# GA-Based Multipopulation Synergistic Gene Screening Strategy on Critical Nodes Detection

Shanqing Yu, Jiaxiang Li, Xu Fang, Yongqi Wang, Jinhuan Wang, Qi Xuan, *Member, IEEE*, and Chenbo Fu, *Member, IEEE*

***Abstract*— Critical node detection (CND) is commonly used to detect nodes with a high impact on network robustness. It has been widely used in disease propagation, social networks, communications, and other fields. As a nondeterministic polynomial-time (NP)-complete problem, the efficiency of solving CND severely limits the scale of the available network. Fortunately, the evolutionary algorithm (EA) is an effective method to solve this problem. However, although EA improves the global search capability of the algorithm by preserving gene diversity, it also introduces many inferior genes, thus expanding the candidate solution space, reducing the search efficiency, and making it difficult to apply the pruning algorithm directly to its solution space. Hence, indirectly reducing the solution space of EA by deleting inferior genes is a feasible pruning method; however, the interaction of multiple genes affects the quality of CND solutions, making it a challenge to pick out inferior individual genes. Therefore, this work proposes a multipopulation synergistic gene screening algorithm based on the parallelism of EA and combined with Ensemble learning for identifying low-quality genes and removing them as a way of pruning the solution space of the algorithm and improving the search efficiency. The algorithm encodes all nodes in the graph as the gene pool of EA and treats a single population as a weak learner to screen the dominant genes in the gene pool and achieve fast pruning of EA's solution space by integrating the dominant individuals in multiple populations. In this work, the experiments demonstrate the effectiveness of the proposed method and analyze the effect of different network structures on the algorithm.**

***Index Terms*— Critical nodes detection (CND), ensemble learning, genetic algorithms (GAs), multipopulations.**

## I. Introduction

CRITICAL nodes detection is one of the most important research topics in complex networks [1], [2], [3], [4]. As a combinatorial optimization problem, the main objective is to find a set of nodes from the network whose deletion will lead to the most significant disruption of network connectivity. For example, in biological networks [5], locating the critical nodes can be used to guide the treatment of various diseases [6]. In propagation networks such as disease [7] and information spreading [8], [9], controlling critical nodes can effectively reduce the rate and scope of diffusion. In communication networks [10], [11], losses due to accidents or deliberate attacks can be reduced by adding protection measures to critical nodes.

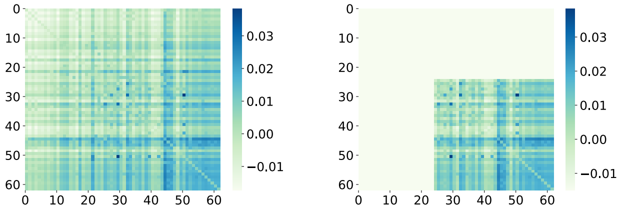
Meanwhile, CND, as a nondeterministic polynomial-time (NP)-complete problem, has a very high computational complexity [12], especially in large-scale networks with multiple critical nodes. Two strategies are often used to solve such problems, i.e., strategies based on greedy and EA. Algorithms using greedy strategies [2], [13], [14], [15], [16], [17] can find acceptable approximate optimal solutions. Due to its powerful global search capability, EA [18], [19], [20], [21], [22], [23], [24] perform even better on this problem, i.e., using the natural selection mechanism of superiority and inferiority to explore and mine superior individuals in the solution space. However, the efficiency of most approaches is still limited by the size of the target network and the number of critical nodes.

Pruning algorithms [25], [26], [27], [28], [29], [30] are helpful methods in search algorithms to avoid unnecessary traversal processes and improve the time and space efficiency of the algorithm. Algorithms optimized by pruning and other optimization strategies are more efficient in execution than the general unpruned algorithms. However, preprocessing the algorithm solution space by pruning methods has yet to be studied when using evolutionary computation to solve CND problems. The problem's difficulty lies in the large number of individuals constituting the solution space, which makes it challenging to remove inferior individuals within an acceptable time. Since the quality of CND solutions is influenced by the interaction of individual genes in the EA, pruning can be achieved by screening genes. Fig. 1 shows a comparative example of the pruned solution space of evolutionary algorithm (EA) with and without simple pruning by removing low degree nodes when searching for two critical nodes in the dolphin network. The horizontal and vertical coordinates represent the nodes sorted by node degree from smallest to largest. The color bar indicates the network connectivity, and the darker the color, the worse the network connectivity after removing the group of nodes. As shown in Fig. 1, it can be seen that after deleting nodes with low degree values, the

Manuscript received 13 February 2023; revised 27 April 2023 and 15 June 2023; accepted 9 July 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62103374, in part by the Key Research and Development Projects in Zhejiang Province under Grant 2021C01117, in part by the National Key Research and Development Program of China under Grant 2018AAA0100800, in part by the Key Research and Development Program of Zhejiang under Grant 2022C01018, in part by the National Natural Science Foundation of China under Grant U21B2001 and Grant 61973273, and in part by the National Key Research and Development Program of China under Grant 2020YFB1006104. *(Corresponding author: Chenbo Fu.)*

The authors are with the Institute of Cyberspace Security, Zhejiang University of Technology, Hangzhou 31000, China (e-mail: yushanqing@zjut.edu.cn; jiaxiangli.zjut@gmail.com; henryfangx@gmail.com; 2111903062@zjut.edu.cn; jhwang@zjut.edu.cn; xuanqi@zjut.edu.cn; cbfu@zjut.edu.cn).

Digital Object Identifier 10.1109/TCSS.2023.3325263



(a) (b)

Fig. 1. Simple illustration of solution space pruning. (a) Solution space before pruning. (b) Solution space after pruning.

significantly reduced candidate solution space can improve the efficiency of searching for the optimal solution. However, CND is a combinatorial optimization problem where multiple nodes interact to get the final result, which means simple and brutal deletion may also ignore the potential high-quality solutions. Therefore, it is necessary to develop a method that can identify the merits of gene nodes and prune them quickly and efficiently to reduce the candidate space and consequently improve the efficiency of the EA for CND.

To address this issue, a pruning strategy was designed by combining ensemble learning [31], [32], [33], [34] and EA. This strategy significantly reduces the number of individuals who include inferior genes in the enormous solution space by eliminating inferior genes from the limited gene pool of EA, thus improving the algorithm search efficiency. Ensemble Learning combines multiple learning algorithms to obtain a more robust generalization performance than a single learner. Furthermore, genetic algorithm (GA) [35], [36], [37], [38], [39] was chosen to implement this algorithm due to its robustness and simplicity, and its parallelism capability is well suited for implementing ensemble learning. The proposed pruning algorithm encodes network nodes as a gene pool of GA, which is then optimized using multiple populations. The algorithm rapidly prunes the gene pool through a small number of iterations, thereby improving the efficiency of the subsequent GA-based CND algorithm. The experiment results on real and artificial network datasets demonstrate the method's effectiveness. The main contributions of this article are as follows.

1) In this work, the solution space pruning problem for the GA-based CND algorithm is reduced to the selection of genes. A multipopulation synergistic pruning algorithm is proposed to solve the problem in combination with Ensemble learning by exploiting the potential parallelism of GA. The algorithm simultaneously considers the interactions between multiple genes. It retains the dominant genes as much as possible while eliminating the inferior genes, rapidly decreasing this problem's solution space. The efficiency of GA-based CND is improved.
2) The effectiveness of the optimization performance of the proposed method is verified by comparative experiments on multiple artificial networks and natural networks. Furthermore, the effectiveness of the proposed method under different network structures is also analyzed.

The rest of this article is organized as follows. Section II reviews related work on critical nodes detection and pruning strategies; Section III presents the solution space pruning strategy proposed in this article; Section IV shows the related experiments and results are concluded in Section V.

## II. Related Work

### A. CND Algorithm

CND is an optimization problem for finding a set of nodes that highly impact the network's connectivity. One of the most popular and practical approaches to solving this problem is to use greedy algorithms, which iteratively make locally optimal choices to approximate the optimal global solution. For the CND problem, greedy strategies based on network metrics such as degree [40], *k*-shell [14] and centrality [41] can approximate the critical nodes but are accompanied by significant losses. Therefore, many researchers have combined network structure information in their work and proposed new network metrics to solve the CND problem using greedy algorithms. For example, Chen et al. [15] proposed a new metric to evaluate network fragmentation, formulated a new nonconvex mixed integer quadratic programming model and used a greedy algorithm to solve that optimization problem. Khomami et al. [16] proposed the community finding influential node (CFIN) algorithm for detecting influential nodes in the network combining the community structure and used a greedy algorithm to optimize the objective function in local search.

In some cases, even if the chosen metric is appropriate, the greedy algorithm suffers from a loss of accuracy. Because the CND problem is a combinatorial optimization problem, the greedy algorithm does not consider the coupling information between the individual solutions in the solving process. EAs are considered standard algorithms for solving combinatorial optimization problems due to their global search capability, and several evolutionary-based algorithms have been proposed for solving CND. For example, Yu et al. [24] proposed a differential evolution framework incorporating network topology information in dealing with the complex network CND problem. The method combines topology information into the genotype design of the differential evolution algorithm to improve the algorithm search efficiency. Xu and Guo [20] proposed a membrane EA (MEA-CNDP) to solve the CND problem for biological targets. Qiu et al. [21] proposed a LIDDE algorithm based on the differential EA (DEA) for solving the influence maximization problem. Zhou et al. [42] proposed a variable population modal algorithm with better performance on the CND problem. Wu et al. [22] proposed a genetic importance based EA (GIEA) to identify a set of critical nodes in a cyber-physical anion power system (CPPS) by maximizing the total load loss received by the end-user. In addition, the multiobjective EA (MOEA) [43], [44], [45], [46] is of high practical value when dealing with combinatorial optimization problems. It can satisfy multiple objective functions while finding a set of solutions. For example, Eliézer and Gaskó [19] used MOEA for CND and proposed three different initialization strategies to improve the performance of MOEA. In addition, swarm intelligence algorithms [47] and machine learning-based methods [48] have also been used to solve the

CND problem. These methods are effective for solving CND problems. However, none of these approaches considers using pruning strategies to improve the algorithm's efficiency.

### *B. Pruning Strategy*

The pruning strategy can improve search algorithms' efficiency by avoiding unnecessary traversal processes. For example, when faced with the verification problem of designing digital systems, Chen et al. [49] proposed a state pruning method, using multiple heuristics to make the depth-first-search (DFS) select the most efficient path for searching. Merz and Sanders [50] proposed a data structure named PReaCH for the reachability query problem in directed graphs, which improves improve the existing search pruning technique by collecting more information from a single DFS traversal. PReaCH-indices significantly outperform previous data structures in query speed with comparable preprocessing costs. In solving the satellite transmission scheduling problem, which is also an NP-complete problem, Zhao et al. [26] developed an optimization model for satellite data transmission scheduling. They proposed three pruning strategies to improve the local search algorithm. Qin et al. [27] proposed a semantic hash based on graph neural networks (GHashing) for implementing approximate pruning to improve the query speed of graph databases when dealing with the graph edit distance problem on large-scale graph databases. Mahesh and Sushnigdha [28] proposed a new meta-heuristic-based optimization technique called the search space reduction (SSR) optimization algorithm. Unlike existing optimization algorithms, this algorithm attempts to solve the problem of getting trapped in a local optimal by randomly generating search agents in each iteration instead of following the best search agent. Moreover, the search space decreases with iterations to ensure the exploration capability of the algorithm.

Algorithms, such as EAs and swarm intelligence optimization algorithms, are effective methods for solving NP-Complete problems. However, the computational efficiency still needs to be improved, and reducing the solution space of candidate solutions is one of the effective ways to improve computational efficiency. Silva et al. [29] proposed a method based on the mesh analysis technique to reduce the solution space of the distribution system reconfiguration problem, followed by the use of metaheuristic particle swarm optimization (PSO) to solve this problem. Kadu et al. [51] proposed a method based on a critical path to reduce the solution space of the GA in solving the problem of efficiently searching for the best solution for the water distribution network. Li et al. [52] proposed a dynamic bit-masking strategy that progressively reduces the search space during the evolution process when using PSO algorithms for feature selection (FS), significantly reducing computational time.

In addition, pruning strategies are often used to reduce the model structure and parameters in machine learning models, thus reducing the training cost of the model and the possibility of model overfitting. For example, Li et al. [30] defined channel pruning on neural networks as the problem of searching for the optimal channel configuration and compared the performance of different currently available pruning methods with random pruning methods. Kwon et al. [53] implemented a sparse quantized neural network weight representation scheme using fine-grained unstructured pruning that allows for the compression and representation of various deep learning models; Roy et al. [54] proposed a method to dynamically prune and train the model at the same time substantially reducing the computational effort in the network training process; Zhang et al. [55] encoded filter sets as genotypes and automatically performed filter selection in the search space by MOEAs. As can be seen above, the pruning strategy significantly reduces the computational complexity of the algorithm and improves its efficiency.

## III. Method

In this work, a multipopulation synergistic gene screening strategy incorporating Ensemble Learning is proposed by exploiting the potential parallelism of GA. In this strategy, each population is considered as a weak learner that independently samples the gene pool and votes on the genes by their frequency of occurrence on the dominant individuals of the population. Finally, the gene pool is censored by integrating the voting results of multiple populations. Low-voting genes are removed, while high-voting genes are retained. After a small number of iterations, the solution space of GA will be significantly censored, thus improving the efficiency of optimal search solutions. The proposed method is described in detail in this section.

### *A. Problem Definition*

The CND problem is to find a set of nodes that have the most significant impact on network connectivity. Therefore, this work uses pairwise connectivity $\mathrm{PC}(G)$ to measure the connectivity of the remaining network after a group of nodes is removed, which reflects the importance of this group of nodes. The lower the pairwise connectivity of the remaining network after the group of nodes is removed, the more critical the group of nodes.

Thus, the CND problem and the solution space pruning problem can be defined as follows: given an integer $k$, and an undirected unweighted network $G(V)$, $V$ is the set of nodes in the network, and also the gene pool of the GA. The pruning strategy aims to quickly obtain a set of a smaller number of nodes $V^- \in V$ such that CND can find a set of nodes $\hat{V} \in V^-$ and satisfy the following equation:

$$\begin{aligned}
&\arg\max \mathrm{Fitness}(G, \hat{V}) = e^{-\mathrm{PC}(\hat{G})}\\
&\text{s.t. } \hat{G} = G - \hat{V}\\
&\qquad |\hat{V}| \le k\\
&\qquad \mathrm{PC}(\hat{G}) = \sum_{C_i \in \hat{G}} \frac{\delta_i(\delta_i - 1)}{2}
\end{aligned} \tag{1}$$

where $k$ is the number of critical nodes and $\hat{G}$ is the remaining network after removing $\hat{V}$ from the original network. $\mathrm{Fitness}(G, \hat{V})$ is the fitness value of the set of nodes $\hat{V}$. In both the pruning module and the CND module in this article, $\mathrm{Fitness}(G, \hat{V})$ is used to evaluate the importance of a set of nodes $\hat{V}$. $\mathrm{PC}(\hat{G})$ is the pairwise connectivity of the network $\hat{G}$; $C_i$ is the connectivity slice in the network $\hat{G}$, and $\delta_i$ is the size of the connectivity component $C_i$, i.e., the number of nodes in this component. From the above equation, $\mathrm{PC}(\hat{G})$ is related to the number of connected tiles and the size of connected tiles, and the smaller the value, the lower the degree of connectivity of the network. In addition, maximizing the fitness function yields a set of critical nodes $\hat{V}^{\text{critical}} = \{\text{node}_1^{\text{critical}}, \text{node}_2^{\text{critical}}, \ldots, \text{node}_k^{\text{critical}}\}$. Furthermore, $V^-$ is obtained by the multipopulation synergistic gene screening module.

### *B. Multipopulation Synergistic Gene Screening Module*

GA is commonly used to solve combinatorial optimization problems, and its primary model was first proposed by Holland [56]. GA mainly uses natural selection mechanisms to solve optimization problems. In this work, a set of nodes is considered as an individual, with each node representing a gene on the individual. Offspring are then generated based on the designed genetic operators, such as crossover and mutation. Individuals with better fitness have a better chance of surviving and reproducing, thus allowing the population to evolve.

Our module also combines ensemble learning. A single population is considered a weak learner for learning the importance of nodes in the network. Furthermore, the importance of nodes in the network is voted by integrating the learning results of multiple populations. Then this is used to guide the censoring of the gene pool. The censored gene pool constituting the set of nodes $V^-$ is then passed into the post gene screened CND module for finding a set of critical nodes. The flow of this module is shown in Fig. 2. It consists of four parts: multipopulation initialization, genetic operations, population-dominant individual selection, and integration of multiindividuals gene screening. First, multipopulation initialization constructs multiple independent sets of individuals by gene sampling; after the genetic operation, the overall quality of the population is improved, and then the population dominant individual selection will obtain the highest quality individuals from each population. Integrated multiindividual gene screening will be based on the dominant individuals selected from multiple populations to form a dominant gene set, and based on the number of times the genes in the gene pool appear in this dominant gene set, the quality of genes is voted as a guide to eliminating the inferior solutions in the solution space. The pruning of the solution space of this problem can be achieved after several iterations of the above steps. These four components are described in detail below.

1) *Multipopulation Initialization:* Each population construction in multipopulation initialization relies on the current gene pool. In contrast, the initial gene pool needs to be constructed based on the set of nodes of the original network $G(V)$, when the algorithm is first executed. The initial gene pool can be expressed as follows:

$$\text{genePool}^0 = V = \{v | v_i \in V, i = 1, 2, 3, \ldots, |V|\}. \tag{2}$$

Each individual in the populations is a collection of critical nodes selected from the current gene pool and represents a candidate solution to the CND problem. An individual in the population can be represented

$$\text{inds}_{yx}^m = \{v_x^m | v_{xi}^m \in \text{genePool}^m i = 1, 2, \ldots, k\} \tag{3}$$

where $\text{inds}_{yx}^m$ is used to denote the $x$th individual in the $y$th population, $v_{xi}^m$ denotes the $i$th gene in the individual, $\text{genePool}^m$ is the current gene pool at the $m$th iteration, and $k$ is the preset number of critical nodes. Multiple population initialization is the process of generating multiple populations, where each population consists of multiple individuals. The generation strategy used in this article is random, and the $y$th population of $m$th iteration can be expressed as follows:

$$\text{population}_y^m = \left\{\text{inds}_{y1}^m, \text{inds}_{y2}^m, \ldots, \text{inds}_{ypopsize}^m\right\} \tag{4}$$

where $\text{inds}_{y1}^m$ is the first individual in $\text{population}_y^m$, and pop_size is the predetermined number of individuals in a single population. The multiple populations generated based on the current $m$th generation gene pool, i.e., the set of multiple above populations can be expressed by

$$\text{Pops}^m = \left\{\text{population}_1^m, \text{population}_2^m, \ldots, \text{population}_{\text{pop\_num}}^m\right\} \tag{5}$$

where pop_num is a predetermined population size.

2) *Genetic Operations:* Elite retention, selection, crossover, and mutation operations are performed on the current multipopulation to generate a better set of populations for the subsequent selection of dominant individuals. Details of crossover, mutation, and other operations are in Section III-C.

3) *Population-Dominant Individual Selection:* The purpose of population dominance individual selection is to screen for dominant individuals, i.e., individuals with high fitness values, among multiple populations. The genes in this group of dominant individuals constitute the dominant genome, which is used for subsequent screening of the inferior genes. The optimal individual $\text{inds}_{ybest}^m$ in a single population $\text{population}_y^m$ within $\text{Pops}^m$ can be obtained by the following equation:

$$\begin{aligned} \text{inds}_{\text{ybest}}^m &= \arg \max_{\text{inds}_{yx}^m} \text{Fitness}(G, \text{inds}_{yx}^m) \\ &\text{s.t. } \text{inds}_{yx}^m \in \text{population}_y^m \end{aligned} \tag{6}$$

where $\text{Fitness}(G, \text{inds}_{yx}^m)$ is the fitness of the $x$th individual $\text{inds}_{yx}^m$ in the population $\text{population}_y^m$, and when the fitness of an individual in the population is maximized, that individual is the optimal individual in the population. $\text{population}_y^m \in \text{Pops}^m$ is the $y$th population in multiple populations at the $m$th iteration of the algorithm. With the above equation, the pop_num optimal individual combinations $\text{inds\_best}^m = \{\text{inds}_{ybest}^m, y = 1, 2, \ldots, \text{pop\_num}\}$ from pop_num populations can be obtained and used to integrate the gene screening of multiple individuals. In the above process, $\text{population}_y^m$ can be obtained by two strategies.

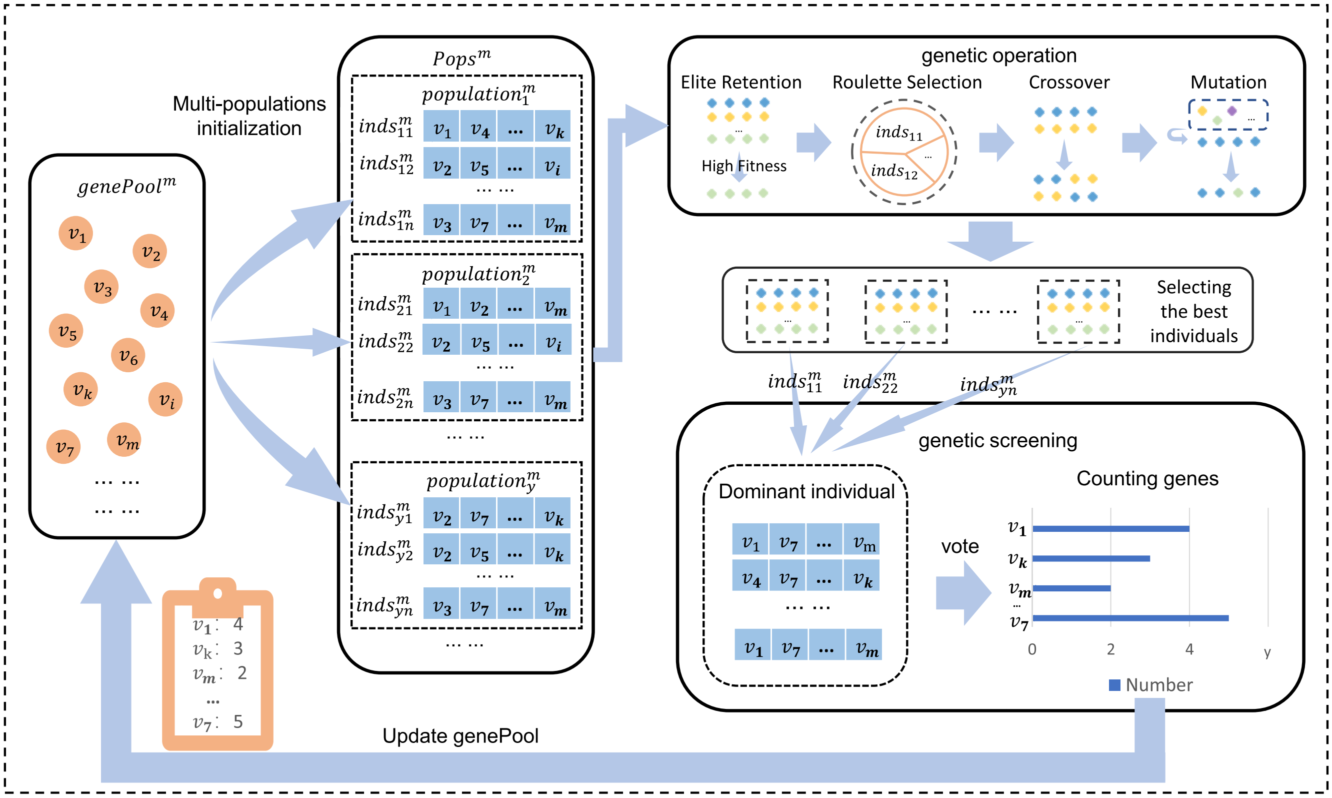


Fig. 2. Algorithmic process of multipopulation synergistic solution space pruning.

In the popInitial strategy, the initial population can be directly used without genetic manipulation, which is obtained by sampling the gene pool. In popEvolutiuon, population is updated by iterative genetic manipulation of the initial population, which has a higher quality of the population and dominant individuals and is beneficial to improve the quality of gene screening, but it will cause an increase in time consumption of the process.

4) *Integration of Multiindividuals Genetic Screening:* After obtaining multiple dominant individuals $\text{inds\_best}^m$, they will be used to screen for genes in the current gene pool. Gene screening is a process of retaining dominant genes and eliminating inferior genes. In this article, the node genes in the current gene pool are ranked by their frequency of occurrence in the dominant individual population, and the top $\text{Num}(\text{genePool}^m)$ genes are selected to construct the $m$th genePool

$$p(v_i^m) = \sum_{\text{indi}\in\text{inds\_best}^m} IF(v_i^m \in \text{indi}) \tag{7}$$

where $p(v_i^m)$ denotes the frequency of occurrence of $v_i^m$ in inds_best for the genes in the gene pool updated $m$ times, $IF(v_i^m \in \text{indi})$ denotes whether $v_i^m$ appears in indi, and 1 if it appears, 0 if it does not. The nodes with the highest frequencies are retained, and the nodes with lower frequencies are removed from the current gene pool to build a new gene pool. The following equation obtains the number of genes in the gene pool after $m$ updates:

$$\text{Num}(\text{genePool}^m) = (1-m\cdot\alpha)\cdot\text{Num}(\text{genePool}^0) \tag{8}$$

where $\text{Num}(\text{genePool}^m)$ is the number of genes in the gene pool after the $m$th update and $\alpha$ is the deletion factor in the gene screening process, which determines the ratio of the number of genes deleted from the gene pool at each iteration of the popEvolution and popInitial algorithms to the number of genes in the initial gene pool.

Repeated execution of the above four steps can gradually reduce the number of genes in the gene pool until the number of genes satisfies the preset demand. The final censored gene pool constitutes the censored node set $V^-$. Due to the dominant individual selection mechanism and gene voting mechanism, the genes appearing on the dominant individual have a higher probability of being retained, so the number of candidate solutions in the solution space composed of $V^-$ is reduced. However, the dominant solution is retained as much as possible through selection. Furthermore, the average quality of candidate solutions is improved, which benefits the subsequent optimization algorithm for an efficient solution of the target problem. The pseudo-code of the multipopulation synergistic gene screening algorithm using the popEvolution strategy is shown in Algorithm 1.

### C. Postgene Screened CND Module

Based on the set of nodes $V^-$ obtained after the multipopulation synergistic gene screening, this article uses GA to solve

**Algorithm 1** Multipopulation Synergistic Gene Screening Algorithm

**Input**: The number of Populations: $pop\_num$, Population size: $pop\_size$, Network: $G(V)$, The number of critical nodes: k, The number of evolution of subpopulations: $subpop\_g$, Deletion factor: $\alpha$, The retention rate of genes: $retention\_rate$.

**Output**: node set after deletion: $V^-$.

```
$m = 0$.
$genePool^0 \leftarrow \{v|v_i \in V, i = 1, 2, 3, \ldots, |V|\}$.
while $Num(genePool^m) > retention\_rate \cdot Num(genePool^0)$ do
    $Pops^m \leftarrow \emptyset$.
    for $y = 0; y < pop\_num; y++$ do
        $Population^m_y \leftarrow \emptyset$.
        for $x = 0; x < pop\_size; x++$ do
            $\text{inds}^m_{yx} \leftarrow \left\{v^m_x | v^m_{xi} \in genePool^m, i = 1, 2, \ldots, k\right\}$.
            $Population_i.append(ind^m s_{yx})$.
        end for
        $Pops^m.append(Population^m_y)$.
    end for
    $Pops^{m'} \leftarrow Genetic\_Operations(Pops^m, subpop\_g)$.
    $\text{inds\_best}^m \leftarrow \emptyset$.
    for $y = 0; y < pop\_num; y++$ do
        $\text{inds}^m_{ybest} \leftarrow$ Individuals with the highest fitness in $Pops^{m'}[y]$.
        $\text{inds\_best}^m.append(\text{inds}^m_{ybest})$.
    end for
    Counting the frequency of occurrence of genes in $genePool^m$ in $\text{inds\_best}^m$ as the number of votes for that gene.
    $genePool^{m+1} \leftarrow (1-m \cdot \alpha) \cdot Num(genePool^0)$ gene with the highest number of votes in $genePool^m$.
    $m = m + 1$.
end while
$V^- \leftarrow genePool^m$.
return $V^-$.
```

the CND problem. Fig. 3 illustrates the overall flow of the algorithm. The genotype design and fitness function of the algorithm are the same as the one in Sections III-A and III-B. And the population initialization, selection, crossover, mutation, and elite retention operators are described below.

1) *Initialization:* The initialization of the population is based on the node set $V^-$. A fixed number of genes are randomly selected to make up the individuals in the first generation, and no duplication of genes is allowed in the individuals. The resulting population can be denoted as $\text{Population}_0$. After obtaining the initial population, the fitness set of the individuals can be calculated in the population and denoted as $\text{Fitness}_0$.
2) *Elite Retention:* In the elite retention process, the elite_num individuals with the highest fitness are directly obtained from the current population to form the elite set elite_inds. This set can be directly inherited to the next generation without crossover and mutation operations.

**Algorithm 2** GAs With Elite Retention

**Input**: Population size: $pop\_size$, Network: $G(V)$, Number of elites: $elite\_num$, Crossover rate: $pc$, Mutation rate: $pm$, Number of iterations: $g$, Censored node set $V^-$.

**Output**: Collection of critical nodes: $\hat{V}^{critical}$.

```
$i = 0$.
$Population_0 \leftarrow Initialization(genePool)$.
while $i < g$ do
    $Fitness_i \leftarrow Get\_Fittest(Population_i)$.
    $elite\_\text{inds}_i \leftarrow Elite\_retention(elite\_num)$.
    $select\_\text{inds}_i \leftarrow Selection(Population_i, pop\_size - elite\_num)$.
    $crossover\_\text{inds}_i \leftarrow Crossover(select\_\text{inds}_i)$.
    $mutate\_\text{inds}_i \leftarrow Mutation(crossover\_\text{inds}_i)$.
    $Population_{i+1} = elite\_\text{inds}_i + mutate\_\text{inds}i$.
    $i = i + 1$.
end while
$Fitness_g \leftarrow Get\_Fittest(Population_g)$.
$top\_pop \leftarrow$ The individual with the highest fitness in $Population_g$.
$\hat{V}^{critical} \leftarrow$ Get the collection of critical nodes from $top\_pop$.
```

3) *Selection:* In the selection process, (pop_num − elite_num) individuals are selected from the current population using the roulette operator and noted as selected_inds for subsequent crossover and mutation operations.
4) *Crossover:* In the crossover process, two individuals are selected from selected_inds in turn, and the point crossover operation is performed according to the crossover rate pc. After all the individuals in selected_inds have completed the crossover operation, the resulting set of individuals can be recorded as crossover_inds. In the crossover process, it is also necessary to avoid gene duplication.
5) *Mutation:* In the mutation process, the mutation operation is performed for each individual in the crossover individual set crossover_inds based on the mutation rate $pm$. Each node gene in the individual can be randomly mutated to any node in the reduced gene pool, and mutate_inds can be obtained after completing the mutation operation for each individual. As before, gene duplication should be avoided in the mutation process.

After executing the above genetic operation, the resulting elite_inds and mutate_inds are combined to obtain the next generation. When the average fitness of the individuals converges completely or the number of iterations reaches a specified number, the optimal individual in the population will be considered as the solution of the algorithm. The pseudocode of this algorithm is shown in Algorithm 2.

## IV. Experiment

### A. Datasets

In order to verify the effectiveness of the proposed method in this work, experiments were conducted on several synthetic networks and real networks. First, the datasets used in the

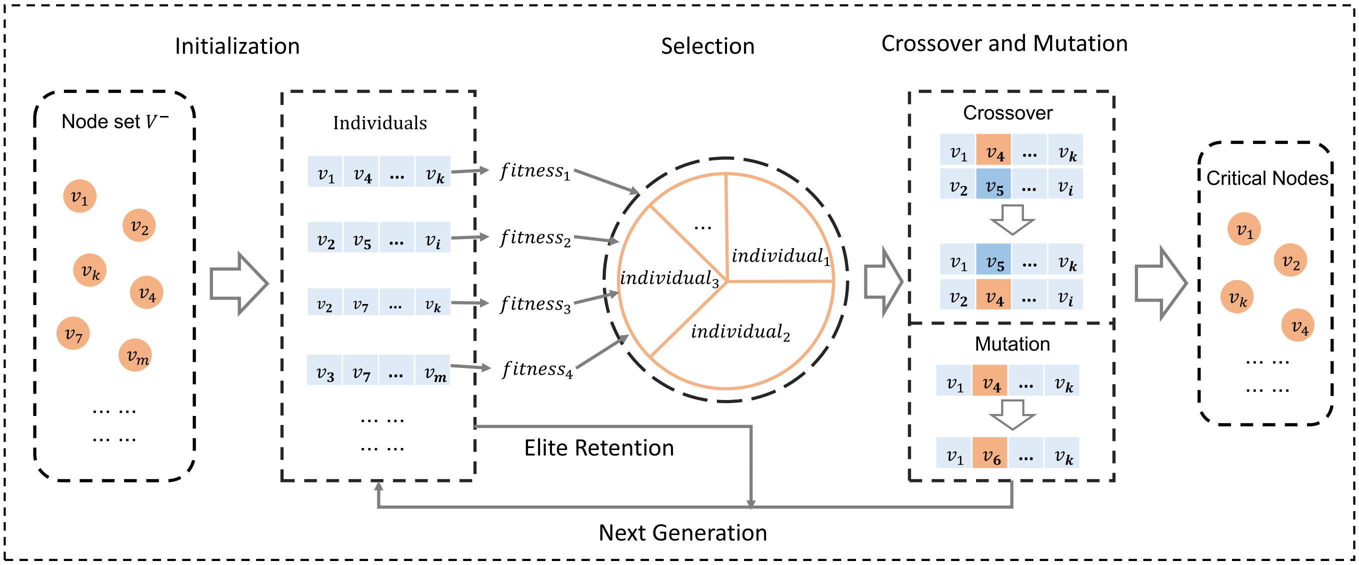


Fig. 3. GA-based CND algorithm.

TABLE I
DESCRIPTION ON EXPERIMENTAL DATASETS

| Datasets | Number of nodes | Number of edges | Average degree |
|---|---|---|---|
| ER500 | 466 | 700 | 3 |
| FF500 | 500 | 828 | 3.31 |
| WS500 | 500 | 1496 | 5.98 |
| BA500 | 500 | 499 | 2 |
| humanDiseasome | 516 | 1188 | 4.6 |
| powergrid | 4941 | 6594 | 2.67 |
| Circuit | 252 | 399 | 3.17 |
| A01 | 311 | 640 | 4.94 |

experiments are described. For the synthetic dataset, this work uses the classical synthetic networks Erdős–Rényi (ER) [57] stochastic network, Watts-Strogatz (WS) [58] (small world network), Barabási-Albert (BA) [59] (scale-free network), and the ForestFire (FF) [60] (forest fire synthetic network), which is commonly used in CND problems. Specifically, every two nodes in the ER random network are connected to each other with equal probability; WS small-world networks have short shortest paths between two nodes; the nodes in BA scale-free networks have high heterogeneity and critical nodes have high degree. For the real networks in this work, HumanDiseasome [61] (the disease network dataset), Powergrid [62] (the power network dataset), Circuit [63] (the circuit network), and A01 [64] (the citation network) are chosen. The relevant information on the above networks is shown in Table I.

In this work, numerical experiments on the above dataset are compared with several pruning strategies to illustrate the impact of pruning methods on the search efficiency of the algorithm and the performance of the algorithm, and to verify the effectiveness of the proposed method.

## B. Baseline

The following is a brief description of the baseline method.

1) *Pruning-Free Strategy(No_Cutoff):* Without solution space pruning, directly search for the critical nodes by GA.
2) *Random Pruning Strategy(Random):* A random pruning strategy was used, whereby some nodes in the original network were randomly removed from the gene pool. Then the critical nodes are calculated by GA.
3) *Greedy Pruning Strategy (Greedy):* The greedy strategy-based pruning method was used to remove the nodes with the lowest degree value from the original network. The nodes with low degree value are removed from the gene pool, and then the critical nodes are calculated by GA.
4) *TDE-Degree:* The degree-based differential evolution algorithm proposed by Yu et al. [24] improves the search efficiency of the differential evolution-based CND algorithm by incorporating the degree value information in the network into the genotype of the differential evolution algorithm.
5) *Pruning Strategies Based on Multiprimitive Populations (popInitial):* The solution space pruning method based on the synergistic gene screening strategy for multipopulations proposed in this article, where the selection of dominant population individuals is implemented on each initial population. Then the critical nodes are calculated by GA.
6) *Pruning Strategies Based on Multioptimal Populations (popEvolution):* The solution space pruning method based on the synergistic gene screening strategy for multipopulations proposed in this article. The selection of dominant population individuals is achieved on an optimized population after genetic operations. Then the critical nodes are calculated by GA.

## C. Parameters

To ensure the comparability of the experiments, the same parameters are used for the various pruning strategies. The

TABLE II
DESCRIPTION ON EXPERIMENTAL PARAMETERS

| Parameters | Value | Description |
|---|---|---|
| $pop\_num$ | 10 | The number of populations in $popInitial$ and $popEvolution$, i.e. the number of learners in ensemble learning |
| $pop\_size$ | 100 | Number of individuals in a single population of EA-based algorithms |
| $g$ | 5000 | The number of iteration of GA-Based CND algorithm and TDE-degree |
| $elite\_num$ | 0.1 | The elite retention rate in the population of popInitial, popEvolution, and the GA-based CND algorithm |
| $pc$ | 0.2 | Crossover rates for several EA-based algorithms |
| $pm$ | 0.6 | Mutation rate for several EA-based algorithms |
| $F$ | 0.2 | Differential weight in TDE-degree |
| $k$ | $0.1 * \|V\|$ | Number of critical nodes, taken as 10% of the number of network nodes |
| $retention\_rate$ | 0.4 | The proportion of remaining genes in the gene pool after pruning by several pruning algorithm |
| $\alpha$ | 0.2 | The ratio of the number of genes removed from the gene pool to the original number of genes for each iteration of popEvolution and popInitial |
| $subpop\_g$ | 30 | Number of evolutions of subpopulations in $popEvolulution$ |

parameter settings of several pruning algorithms, GA-based CND algorithm, and TDE-degree are shown in Table II.

### D. Algorithm Performance Analysis

This section compares the GA-based CND algorithm after various pruning strategies and TDE-degree to verify the effect of pruning strategies on algorithm efficiency. The time cost required for pruning is ignored when comparing performance because it requires much less computation time compared with the critical node detection process which requires a large number of iterations. The results obtained are shown in Table III. It is worth noting that due to the stochastic nature of the EA, there is bias in the algorithm's calculated results each time. So, the data in Table III were averaged through five experiments to mitigate the bias caused by this randomness.

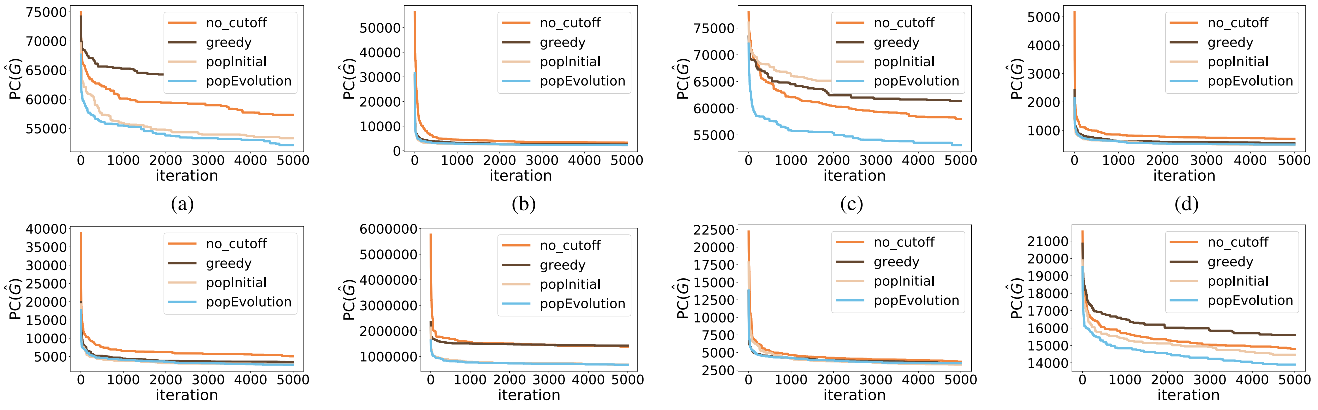
The data in Table III show that the proposed multipopulation synergistic gene-screening-based pruning strategies popInitial and popEvolution outperform the original method no_cutoff, other pruning methods and TDE-degree for the same number of population iterations. popEvolution outperforms popInitial in more cases, which indicates that optimizing the quality of optimal individuals in the population during the pruning process has a positive effect on the subsequent integration of multiple individuals for gene screening and the critical node detection process. The TDE-degree algorithm deals with the CND problem by incorporating the network node degree information in the genotype, but this operation also limits the algorithm's performance. In this article on TDE, the authors used the network robustness index R [24] as the objective function, and the calculation of this function strongly correlates with the degree value of the nodes in the network. Hence, the execution of the TDE-degree in this article is more efficient. Still, when the objective function is replaced with the pairwise connectedness PC($\hat{G}$), which has a weak correlation with the degree value, the algorithm tends to fall into the local optimum. Under the pruning method based on the random strategy, the algorithm's performance degrades significantly. This is because some high-quality genes are extremely easy to be eliminated under the random pruning strategy, resulting in the overall low quality of the candidate solutions in the solution space after pruning, and even the quality of the optimal candidate solutions after pruning being significantly worse than that of the optimal candidate solutions in the original solution space. The same reason can be used to explain the poor performance of the pruning method based on the greedy strategy on datasets with low heterogeneity, such as WS500 and ER500. Fig. 4 shows the pairwise connectivity of the optimal individuals in the population for each iteration. The numerical results in Table III and the convergence curve in Fig. 4 are obtained by taking the mean of the five times experimental results due to the randomness of the EA. Since the performance of TDE-degree and the pruning algorithm based on the random strategy is generally inferior to that of the no_cutoff method, it is not shown in the figure. It can be seen that after popInitial and popEvolution pruning, the algorithm can reduce the pairwise connectivity of the network significantly at the beginning of the iterations compared with the no_cutoff method. The greedy pruning method has the same performance in several networks. However, it is generally inferior to the two population-based pruning algorithms proposed in this article. At the same time, the greedy pruning strategy has a similar convergence curve with popInitial and popEvolution on the BA500 and FF500 datasets with high heterogeneity. Compared to the GA-based CND algorithm, the time required for popInitial and popEvolution is insignificant. In addition, compared to other methods, popInitial and popEvolution have a smaller probability of falling into a local optimum because multiple populations increase the adequacy of gene pool sampling, allowing a higher probability of the dominant gene being retained when the inferior gene is eliminated. Therefore, the proposed pruning method has performance improvements compared with the GA-based CND algorithm.

### E. Correlation Analysis of Algorithm Effect and Network Structure

The results in Table III show that the proposed algorithm performs better in different networks compared to the no_cutoff method. Moreover, in general, it can be known that it shows better performance on BA scale-free networks than on FF, WS, and ER networks. Since the nodes in BA scale-free networks are more heterogeneous, and the degree distribution is closer to the power-law distribution, this work considers that this method and the greedy strategy-based

TABLE III
RESULT OF EXPERIMENTS ON ALGORITHM PERFORMANCE

| datasets | PC($\hat{G}$) of final iteration | | | | | |
|---|---|---|---|---|---|---|
| | no_cutoff | random | greedy | TDE-degree | popInitial | **popEvolution** |
| ER500 | 57328.8 | 64686.4 | 62828.8 | 69472.0 | 53302.2 | **52112.8** |
| FF500 | 3307.4 | 3169.8 | 2562.8 | 13726.0 | 2267.6 | **2261.8** |
| WS500 | 57992.2 | 72135.4 | 61366.0 | 70732.0 | 63140.8 | **53082.0** |
| BA500 | 701.6 | 28169.6 | 541.8 | 1464.0 | **476.2** | 499.2 |
| HumanDiseasome | 5046.4 | 31061.2 | 3461.2 | 13726.0 | 2990.8 | **2771.2** |
| Powergrid | 1383230.8 | 2279412.4 | 1423846.4 | 2892824.0 | 680566.8 | **674489.4** |
| Circuit | 3692.8 | 19561.6 | 3462.0 | 10688.0 | **3281.6** | 3470.4 |
| A01 | 14799.4 | 17573.6 | 15595.6 | 18942.0 | 14464.0 | **13898.6** |



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 4. Convergence plots of the network pairwise connectivity corresponding to the optimal individual in each iteration when executing the GA-based CND algorithm after pruning the GA solution space on eight networks. (a) ER500. (b) FF500. (c) WS500. (d) BA500. (e) HumanDiseasome. (f) Powergrid. (g) Circuit. (h) A01.

TABLE IV
RESULT OF EXPERIMENTS ON THE INFLUENCE OF NETWORK HETEROGENEITY

| $p$ | PC($\hat{G}$) | | | Degree of promotion % | |
|---|---|---|---|---|---|
| | no_cutoff | greedy | **popEvolution** | greedy % | **popEvolution %** |
| 0.1 | 96316.9 | 95747.6 | **94961.8** | 0.59% | **1.41%** |
| 0.3 | 94961.6 | 93617.8 | **92238** | 1.42% | **2.87%** |
| 0.5 | 90999 | 87369.1 | **85710.8** | 3.99% | **5.81%** |
| 0.7 | 68439 | 61890.7 | **59334.9** | 9.57% | **13.30%** |
| 0.9 | 974.5 | 637.5 | **549.3** | 34.58% | **43.63%** |

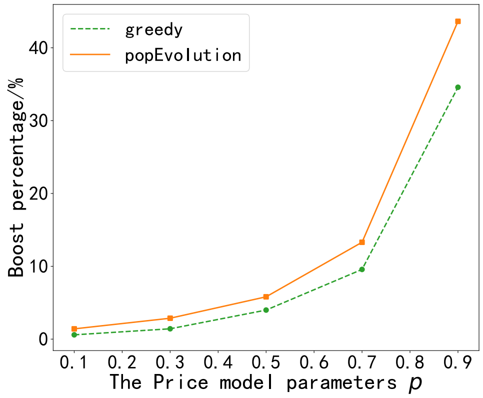


Fig. 5. Result of correlation experiments between algorithm effect and network heterogeneity.

method perform better on more heterogeneous networks. For this reason, the proposed hypothesis is tested experimentally in this subsection. In order to construct networks with different heterogeneity, this work uses the Price model [65], [66] to construct networks with different heterogeneity for experiments. The Price model construction method is shown in Algorithm 3.

Among the parameters of the above algorithm, $p$ is the priority connection probability, which is used to confirm whether to perform a priority connection when accessing a new node, i.e., whether to give a higher connection probability to the node with a higher degree value. A larger priority connection probability $p$ indicates that nodes with larger degree values will connect to newer access nodes and the network will have a higher degree of heterogeneity. In generating networks with heterogeneous differences based on the above algorithm, we assign values 0.1, 0.3, 0.5, 0.7, and 0.9 to the parameter $p$, respectively, while ensuring that the other parameters are consistent. Specifically, the target network size $N$, the initial network size $m_0$, and the connectivity factor $m$ are set to 500, 8, and 3, respectively. Experiments are conducted in the constructed network dataset, and the results are shown in Table IV and Fig. 5.

As shown in Table IV, after implementing the pruning methods greedy and popEvolution in the network dataset constructed based on the Price model, the performance of the CND algorithm is improved in both cases compared with that

**Algorithm 3** Network Construction Algorithm Based on Price Model

**Input**: Target network size: $N$, priority connection probability: $p$, initial network size: $m_0$, connection coefficient: $m$
**Output**: Synthesis Network

```
Generate an initial strongly connected network with node number $m_0$
Store the nodes pointed to by each edge in the initial network in the array $Array$
for i in range($m_0$, $N$) do
    $m\_node \leftarrow \emptyset$
    for j in range(0, $m$) do
        randomly generated $r \in [0, 1)$.
        if $r > p$ then
            choose a node at random and put it into $m\_node$.
        else if $r < p$ then
            choose a random node from array and put it into $m\_node$.
        end if
    end for
    Add a node to the network that is connected to the node in $m\_node$.
    Put the concatenated edges of the added nodes into $Array$.
end for
Output synthetic dataset.
```

without pruning. In addition, the degree of algorithm performance improvement increases with the growth of parameter $p$, i.e., the heterogeneity of the network.

## V. Conclusion

In this work, to solve the problem of large solution space and the existence of numerous inferior solutions when using GA to solve CND problems, a multipopulation synergistic gene screening algorithm based on integrated learning is proposed. The algorithm indirectly prunes the GA solution space by screening and removing inferior genes from the gene pool to reduce the number of individuals containing inferior genes significantly. The algorithm can consider the coupling information among multiple genes and measure the quality of genes by their frequency of occurrence on the dominant individuals in multiple populations. The parallel execution of multiple populations allows the gene pool to be adequately sampled, increasing the probability of dominant genes being retained while inferior genes are censored. The proposed algorithm reduces the size of the solution space of the GA-based CND algorithm, increases the probability of the dominant solution being selected, and improves the search efficiency of the algorithm. The experiments on several artificial and real networks compare the performance of GA-based CND algorithms with and without pruning strategies. Experimental results show that the proposed algorithm can improve the efficiency of the GA-Based CND algorithm on several networks and performs better on heterogeneous networks.

As more and more attention is focused on the CND problem for complex networks, much work remains to be investigated. The solution space pruning strategy proposed in this article can improve the probability of dominant solutions not being rejected to a certain extent through multipopulation synergistic genetic screening. However, the genetic screening-based strategy will unavoidably reject some dominant solutions in a specific class of network datasets. Therefore, based on pruning at the gene level, it is still promising further to investigate pruning methods at the candidate solution level to ensure the probability of the dominant solution being retained through a more refined solution space pruning strategy, thus better maintaining the performance of the algorithm based on achieving efficient search.

## Acknowledgment

The authors would like to thank all the members in the IVSN Research Group, Zhejiang University of Technology for the valuable discussion about the ideas and technical details presented in this article.

## References

[1] D. T. Nguyen, Y. Shen, and M. T. Thai, "Detecting critical nodes in interdependent power networks for vulnerability assessment," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 151–159, Mar. 2013.

[2] 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.

[3] 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.

[4] 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.

[5] X. Liu et al., "Computational methods for identifying the critical nodes in biological networks," *Briefings Bioinf.*, vol. 21, no. 2, pp. 486–497, Mar. 2020.

[6] A. Basu, P. E. Ash, B. Wolozin, and A. Emili, "Protein interaction network biology in neuroscience," *Proteomics*, vol. 21, nos. 3–4, Feb. 2021, Art. no. 1900311.

[7] M. Coscia, F. M. H. Neffke, and R. Hausmann, "Knowledge diffusion in the network of international business travel," *Nature Human Behaviour*, vol. 4, no. 10, pp. 1011–1020, Aug. 2020.

[8] J. Zhu, P. Ni, and G. Wang, "Activity minimization of misinformation influence in online social networks," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 4, pp. 897–906, Aug. 2020.

[9] N. Kolli and B. Narayanaswamy, "Influence maximization from cascade information traces in complex networks in the absence of network structure," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1147–1155, Dec. 2019.

[10] Z. Sičanica and I. Vujaklija, "Resilience to cascading failures. a complex network approach for analysing the croatian power grid," in *Proc. 43rd Int. Conv. Inf., Commun. Electron. Technol. (MIPRO)*, 2020, pp. 918–922.

[11] F. Yang, C. Ma, C. Chang, and J. Huang, "Constrained optimization of critical node detection with cost model in communication networks," in *Proc. IEEE 5th Inf. Technol., Netw., Electron. Autom. Control Conf. (ITNEC)*, vol. 5, Jul. 2021, pp. 1754–1758.

[12] T. N. Dinh and M. T. Thai, "Precise structural vulnerability assessment via mathematical programming," in *Proc. MILCOM Mil. Commun. Conf.*, Nov. 2011, pp. 1351–1356.

[13] H. Xu, J. Zhang, J. Yang, and L. Lun, "Identifying important nodes in complex networks based on multiattribute evaluation," *Math. Problems Eng.*, vol. 2018, pp. 1–11, May 2018.

[14] 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.

[15] 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.

[16] 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.

[17] D.-S. Yang, Y.-H. Sun, B.-W. Zhou, X.-T. Gao, and H.-G. Zhang, "Critical nodes identification of complex power systems based on electric cactus structure," *IEEE Syst. J.*, vol. 14, no. 3, pp. 4477–4488, Sep. 2020.

[18] K. R. Opara and J. Arabas, "Differential evolution: A survey of theoretical analyses," *Swarm Evol. Comput.*, vol. 44, pp. 546–558, Feb. 2019.

[19] E. Béczi and N. Gaskó, "Approaching the bi-objective critical node detection problem with a smart initialization-based evolutionary algorithm," *PeerJ Comput. Sci.*, vol. 7, p. e750, Oct. 2021.

[20] Y. Xu and P. Guo, "MEA-CNDP: A membrane evolutionary algorithm for solving biobjective critical node detection problem," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–20, Nov. 2021.

[21] 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.

[22] G. Wu, M. Li, and Z. S. Li, "A gene importance based evolutionary algorithm (GIEA) for identifying critical nodes in cyber–physical power systems," *Rel. Eng. Syst. Saf.*, vol. 214, Oct. 2021, Art. no. 107760.

[23] 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.

[24] S. Yu et al., "An improved differential evolution framework using network topology information for critical nodes detection," *IEEE Trans. Computat. Social Syst.*, vol. 10, no. 2, pp. 448–457, Apr. 2023.

[25] R. Reed, "Pruning algorithms—A survey," *IEEE Trans. Neural Netw.*, vol. 4, no. 5, pp. 740–747, 1993.

[26] M. Zhao, Q. He, S. Li, and M. Ren, "An improved local search algorithm with pruning for satellite data transmission scheduling problem," in *Proc. IEEE 21st Int. Conf. Softw. Qual., Rel. Secur. Companion (QRS-C)*, Dec. 2021, pp. 561–568.

[27] Z. Qin, Y. Bai, and Y. Sun, "GHashing: Semantic graph hashing for approximate similarity search in graph databases," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 2062–2072.

[28] A. Mahesh and G. Sushnigdha, "A novel search space reduction optimization algorithm," *Soft Comput.*, vol. 25, no. 14, pp. 9455–9482, Jul. 2021.

[29] L. I. Silva, E. A. Belati, C. Gerez, and I. C. Silva Junior, "Reduced search space combined with particle swarm optimization for distribution system reconfiguration," *Electr. Eng.*, vol. 103, no. 2, pp. 1127–1139, Apr. 2021.

[30] Y. Li, K. Adamczewski, W. Li, S. Gu, R. Timofte, and L. Van Gool, "Revisiting random channel pruning for neural network compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 191–201.

[31] O. Sagi and L. Rokach, "Ensemble learning: A survey," *Wiley Interdiscipl. Reviews: Data Mining Knowl. Discovery*, vol. 8, no. 4, p. e1249, 2018.

[32] K. Wang, T. Hu, and B. Zhao, "An unsupervised deep neural network approach based on ensemble learning to suppress seismic surface-related multiples," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5924514.

[33] K. Wang, T. Hu, and S. Wang, "Iterative deblending using unsupervised learning with double-deep neural networks," *Geophysics*, vol. 88, no. 3, pp. V187–V205, May 2023.

[34] K. Wang, T. Hu, B. Zhao, and S. Wang, "An unsupervised learning method to suppress seismic internal multiples based on adaptive virtual events and joint constraints of multiple deep neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5902918.

[35] M. Kumar, M. Husian, N. Upreti, and D. Gupta, "Genetic algorithm: Review and application," *Int. J. Inf. Technol.*, vol. 2, no. 2, pp. 451–454, 2010.

[36] J. Chen et al., "GA-based Q-attack on community detection," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 3, pp. 491–503, Jun. 2019.

[37] J. Chen, Y. Chen, L. Chen, M. Zhao, and Q. Xuan, "Multiscale evolutionary perturbation attack on community detection," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 1, pp. 62–75, Feb. 2021.

[38] S. Yu, J. Zheng, J. Chen, Q. Xuan, and Q. Zhang, "Unsupervised Euclidean distance attack on network embedding," in *Proc. IEEE 5th Int. Conf. Data Sci. Cyberspace (DSC)*, Jul. 2020, pp. 71–77.

[39] T. Harada and E. Alba, "Parallel genetic algorithms: A useful survey," *ACM Comput. Surveys*, vol. 53, no. 4, pp. 1–39, Jul. 2021.

[40] 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.

[41] Y. Yang, L. Yu, X. Wang, Z. Zhou, Y. Chen, and T. Kou, "A novel method to evaluate node importance in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 526, Jul. 2019, Art. no. 121118.

[42] 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.

[43] G. Mao-Guo, J. Li-Cheng, Y. Dong-Dong, and M. Wen-Ping, "Research on evolutionary multi-objective optimization algorithms," *J. Softw.*, vol. 20, no. 2, pp. 271–289, 2009.

[44] Y. Guo, G. Chen, M. Jiang, D. Gong, and J. Liang, "A knowledge guided transfer strategy for evolutionary dynamic multiobjective optimization," *IEEE Trans. Evol. Comput.*, early access, Nov. 16, 2022, doi: 10.1109/TEVC.2022.3222844.

[45] Y.-N. Guo, X. Zhang, D.-W. Gong, Z. Zhang, and J.-J. Yang, "Novel interactive preference-based multiobjective evolutionary optimization for bolt supporting networks," *IEEE Trans. Evol. Comput.*, vol. 24, no. 4, pp. 750–764, Aug. 2020.

[46] G. Chen, Y. Guo, M. Huang, D. Gong, and Z. Yu, "A domain adaptation learning strategy for dynamic multiobjective optimization," *Inf. Sci.*, vol. 606, pp. 328–349, Aug. 2022.

[47] F. Liu, B. Xiao, and H. Li, "Finding key node sets in complex networks based on improved discrete fireworks algorithm," *J. Syst. Sci. Complex.*, vol. 34, no. 3, pp. 1014–1027, Jun. 2021.

[48] G. Zhao, P. Jia, C. Huang, A. Zhou, and Y. Fang, "A machine learning based framework for identifying influential nodes in complex networks," *IEEE Access*, vol. 8, pp. 65462–65471, 2020.

[49] Y. Chen, D. Abts, and D. J. Lilja, "State pruning for test vector generation for a multiprocessor cache coherence protocol," in *Proc. 15th IEEE Int. Workshop Rapid Syst. Prototyping*, Jul. 2004, pp. 74–77.

[50] F. Merz and P. Sanders, "PReaCH: A fast lightweight reachability index using pruning and contraction hierarchies," in *Proc. Eur. Symp. Algorithms*. Cham, Switzerland: Springer, 2014, pp. 701–712.

[51] M. S. Kadu, R. Gupta, and P. R. Bhave, "Optimal design of water networks using a modified genetic algorithm with reduction in search space," *J. Water Resour. Planning Manage.*, vol. 134, no. 2, pp. 147–160, Mar. 2008.

[52] A.-D. Li, B. Xue, and M. Zhang, "Improved binary particle swarm optimization for feature selection with new initialization and search space reduction strategies," *Appl. Soft Comput.*, vol. 106, Jul. 2021, Art. no. 107302.

[53] S. J. Kwon, D. Lee, B. Kim, P. Kapoor, B. Park, and G.-Y. Wei, "Structured compression by weight encryption for unstructured pruning and quantization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2020, pp. 1909–1918.

[54] S. Roy, P. Panda, G. Srinivasan, and A. Raghunathan, "Pruning filters while training for efficiently optimizing deep learning networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7.

[55] Y. Zhang, Y. Zhen, Z. He, and G. G. Yen, "Improvement of efficiency in evolutionary pruning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.

[56] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[57] P. Erdos and A. Rényi, "On the evolution of random graphs," *Pub. Math. Inst. Hungarian Acad. Sci.*, vol. 5, no. 1, pp. 17–60, 1960.

[58] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'smallworld' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

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

[60] B. Drossel and F. Schwabl, "Self-organized critical forest-fire model," *Phys. Rev. Lett.*, vol. 69, no. 11, pp. 1629–1632, Sep. 1992.

[61] K. Goh, M. E. Cusick, D. Valle, B. Childs, M. Vidal, and A. L. Barabási, "The human disease network," *Proc. Nat. Acad. Sci. USA*, vol. 104, no. 21, pp. 8685–8690, 2007.

[62] J.-W. Wang and L.-L. Rong, "Cascade-based attack vulnerability on the U.S. power grid," *Saf. Sci.*, vol. 47, no. 10, pp. 1332–1336, Dec. 2009.

[63] R. Milo et al., "Superfamilies of evolved and designed networks," *Science*, vol. 303, no. 5663, pp. 1538–1542, Mar. 2004.

[64] (2001). *Graph A*. [Online]. Available: http://vlado.fmf.uni-lj.si/pub/networks/data/cite/default.htm

[65] D. J. D. S. Price, “Networks of scientific papers: The pattern of bibliographic references indicates the nature of the scientific research front,” *Science*, vol. 149, no. 3683, pp. 510–515, Jul. 1965.

[66] D. D. S. Price, “A general theory of bibliometric and other cumulative advantage processes,” *J. Amer. Soc. Inf. Sci.*, vol. 27, no. 5, pp. 292–306, Sep. 1976.

**Shanqing Yu** received the M.S. degree from the School of Computer Engineering and Science, Shanghai University, Shanghai, China, in 2008, and 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.

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

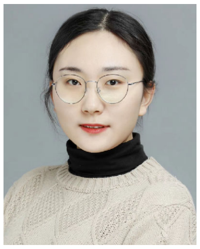
His current research interests include social network analysis and evolutionary computing.

**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.

**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.

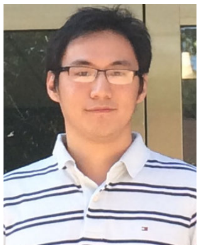


**Jinhuan Wang** received the B.S. and M.S. degrees from the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China, in 2017 and 2020, respectively, and the Ph.D. degree from the School of Information Engineering, Zhejiang University of Technology.

Her research interests include graph data mining, blockchain, and cyberspace security.

**Qi Xuan** (Member, IEEE) received the B.S. and Ph.D. degrees in control theory and engineering from Zhejiang University, Hangzhou, China, in 2003 and 2008, respectively.

He was a Post-Doctoral Researcher with the Department of Information Science and Electronic Engineering, Zhejiang University, from 2008 to 2010, and a Research Assistant with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, in 2010 and 2017. From 2012 to 2014, he was a Post-Doctoral Fellow with the Department of Computer Science, University of California at Davis, Davis, CA, USA. He is currently a Professor with the Institute of Cyberspace Security, College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His current research interests include network science, graph data mining, deep learning, and cyberspace security.



**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 algorithms design, social networks, data mining, chaos synchronization, network dynamics, and machine learning.