Input: x_{mutate}, $Population^g$, crossover rate pc, number of nodes k
Output: $x_{crossover}$

```
x_r = Roulete_superior(Population^g)
x_crossover = x_mutate
for i = 0; i < k; i++ do
    if random(0, 1) > pc then
        x_crossover[i] = x_r[i]
    end if
end for
Return x_crossover
```

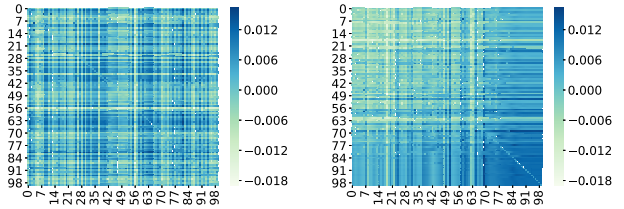


Fig. 4. Solution space of selecting two critical nodes in networks. The abscissa and ordinate, respectively, represent a candidate critical node, and the color depth represents the fitness of the candidate solution. (a) With no topology information combining. (b) With degree topology information combining.

4) Selection: The selection operation is the decision-making process of individual retention in the population. The selection operator used in this article is consistent with the original DE algorithm [30], and each individual in the population should compete with the individual obtained after mutation and crossover, and the one with higher fitness will be retained. This operation is based on the following equation:

$$\text{indi\_geno}_i^{g+1} = \begin{cases} \text{indi\_geno}_i^g, & \text{Fitness}(\text{indi\_geno}_i^g) \\ & > \text{Fitness}(x_{\text{crossover}}) \\ x_{\text{crossover}}, & \text{else.} \end{cases} \quad (8)$$

C. Visual Interpretation on Degree-Based TDE

Fig. 4 shows the solution space of selecting two critical nodes in the network with 100 nodes, where both the abscissa and the ordinate represent the candidate critical node name and the color depth represents the fitness value of the candidate solution. It can be seen that when the node order in the abscissa and ordinate uses meaningless serial numbers directly, the figure of solution space appears chaotic. However, when using degree centrality instead of serial numbers, the figure of solution space becomes orderly, as shown in Fig. 4(b). In this case, nodes with similar degree values will have similar representations, which means they are close to each other on both the axes. From another perspective, it can be regarded as degree-based ordering for the solution space, which leads to the search process in such a solution space being easier.

Specifically, the sorted solution space satisfies the following features.

1) The trend of diagonal increasing. The solution space shows an increasing diagonal trend, that is, the average quality of the solutions in the lower right region is better than that in the upper left region. It is because combinations of nodes with larger degree values usually have a larger impact on network connectivity.
2) Block distribution. There are multiple block regions in the solution space where the qualities of the solutions are relatively similar. It is because the degrees of nodes in this area are relatively close, and the impacts on network connectivity are similar.

Among the above two features, the former is a kind of orderliness, which is beneficial to evolutionary search. While

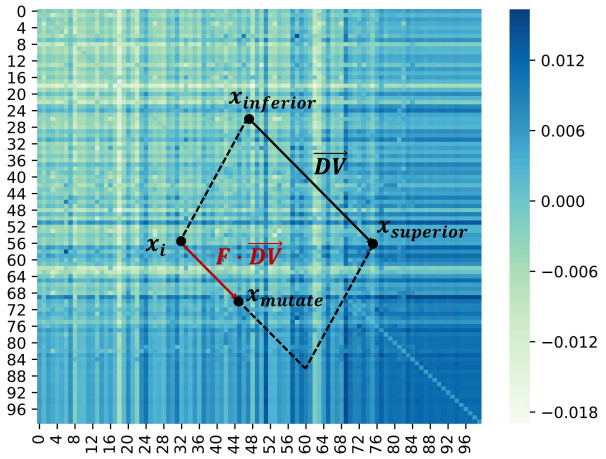


Fig. 5. Differential variable between the superior and inferior individuals can provide guidance information to improve the search efficiency of evolution search.

the latter is a kind of disorderliness, and blocks with large area will reduce the regularity of the solution space.

In this situation, due to the property of diagonal increasing in solution space, the superior individuals would have a higher probability of appearing in the high fitness area (lower, right), and the inferior individuals would appear in the low fitness area (upper, left). Thus, by involving differential variables obtained from the difference between the superior and inferior individuals, the DE algorithm would have a higher probability of searching for the optimal solution through the area in which nodes have a larger degree (see Fig. 5). The above process can be viewed as a preference-based search method. Compared with the search process in the original solution space, the proposed preference-based search method has a better convergence rate. That Is because the former includes completely random mutation and has a greater chance of repeatedly searching through the same solutions due to the chaotic solution distribution. On the other hand, the block distribution in the solution space is caused by nodes with the same or similar degree values, which cannot be effectively ordered in the block region. Thus, if the difference among nodes' degree is small, the block area will be large, and the solution space is less ordered, so the improvement on convergence efficiency can be limited.

Based on the analysis above, it can be concluded that the more orderly the solution space is after combining degree information, the more obvious the increasing diagonal trend becomes and the smaller block area appears, which leads to a greater improvement in algorithm efficiency. It will be explained in Section IV.

IV. Experiment

In this section, we apply our proposed method on four synthetic datasets and four real datasets to verify the effectiveness of the algorithm.

A. Experiments on Synthetic Datasets

To evaluate the effectiveness of our proposed method, we generate four networks, i.e., regular graph (RG) network, Erdös–Rényi (ER) network, Watts–Strogatz (WS) network, and Barabási–Albert (BA) network. For a fair comparison, all the networks have the same number of nodes and edges approximately. The details of these datasets are described as follows.

1) RG: The RG used in this article is the incomplete regular network, in which each node has the same degree value. The number of nodes in RG is 300, and the number of edges is 1200.
2) ER Network [10]: Nodes are connected according to a certain probability in ER random networks. The number of nodes in ER is 300, AND the number of edges is 1196.
3) WS Network [38]: Compared with the random network with the same size nodes, the WS network has a shorter average path length and larger clustering coefficient. The number of nodes in WS is 300, and the number of edges is 1200.
4) BA Network [3]: In BA networks, the degree distribution of nodes satisfies the power-law distribution and has high heterogeneity. The number of nodes in BA is 300, and the number of edges is 1184.

The baselines used in the experiments are described as follows.

1) Random: Select k nodes randomly as critical nodes.
2) Greedy-Degree: Select k nodes with maximum degree as critical nodes.
3) Greedy-R: Select k nodes that have the greatest impact on the network robustness metric R after deletion, i.e., the node with the greatest Fitness(), as the crucial nodes.
4) TDE-Random: The TDE framework proposed in the article searches for k critical nodes, but a random encoding is used to design individual genotypes. No network topology information is fused in the encoding process, so TDE-random can be regarded as a method to handle nodes entirely discretely.
5) TDE-Degree: Adopt the TDE framework proposed in the article to search k critical nodes, and individual genotype was designed with node degree information. In this way, the correlation among nodes is fused in the evolutionary search process.

The parameters in TDE-random and TDE-degree are set to the same to ensure comparability. Table I shows the numerical results of four methods on CND. Fig. 6 shows the convergence curves of TDE-random and TDE-degree for comparing the convergence efficiency of the evolution-based methods. The curves in the figure represent the fitness value of the best individual with the iteration of the population. The faster the curve rises, the higher the convergence efficiency. The numerical results and the convergence curve are obtained by taking the mean of the ten times experimental results due to the randomness of the evolutionary algorithm.

As shown in Table I, the evolution-based approach performs better than greedy-degree, greedy-R, and random; greedy-R does not perform as well as greedy-degree on the four networks. The main reason is that greedy-R tends to select a single node that significantly impacts network connectivity,

TABLE I

FITNESS OF ALGORITHMS ON CND. R_0 IS THE ORIGINAL VALUE OF ROBUSTNESS INDEX R, AND k IS THE NUMBER OF CRITICAL NODES

dataset	R_0	k	algorithms	Fitness	Fitness/R_o
RG	0.401951	30	Random	0.022637	5.63%
			Greedy-degree	0.023044	5.73%
			Greedy-R	0.017137	4.26%
			TDE-random	**0.030490**	**7.59%**
			TDE-degree	**0.030072**	**7.48%**
ER	0.352920	30	Random	0.022088	6.26%
			Greedy-degree	0.034292	9.72%
			Greedy-R	0.017343	4.91%
			TDE-random	0.036617	10.38%
			TDE-degree	**0.036692**	**10.40%**
WS	0.382932	30	Random	0.023987	6.26%
			Greedy-degree	0.027514	7.20%
			Greedy-R	0.026151	6.82%
			TDE-random	0.037415	9.77%
			TDE-degree	**0.037678**	**9.84%**
BA	0.257781	30	Random	0.023774	9.22%
			Greedy-degree	0.056735	22.01%
			Greedy-R	0.046849	18.17%
			TDE-random	0.055821	21.65%
			TDE-degree	**0.058130**	**22.55%**

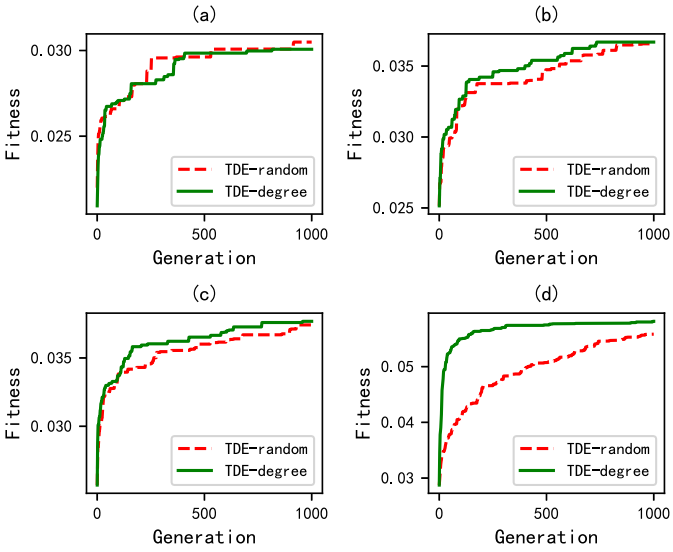


Fig. 6. Convergence curve of the evolution-based methods on CND with the increase in generation on synthetic networks.

such as the hub between connected block nodes, and such nodes do not have a high degree in the network. From the k-core decomposition theory in [37], it can be seen that the set of core nodes that have a significant impact on the network is connected to about 70% or more of the core nodes, i.e., such nodes usually have a relatively large degree, and removing a single such node does not have a significant impact on the network connectivity. Therefore, greedy-R is usually worse than greedy-degree. On RG and ER networks with more uniform degree distributions, the random strategy has a higher probability of selecting nodes at the network's core than greedy-R due to the property that greedy-R prefers hub nodes. Thus, its performance is slightly more robust than that of greedy-R. While the performance of TDE-random and TDE-degree is similar, TDE-degree shows a slight advantage. However, Fig. 6, which shows the convergence efficiency

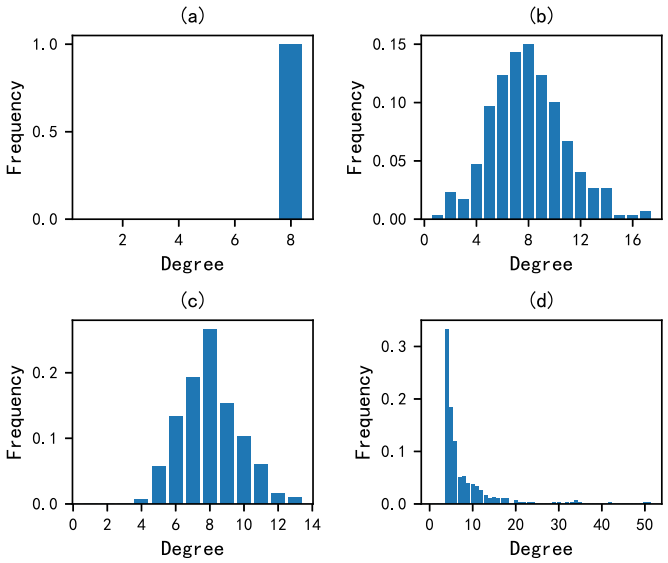


Fig. 7. Degree distribution of four synthetic networks. (a) RG. (b) ER. (c) WS. (d) BA.

of the methods, demonstrates different trends of the two evolution-based methods in different types of networks: TDE-degree shows no advantage in the RG network, a slight advantage in the WS and ES networks, and an obvious advantage in the BA network. We analyze the above experimental results from the perspective of degree value, and Fig. 7 shows the degree distribution of the four networks.

In the RG network, two evolutionary algorithms have similar performance. That is because nodes in the RG network process the same degree value [see Fig. 7(a)], leading to the nodes' representations combining degree-based topology information being consistent, introducing no more additional discriminative information. As a result, TDE-degree has a similar performance with TDE-random in the RG network. Similar phenomena can also be found in the ER and WS networks for their degree value is limited in the vicinity of the average degree in distribution [see Fig. 7(b) and (c)]. Contrarily, TDE-degree algorithm performance achieves significant improvement in the BA network. That is because the nodes in the BA network follow power-law distribution [see Fig. 7(d)], and the high heterogeneity in degree values could introduce discriminative information effectively into the algorithm.

From another perspective, evolution search with topology information fusing can be regarded as search with preference as mentioned in Section III, and nodes with larger degree values will be searched with higher probability in TDE-degree. Such preference-based search method for high-degree nodes can get better performance in heterogeneity network, e.g., BA network, but have poor improvement on homogeneity network, e.g., WS- and ER-like networks, and even may reduce the search capability for low attention on nodes with low degree relatively.

B. Analysis on Solution Space

To verify the obtained experimental results further, we analyze it from the perspective of solution space. Fig. 8 shows the

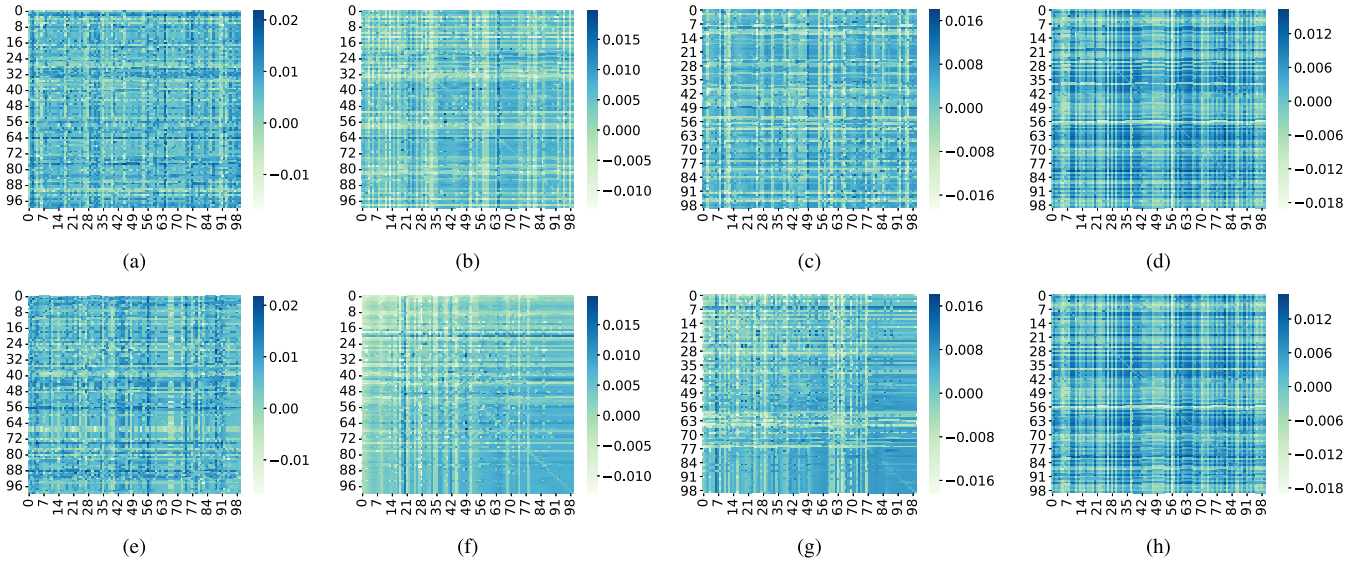


Fig. 8. Visual solution space before and after fusing degree information on four synthetic networks. (a) Unsorted_RG. (b) Unsorted_ER. (c) Unsorted_WS. (d) Unsorted_BA. (e) Sorted_RG. (f) Sorted_ER. (g) Sorted_WS. (h) Sorted_BA.

solution space of CND on four networks, that is, the solution space of select two nodes as critical nodes from the ER, RG, ES, and BA networks, respectively. To reduce the size of solution space and make the visualization of results more straightforward, we reconstruct four networks with 100 nodes. Identically, the abscissa and ordinate represent a candidate critical node, respectively, where the color depth represents the fitness value of the candidate solution.

Fig. 8(a)–(d) shows the unsorted solution space of CND, and it can be seen that the solution space without topology information combining is chaotic and irregular. While in Fig. 8(e)–(h), the solution spaces after degree-based topology information being involved show different features in four types of networks. In the RG network, the sorted solution space with degree information is not ordered, and it is as chaotic as the original one. That is because all the nodes in this network have the same degree value which leads to no trend of diagonal increasing, and block distribution is represented in the whole solution space. The efficiency of the algorithm is not improved in this situation. While in the ER and WS networks, due to the more various degree values, the orderliness of sorted solution space appears, as well as the trend of diagonal increasing. As for block distribution, the blocks increase in number and reduce in area. As a result, the efficiency of the algorithm gets a limited improvement. In the BA network, the significant variance of node degree values makes the sorted solution space much more orderly, and the trend of diagonal increasing is more obvious than that in the ER and WS networks. In this situation, the search efficiency of the algorithm gets a greater improvement as shown in Fig. 6.

From the experimental results and the visualized solution space above, it can be seen that the improvement in algorithm efficiency is related positively to the orderliness of sorted solution space with degree information. It is consistent with the reasoning in Section III.

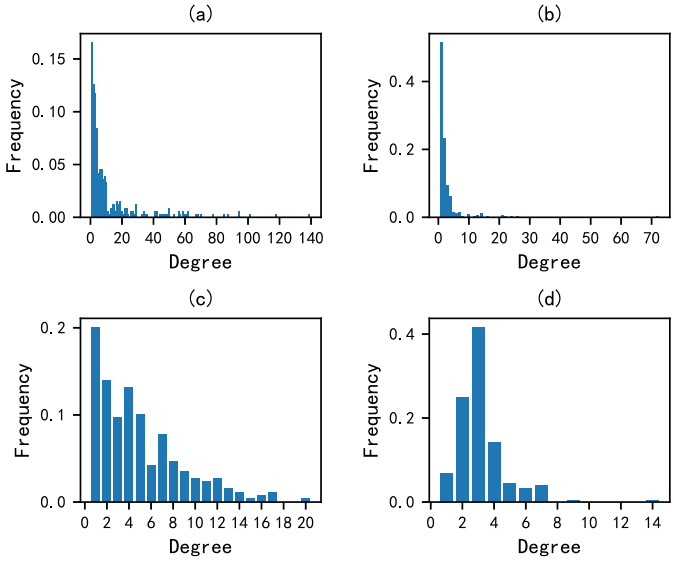


Fig. 9. Degree distribution of four real networks. (a) USAir97. (b) Ecoli. (c) A01. (d) Circuit.

C. Experiments on Real Datasets

To further verify the effectiveness of our proposed method, we analyze the effectiveness of the algorithm on real networks with different node degree distributions. Fig. 9 shows the degree distribution of four real networks, where USAir97 is a traffic network with 332 nodes and 2126 edges, Ecoli is a biological network with 328 nodes and 456 edges, A01 is a citation network with 259 nodes and 640 edges, and Circuit is a circuit network with 252 nodes and 399 edges. The convergence curve of the algorithm before and after fusing node degree information is shown in Fig. 10. It can be seen from the result that in the heterogeneous networks (i.e., USAir97, Ecoli, and A01), TDE-degree performances are better than TDE-random. In addition, the number of iterations required for TDE-degree convergence is much smaller than

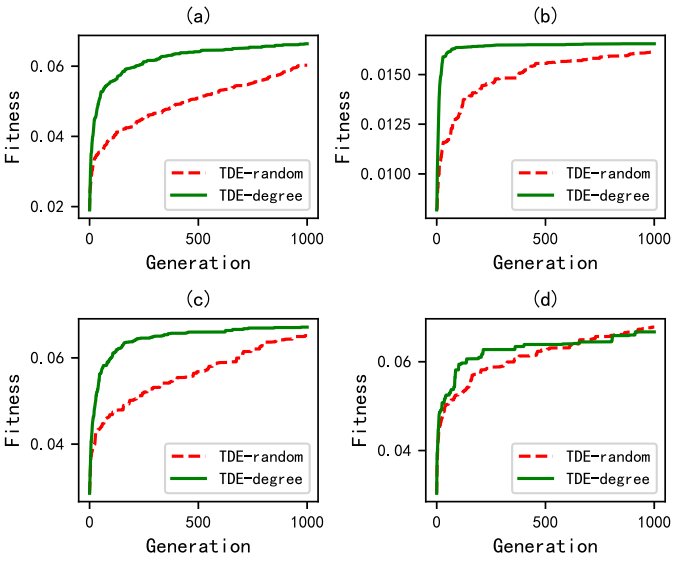


Fig. 10. Convergence curve of the evolution-based methods on CND with the increase in generation on real networks. (a) USAir97. (b) Ecoli. (c) A01. (d) Circuit.

that of TDE-random. Due to the random encoding property of TDE-random, the number of iterations needed for convergence is similar to that of the traditional DE. Compared with the number of iterations, the time required for the decoding operation in the TDE method is insignificant. Therefore, TDE-degree has performance improvement compared with the traditional DE. However, in homogeneous networks (i.e., Circuit), fusing degree information cannot improve the search capability of the evolution algorithm. These results are consistent with the former analysis.

V. Conclusion

In this work, we proposed a TDE framework for CND called TDE, which consists of two parts, i.e., the topology-combined genotype design module for fusing topology information and the DE module for identifying critical nodes. The topology combined with the genotype design module achieves topological information fusion by mapping-based encoding and similarity-based decoding. In addition, in the DE module, compared with higher order topological information such as node embedding, this article adopts the node degree that is more applicable to the CND problem to design individual genotypes, and the suitable variation operator and selection operator are designed to use the topological network information effectively. The proposed approach improves the performance of the evolutionary algorithm by designing the genotype with network topology information. Experiments on the synthetic and real datasets show it is feasible to improve the search efficiency of the algorithm by fusing network topology information. Our study highlights the importance of the underlying network topology information and provides insights into the design of the elements of algorithms to further improve the performance on CND.

References

[1] R. Albert, H. Jeong, and A.-L. Barabási, "Error and attack tolerance of complex networks," *Nature*, vol. 406, no. 6794, pp. 378–382, Jul. 2000.

[2] R. Aringhieri, A. Grosso, P. Hosteins, and R. Scatamacchia, "A general evolutionary framework for different classes of critical node problems," *Eng. Appl. Artif. Intell.*, vol. 55, pp. 128–145, Oct. 2016.

[3] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, Oct. 1999.

[4] A.-L. Barabási and E. Bonabeau, "Scale-free networks," *Sci. Amer.*, vol. 288, no. 5, pp. 60–69, May 2003.

[5] A. K. Bhowmick, M. Gueuning, J.-C. Delvenne, R. Lambiotte, and B. Mitra, "Temporal sequence of retweets help to detect influential nodes in social networks," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 3, pp. 441–455, Jun. 2019.

[6] P. Bovy and R. Thijs, *Modelling for Transportation Systems Planning*. Delft, The Netherlands: DUP Satellite, 2002, pp. 3–10.

[7] W. Chen, M. Jiang, C. Jiang, and J. Zhang, "Critical node detection problem for complex network in undirected weighted networks," *Phys. A, Stat. Mech. Appl.*, vol. 538, Jan. 2020, Art. no. 122862.

[8] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[9] M. Di Summa, A. Grosso, and M. Locatelli, "Branch and cut algorithms for detecting critical nodes in undirected graphs," *Comput. Optim. Appl.*, vol. 53, no. 3, pp. 649–680, Dec. 2012.

[10] P. Erdös and A. Rényi, "On random graphs I," *Publicationes Mathematicae Debrecen*, vol. 6, nos. 290–297, p. 18, 1959.

[11] C. Fu et al., "A novel spatiotemporal behavior-enabled random walk strategy on online social platforms," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 3, pp. 807–817, Jun. 2022.

[12] C. Fu et al., "Link weight prediction using supervised learning methods and its application to Yelp layered network," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 8, pp. 1507–1518, Aug. 2018.

[13] C. Fu, Y. Zheng, Y. Liu, Q. Xuan, and G. Chen, "NES-TL: Network embedding similarity-based transfer learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 1607–1618, Jul. 2020.

[14] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "Networks formed from interdependent networks," *Nature Phys.*, vol. 8, no. 1, pp. 40–48, 2012.

[15] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 855–864.

[16] P. Hansen, N. Mladenović, and J. A. M. Pérez, "Variable neighbourhood search: Methods and applications," *Ann. Oper. Res.*, vol. 175, no. 1, pp. 367–407, 2010.

[17] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han, "Attack vulnerability of complex networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 65, no. 1, 2002, Art. no. 056109.

[18] M. M. D. Khomami, A. Rezvanian, M. R. Meybodi, and A. Bagheri, "CFIN: A community-based algorithm for finding influential nodes in complex social networks," *J. Supercomput.*, vol. 77, no. 3, pp. 2207–2236, Mar. 2021.

[19] M. Lalou, M. A. Tahraoui, and H. Kheddouci, "The critical node detection problem in networks: A survey," *Comput. Sci. Rev.*, vol. 28, pp. 92–117, May 2018.

[20] J. M. Lewis and M. Yannakakis, "The node-deletion problem for hereditary properties is NP-complete," *J. Comput. Syst. Sci.*, vol. 20, no. 2, pp. 219–230, Apr. 1980.

[21] A. Y. Lokhov, M. Mézard, H. Ohta, and L. Zdeborová, "Inferring the origin of an epidemic with a dynamic message-passing algorithm," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 90, no. 1, Jul. 2014, Art. no. 012801.

[22] H. R. Lourenço, O. C. Martin, and T. Stützle, "Iterated local search: Framework and applications," in *Handbook of Metaheuristics*. Cham, Switzerland: Springer, 2019, pp. 129–168.

[23] L. Ma, S. Cheng, and Y. Shi, "Enhancing learning efficiency of brain storm optimization via orthogonal learning design," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 11, pp. 6723–6742, Nov. 2021.

[24] L. Ma et al., "Learning to optimize: Reference vector reinforcement learning adaption to constrained many-objective optimization of industrial copper burdening system," *IEEE Trans. Cybern.*, early access, Jul. 14, 2021, doi: 10.1109/TCYB.2021.3086501.

[25] W. Pullan, "Heuristic identification of critical nodes in sparse real-world graphs," *J. Heuristics*, vol. 21, no. 5, pp. 577–598, Oct. 2015.

[26] L. Qiu, X. Tian, J. Zhang, C. Gu, and S. Sai, "LIDDE: A differential evolution algorithm based on local-influence-descending search strategy for influence maximization in social networks," *J. Netw. Comput. Appl.*, vol. 178, Mar. 2021, Art. no. 102973.

[27] J. Rezaei, F. Zare-Mirakabad, S. A. MirHassani, and S.-A. Marashi, "EIA-CNDP: An exact iterative algorithm for critical node detection problem," *Comput. Oper. Res.*, vol. 127, Mar. 2021, Art. no. 105138.
[28] C. M. Schneider, A. A. Moreira, J. S. Andrade, Jr., S. Havlin, and H. J. Herrmann, "Mitigation of malicious attacks on networks," *Proc. Nat. Acad. Sci. USA*, vol. 108, no. 10, pp. 3838–3841, 2011.
[29] S. Shen and J. C. Smith, "Polynomial-time algorithms for solving a class of critical node problems on trees and series-parallel graphs," *Networks*, vol. 60, no. 2, pp. 103–119, Sep. 2012.
[30] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[31] Y. Su, K. Zhou, X. Zhang, R. Cheng, and C. Zheng, "A parallel multi-objective evolutionary algorithm for community detection in large-scale complex networks," *Inf. Sci.*, vol. 576, pp. 374–392, Oct. 2021.
[32] M. Ventresca, "Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem," *Comput. Oper. Res.*, vol. 39, no. 11, pp. 2763–2775, Nov. 2012.
[33] M. Ventresca and D. Aleman, "A fast greedy algorithm for the critical node detection problem," in *Proc. Int. Conf. Combinat. Optim. Appl.* Cham, Switzerland: Springer, 2014, pp. 603–612.
[34] M. Ventresca and D. Aleman, "A randomized algorithm with local search for containment of pandemic disease spread," *Comput. Oper. Res.*, vol. 48, pp. 11–19, Aug. 2014.
[35] A. Veremyev, V. Boginski, and E. L. Pasiliao, "Exact identification of critical nodes in sparse networks via new compact formulations," *Optim. Lett.*, vol. 8, no. 4, pp. 1245–1259, Apr. 2014.
[36] B. Vitoriano, M. Ortuño, G. Tirado, and J. Montero, "A multi-criteria optimization model for humanitarian aid distribution," *J. Global Optim.*, vol. 51, no. 2, pp. 189–208, Oct. 2011.
[37] X.-F. Wang, X. Li, and G.-R. Chen, *Network Science: An Introduction*, vol. 4. Beijing, China: Higher Educ. Press, 2012, pp. 95–142.
[38] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.
[39] Q. Xuan et al., "Subgraph networks with application to structural feature space expansion," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2776–2789, Jun. 2019.
[40] X. Yang and F. Xiao, "An improved gravity model to identify influential nodes in complex networks based on k-shell method," *Knowl.-Based Syst.*, vol. 227, Sep. 2021, Art. no. 107198.
[41] S. Yu et al., "Target defense against link-prediction-based attacks via evolutionary perturbations," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 2, pp. 754–767, Feb. 2021.
[42] J. Zhou, J. Shen, S. Yu, G. Chen, and Q. Xuan, "M-Evolve: Structural-mapping-based data augmentation for graph classification," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 190–200, Jan. 2021.
[43] Y. Zhou, J.-K. Hao, Z.-H. Fu, Z. Wang, and X. Lai, "Variable population memetic search: A case study on the critical node problem," *IEEE Trans. Evol. Comput.*, vol. 25, no. 1, pp. 187–200, Feb. 2021.
[44] Y. Zhou, J.-K. Hao, and F. Glover, "Memetic search for identifying critical nodes in sparse graphs," *IEEE Trans. Cybern.*, vol. 49, no. 10, pp. 3699–3712, Oct. 2019.

Shanqing Yu received the M.S. and Ph.D. degrees from the Graduate School of Information, Production and Systems, Waseda University, Tokyo, Japan, in 2008 and 2011, respectively.

She is currently a Lecturer with the Institute of Cyberspace Security and the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. Her research interests cover intelligent computation and data mining.

Yongqi Wang received the B.S. and M.S. degrees from the Zhejiang University of Technology, Zhejiang, China, in 2019 and 2022, respectively.

His research interests are complex networks and intelligent computing.

Jiaxiang Li received the B.S. degree from the Henan University of Science and Technology, Luoyang, China, in 2021. He is currently pursuing the M.S. degree with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China.

His current research interests include social network analysis and evolutionary computing.

Xu Fang received the B.S. degree from the Civil Aviation University of China, Tianjin, China, in 2014. He is currently pursuing the M.S. degree with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China.

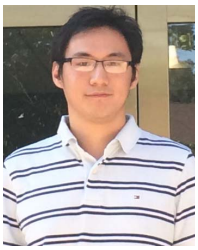
His current research interests include social network analysis and machine learning.

Jinyin Chen received the B.S. and Ph.D. degrees from the Zhejiang University of Technology, Hangzhou, China, in 2004 and 2009, respectively. She studied evolutionary computing at the Ashikaga Institute of Technology, Ashikaga, Japan, in 2005 and 2006.

She is currently a Professor with the Zhejiang University of Technology. Her research interests include artificial intelligence security, graph data mining, and evolutionary computing.

Ziwan Zheng received the B.S. degree in geographic information science (GIS) from Zhejiang University, Hangzhou, China, in 2016.

He currently works with the Zhejiang Police College, Hangzhou, as a Lecturer. His research interests include GIS and policing.



Chenbo Fu (Member, IEEE) received the B.S. degree in physics from the Zhejiang University of Technology, Hangzhou, China, in 2007, and the M.S. and Ph.D. degrees in physics from Zhejiang University, Hangzhou, in 2009 and 2013, respectively.

He was a Post-Doctoral Researcher with the College of Information Engineering, Zhejiang University of Technology. He was a Visiting Scholar with the Department of Computer Science, University of California at Davis, Davis, CA, USA, in 2014. In 2017, he was a Research Assistant with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, SAR, China. He is currently a Lecturer with the Institute of Cyberspace Security, Zhejiang University of Technology. His research interests include network-based algorithm design, social networks, data mining, chaos synchronization, network dynamics, and machine learning.