



Multi-objective memetic algorithm for core-periphery structure detection in complex network

Guo Li¹ · Zexuan Zhu^{1,2} · Lijia Ma¹ · Xiaoliang Ma¹

Received: 3 January 2021 / Accepted: 16 July 2021 / Published online: 28 July 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

Core-periphery structure detection (CPSD) in complex networks is essential to reveal functional nodes in the complicated systems, *e.g.*, influential nodes in a social network and central cells in a biological network. Some progress has been made in solving the CPSD problem with heuristic algorithms. However, CPSD is naturally an NP-hard optimization problem and the core-periphery structures (CPSs) in real networks usually are not clearly distinguishable. The majority of the existing CPSD methods are single-objective methods relying on some assumptions, preference, and/or prior knowledge. They can provide only one trade-off solution that is inevitably biased and lacks of flexibility in terms of resolution. To address this issue, this paper formulates the CPSD problem as a multi-objective optimization problem (MOP), *i.e.*, minimizing the core-node size and maximizing the core-node capacity of the CPSs, simultaneously. Solving the MOP can provide more accurate CPSs and allow one to explore the network structure at different preferred resolutions. A multi-objective memetic algorithm (called MOMA-PCLS) is accordingly proposed to solve the formulated problem. A new plateau-climbing local search (PCLS) method incorporating the information of the heavy-tailed distribution of the node capacity is introduced to fine-tune the individual solutions in MOMA-PCLS. By combining the evolutionary operations and PCLS, MOMA-PCLS manages to improve the search efficiency significantly. Experimental results on both synthetic and real-world data show the superiority of MOMA-PCLS to other state-of-the-art algorithms in detecting CPSs of complex networks.

Keywords Core-periphery structure · Multi-objective optimization · Memetic algorithm · Complex networks

1 Introduction

In recent years, core-periphery structures (CPSs) in complex networks have gained increasingly widespread academic concern thanks to their ubiquity and essential functionality in real-world social, biological, economic, and administra-

tive systems [7]. A CPS of a network is composed of a set of core and peripheral nodes, where the core nodes are located at the center while the peripheral nodes are individually connected to the core nodes (the conception is detailed in Sect. 3. CPS should be distinguished from the better-known community structure [23]. The difference between these two structures is the presence of peripheral nodes in CPS, which makes the hierarchy of the network structure more apparent.) Core-periphery structure detection (CPSD) is critical for the analysis of system stability and robustness [35]. Currently, most existing CPSD methods are heuristic algorithms based on block model comparison or optimization of core-periphery metrics [36]. For example, Borgatti and Everett [2] proposed an idealized block model pattern depicted on behalf of the CPS. If a network contains a CPS, the corresponding adjacency matrix should have a similar idealized pattern. The LapSgnCore algorithm [5] maximizes the connectivity between core nodes and minimizes the connectivity between peripheral nodes, simultaneously, by using the core-periphery connectivity metric. The Rombach algorithm [30]

✉ Zexuan Zhu
zhuzx@szu.edu.cn

Lijia Ma
omegamalj@gmail.com

Guo Li
szuliguo@szu.edu.cn

Xiaoliang Ma
maxiaoliang@yeah.net

¹ College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China

² Guangdong Provincial Key Laboratory of Brain-Inspired Intelligent Computation, Southern University of Science and Technology, Shenzhen 518055, China

uses a new core quality metric R_γ to evaluate the CPS quality. The KM-config algorithm [17] compares the dissimilarity of the detected CPS with a null model. The aforementioned methods have achieved promising results in CPSD, yet they also suffer from some limitations. The majority of the existing CPSD methods are single-objective methods relying on some assumptions, preferences, and/or prior knowledge, e.g., the number of core nodes and predefined thresholds of some objective metrics [18]. However, the CPSs in real complex networks usually have no clear partition and the identification of CPSs could be resolution dependent. The existing CPSD methods can provide only one trade-off solution, which is inevitably biased and inflexible in terms of resolution.

To solve this problem, we formulate CPSD as a multi-objective optimization problem (MOP) by minimizing the number of core nodes and maximizing the occupied network capacity (NC) of the core nodes (i.e., minimizing the occupied network capacity of periphery nodes), simultaneously, which can provide more accurate detection and resolution options. Multi-objective evolutionary algorithms (MOEAs) capable of optimizing multiple conflicting objectives simultaneously [19,24] serve as a good solution for the aforementioned MOP. MOEAs can optimize the two conflicting objectives simultaneously and provide users a set of non-dominated solutions with various resolution preferences [37]. Nevertheless, conventional MOEAs tend to have a slow convergence issue when solving complex problems like CPSD due to the lack of fast exploitation mechanism. Problem-specific or domain knowledge based local search methods can be introduced into the evolutionary algorithms to accelerate the convergence in local regions [10]. The resultant algorithms are also widely referred to as memetic algorithms (MAs) [3,26]. MAs have shown better performance than the counterpart conventional evolutionary algorithms on both continuous [12] and discrete [45,46] optimization problems.

To solve the formulated MOP, this paper proposes a multi-objective memetic algorithm with a plateau-climbing local search (called MOMA-PCLS) for the CPSD MOP. MOMA-PCLS encodes the network nodes into a binary string with ‘1’ and ‘0’ bits indicating core nodes and periphery nodes, respectively. The solution population is evolved with a combination of evolutionary operators and a plateau-climbing local search (PCLS). The fitness values of the MOP present a stepped distribution in the search space. Classical local search methods like hill-climbing methods tend to get trapped in local optima in such discrete search space. Accordingly, the PCLS method is proposed to prevent the stagnation of the search in a local region with a constant fitness value. PCLS utilizes the heavy-tailed distribution of the node capacity to reduce the search space. The knee point solution in the final non-dominated solution set of MOMA-PCLS is used as the representative to show the CPSs with a reasonable trade-off

resolution. Experimental results on both synthetic and real-world data show that the MOP formulation together with the MOMA-PCLS can detect CPSs more accurately than other state-of-the-art algorithms. In the discrete combination search space, PCLS outperforms the commonly used hill-climbing, simulated annealing (SA) [16], and Tabu search (TS) [9] based local search methods. The main contributions of this work are summarized as follows:

- To the best of our knowledge, this is the first attempt to solve CPSD in a multi-objective optimization paradigm, which provides better accuracy and more resolution options of the detection.
- A new multi-objective memetic algorithm is proposed to solve the MOP and systematically tested on various data sets to demonstrate its effectiveness and efficiency.
- A plateau-climbing local search method is introduced as a meme operator according to the feature of the discrete search space, which shows a faster convergence speed than traditional hill-climbing, SA, and TS based local search methods.

The rest of this paper is organized as follows. An overview of related work is provided in Sect. 2. Section 3 presents the preliminaries and formulation of the CSDPs. Section 4 provides the details of the MOMA-PCLS algorithm. The experimental results are presented in Sect. 5, and finally, Sect. 6 concludes this work.

2 Related work

This section provides a brief review of the existing CPSD algorithms and the MAs for network-aware combinatorial optimization problems.

2.1 Existing CPSD algorithms

Many CPSD algorithms have been proposed to identify CPS. Most of the existing algorithms are based on block model comparison or optimization of some core-periphery metrics. For example, Borgatti and Everett [2] introduced a well-known similarity comparison method for the CPSD. In an ideal CPS, the core nodes are adjacent to each other while the periphery nodes are connected with all core nodes but none other periphery nodes. The similarity between the adjacency matrix of the ideal CPS and that of the target CPS is used as the evaluation metric. Da *et al.* [6] utilized the network capacity to cluster the network nodes, based on which the CPS can be quantitatively evaluated. Kojaku *et al.* [17,18] presented a KM-config algorithm also by utilizing a random walker to detect a discrete CPS, which is similar to the Markov stability formalism. Cucuringu *et al.* [5] pro-

posed three algorithms namely LowRankCore, LapCore, and LapSgnCore to detect CPS. The LowRankCore algorithm aggregates information from multiple dimensions and calculates a score for each node that reflects the likelihood of the node being a core or periphery node. The LapCore algorithm is built on a low-rank approximation of a network's adjacency matrix. The LapSgnCore algorithm utilizes the bottom eigenvector of the random-walk Laplacian to infer a coreness score and then classify the core and periphery nodes. Li *et al.* [20] introduced a Core-Tree index based on distance labeling. By using a Core-Tree decomposition method, the core and periphery nodes can be divided. Jia *et al.* [15] put forward the probability of a random walk to determine the core and periphery attributes of the nodes. Rombach *et al.* [30] designed an approach that provides node values along a continuous spectrum. The proposed algorithm applies a label switching method and adopts a simulated annealing method to optimize the objective.

CPSD has also attracted broad interest in real-world applications. For example, Riaza *et al.* [29] expanded the traditional core-periphery structure to a core-semiperiphery-periphery structure and applied it to the network of miscellaneous imports of metal manufactures. Sarkar *et al.* [33] utilized F_1 -scores to detect core and periphery nodes. The detected CPS is utilized in predicting reside nodes on a large set of diverse time-varying networks. In [41] and [42], the structural correlation between CPS and community structure was investigated in social networks, and a unified method was proposed to detect both CPS and community structures. Jeude *et al.* [14] assigned a p-value to a given partition of the network nodes. This method can detect statistically significant bimodular structures such as bipartite structures and CPS structures. Gu *et al.* [11] unified the notions of community structure and core-periphery structure by the weighted stochastic block model, simultaneously uncovering these two structures in functional brain networks.

CPSD methods can be categorized into two types. One type is to measure the entire network, *e.g.*, comparing the similarity between a network partition and the ideal CPS model. The other is to score individual nodes, and then classify the core/periphery attribute. The existing CPSD methods have shown promising performance in various problems. However, the majority of them are single-objective method relying on some assumptions or prior knowledge. They tend to be biased and cannot accurately reflect the diversified characteristics of the CPSs.

2.2 MAs for network-aware combinatorial optimization

Over the last few years, MAs have been widely used for network-aware combinatorial optimization problems [1]. For example, Ma *et al.* [22] considered cost-aware robust control

in signed networks as a constrained combination optimization problem, then presented a MA with problem-specific knowledge to solve the cost-aware robust control problem. The experiments on both real social and biological networks outperformed several state-of-the-art robust controllability algorithms. Wang *et al.* [40] designed a MA optimization algorithm termed MA-CR_{inter} to enhance the community robustness on real-world interdependent networks. The MA-CR_{inter} algorithm provided different strategies to improve the network sustainability according to the network topology. However, the proposed robustness evaluation metric in community networks cannot be directly applied to the core-periphery networks, which are also robustness networks. Wang *et al.* [39] also proposed a community-based MA to select immunization nodes on an epidemic network. The adopted neighbor-based local search method could achieve lower computational complexity and better performance than the greedy algorithm. Gabardo *et al.* [8] utilized a link clustering MA for overlapping community detection. The local search mechanism iteratively changes the network structure to maximize the network nodes partition density. Ruíz *et al.* [32] proposed a parallel MA based on recurrent neural networks for energy consumption prediction. The parallel MA has been found to be more efficient than the sequential version. However, the difference between community topology and CPS topology makes these MAs hard to transplant to the CPSD problem.

Ibrahim *et al.* [13] proposed a multi-objective adaptive MA for backpropagation neural networks. The optimization objectives are to maximize network performance and minimize network complexity. The statistical analysis results demonstrated a significant performance improvement by the multi-objective MA. Cheng *et al.* [4] devised a local information-based multi-objective MA called LMOEA to improve the quality of community detection in complex networks; LMOEA adopted negative ratio association and ratio cut as optimization objectives and can simultaneously ensure the dense connections within a community and the sparse links between communities. The experimental results showed a high accuracy of community detection. Notably, selecting suitable objectives according to network structure is critical to the detection accuracy. Multi-objective MAs have been successfully applied to detect network structures such as community and exhibit more accurate results [27], yet there are no any MAs designed for CPSD.

This research is the first attempt to design a multi-objective memetic framework to solve the CPSD problem more accurately. In the CPSD field, multi-objective optimization serves as a new promising solution to overcome the inaccuracy problem in traditional single objective CPS metric optimization. In the MA field, this work combines the features of network data to design an efficient PCLS method, which provides effi-

cient search ability compared with traditional local search methods.

3 Preliminaries and problem formulation

To facilitate the understanding of the proposed method, the preliminaries and problem formulation are provided in this section. Let $G = (\mathbb{V}, \mathbb{E})$ be a network where \mathbb{V} is the set of vertices containing n nodes and \mathbb{E} is the set of m edges. The set of all connected node pairs in G is defined as \mathbb{P} (two nodes are connected if and only if there is a path from one node to the other). If all nodes in the network G are connected, then the cardinality of \mathbb{P} , i.e., $T = |\mathbb{P}|$, is equal to $(n - 1)n/2$. The network G can also be represented with a corresponding adjacency matrix A where an element a_{ij} is ‘1’ or ‘0’ indicating that node i is directly connected to node j or not, respectively. The vector $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_i, \dots, \mathbf{x}_n\}$ of decision variables is used to indicate the partition of the nodes. Particularly, node i is a core node if $\mathbf{x}_i = 1$, otherwise the node is a periphery node with $\mathbf{x}_i = 0$. Let $\mathbb{V}_c(\mathbf{x})$ denote the core nodes subset, $\mathbb{E}_c(\mathbf{x})$ denote the edge subset that connects the nodes in $\mathbb{V}_c(\mathbf{x})$, and $\mathbb{P}_c(\mathbf{x})$ be the pairs of nodes in $\mathbb{V}_c(\mathbf{x})$.

To illustrate the CPS, an idealized CPS network of 16 nodes is depicted in Fig. 1 with an accurate partition $= \{1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$. According to the definition of CPS, in a CPS 1) the core nodes are all connected to each other, 2) each periphery node is connected to all core nodes, and 3) the periphery nodes are disconnected to each other. Unfortunately, in reality, the CPS might not be ideal. A relatively relaxed definition should be applied, i.e., 1) the core nodes are densely inter-connected, 2) the periphery nodes are sparsely connected to core nodes, and 3) the periphery nodes are independent with each other. As the example shown in Fig. 2, no perfect CPS exists in the network. Yet, according to the relaxed definition, the first four nodes as highlighted in Fig. 2(b) can be identified as core nodes, and the others as periphery nodes.

Note that the core-attachment structure detection (CASD) [38] in protein-protein interaction networks is closely related to CPSD. However, the definition of core-attachment structure (CAS) is subject to the specific features of biological protein complexes and stricter than that of CPS. CAS is actually a subset of CPS. Since CAS is closer to a complete subgraph, CASD is usually based on clique or k-cliques detection, e.g., Zhou et al. proposed a clique percolation method (CPM) to detect CASs in [44]. CPM generates a maximum number of cliques for comparison to determine the cluster cores. With respect to CASD, CPSD considered in this paper can detect a wider range of core-like structural models.

To evaluate how likely a network containing a CPS, three commonly used CPS metrics are introduced as follows. For

the sake of convenience, the nomenclature of the problem formulation and metrics definitions is provided in Table 1.

3.1 Block model based metric

One of the most commonly used CPS metrics is the block model [2]. Given a partition \mathbf{x} of a network, the adjacency matrix of the ideal CPS can be generated. The adjacency matrix can be used as the standard to measure how close the current network partition is to a CPS. As shown in Fig. 2, the adjacency matrix of the corresponding ideal CPS for partition $\mathbf{x} = \{1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$ can be generated as Fig. 1(b). By evaluating the similarity between the real adjacency matrix in Fig. 2(c) and the ideal one in Fig. 1(b), we can know how likely the network partition is a CPS. In this way, a normalized block model metric $\rho(\mathbf{x})$ is defined as follows:

$$\rho(\mathbf{x}) = \frac{\|A \cdot \tilde{A}(\mathbf{x})\|_1}{\|\tilde{A}(\mathbf{x})\|_1}. \quad (1)$$

where A is the adjacency matrix of the current network, $\tilde{A}(\mathbf{x})$ indicates the adjacency matrix of the ideal CPS given \mathbf{x} , and $\|\cdot\|_1$ calculates the 1-norm of the matrix, i.e., the number of ‘1’s in the matrix. The normalized block model metric $\rho(\mathbf{x}) \in [0, 1]$ computes the pattern similarity between the current network partition and the ideal CPS. The larger the value is, the more likely the current network partition identify a CPS. A maximum value 1 indicates a perfect CPS detection. Note that the normalized block model metric is extended from the original metric proposed in [2] to enable the comparison between CPSs with different number of core nodes.

3.2 Null model based metric

The second CPS metric is the null model based metric. A null model refers to a random network without any structures. A higher similarity between a network and a null model indicates a lower possibility of the identification of a CPS. A randomly generated configuration network that preserves all nodes’ degree can be regarded as a null model (as shown in Fig. 3). To generate a null model, one can pass the degree sequence of the network to the *configuration_model* function in *NetworkX* package [25]. The null model based CPS metric Q_{cp} [17] is defined as follows:

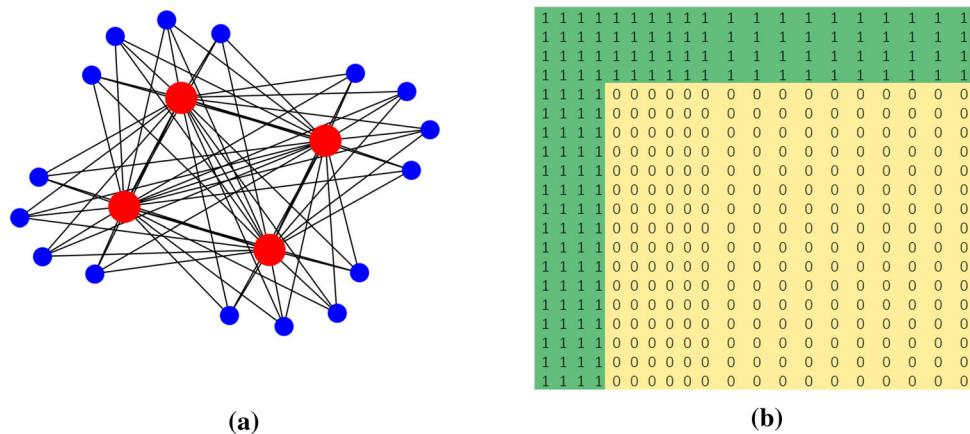
$$Q_{cp}(\mathbf{x}) = \frac{\sum_{i=1}^M \|(A - \hat{A}_i) \cdot \tilde{A}(\mathbf{x})\|_1}{M}, \quad (2)$$

where A is the adjacency matrix of the current network, $\tilde{A}(\mathbf{x})$ indicates the adjacency matrix of the ideal CPS given \mathbf{x} , and $\|\cdot\|_1$ calculates the 1-norm of a matrix. \hat{A}_i indicates the

Table 1 Parameter Notations

Notation	Definition
G	A network
\mathbb{V}	The vertices in the network G
\mathbb{E}	The edges in the network G
\mathbb{P}	The node pairs in the network G
\mathbf{x}	A decision vector of a CPS partition
Ω	The decision space
$\mathbb{V}_c(\mathbf{x})$	The vertices in the core subnetwork based on the CPS partition \mathbf{x}
$\mathbb{E}_c(\mathbf{x})$	The edges in the core subnetwork based on the CPS partition \mathbf{x}
$\mathbb{P}_c(\mathbf{x})$	The node pairs in the core subnetwork based on the CPS partition \mathbf{x}
$G(\mathbf{x})$	A core-periphery network G with the CPS partition \mathbf{x}
$A(\mathbf{x})$	The adjacency matrix of $G(\mathbf{x})$
$\tilde{G}(\mathbf{x})$	An ideal core-periphery network of G based on the CPS partition \mathbf{x}
$\tilde{A}(\mathbf{x})$	The adjacency matrix of $\tilde{G}(\mathbf{x})$
\hat{G}	A randomly generated configuration model that preserves each node's degree of G
\hat{A}	The adjacency matrix of \hat{G}
N	The number of nodes in G
T	The number of node pairs in G
M	The number of generated configuration models
L	The shortest path length between a pair of nodes
$\ddot{\mathbb{V}}$	The vertices in G arranged in reverse order of closeness centrality
$C(G)$	The network capacity of G
\overline{G}_k	Deleting 1 to k -th nodes of G
ΔC_k	Dropped network capacity after deleting the k -th node
e	The ratio of removed nodes to all nodes
r	The ratio of dropped network capacity to initial network capacity

Fig. 1 An ideal core-periphery pattern with partition $\mathbf{x} = \{1, 1, 1, 1, 0\}$: (a) the ideal CPS pattern with the first four nodes (red nodes) highlighted as core nodes, (b) is the adjacency matrix of the network



adjacent matrix of the i -th randomly generated null models $\hat{A} = [\hat{A}_1, \hat{A}_2, \dots, \hat{A}_i, \dots, \hat{A}_M]$, where M is the number of null models. The $Q_{cp}(\mathbf{x})$ metric evaluates the dissimilarity between a given partition and the null model. The smaller $Q_{cp}(\mathbf{x})$ value is, the partition is less likely to be a CPS.

3.3 Network capacity based metric

The third CPS metric is defined based on network capacity [6]. Before introducing the metric, some basic definitions are provided as follows. The network capacity of a network G ,

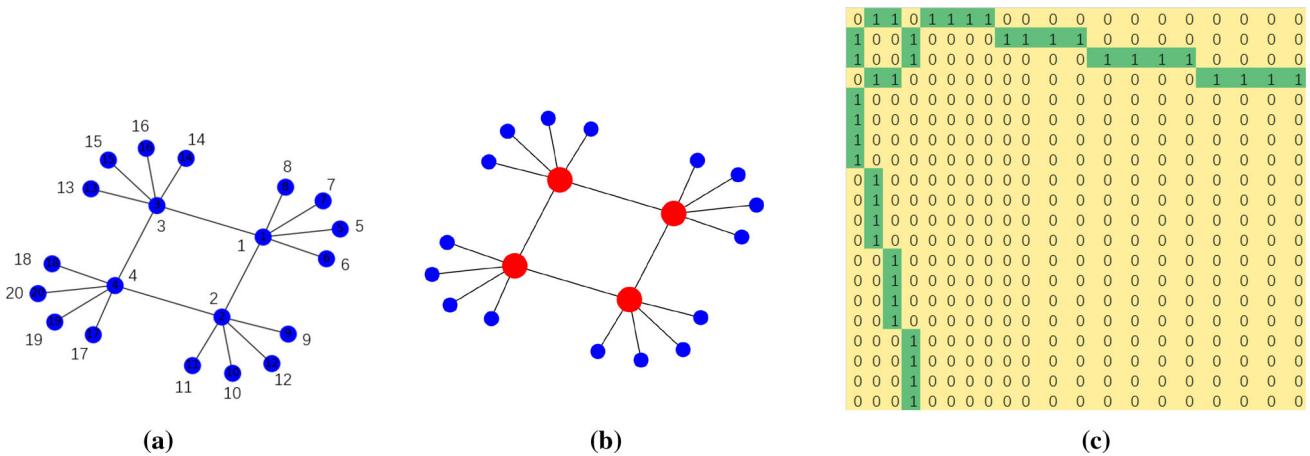


Fig. 2 Relaxed CPS partition (a) a network with no perfect CPS. (b) a CPS pattern with $\mathbf{x} = \{1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\}$. (c) is the adjacency matrix of the network

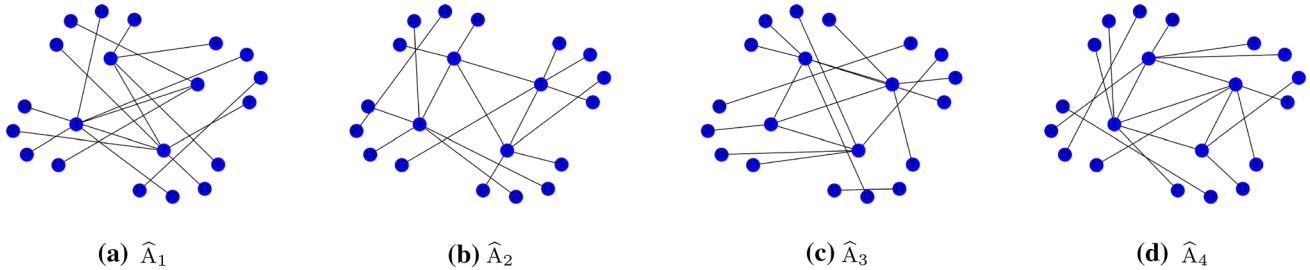


Fig. 3 A group of null models. (a) to (d) are the randomly generated configuration networks which preserve the degree or its mean value of each node in G

i.e., $C(G)$, is defined as

$$C(G) = \sum_{i=1}^T \frac{1}{L_i}, \quad (3)$$

where T is the total number of connected node pairs in G , i.e., $T = |\mathbb{P}|$, and L_i is the length of the shortest path between the i -th connected node pair in \mathbb{P} .

Based on the network capacity, a core-periphery network quantitative criterion [6] is defined as:

$$\sum_{k=1}^{e*n} \Delta C_k \geq r * C(G), \quad e, r \in [0, 1], \quad (4)$$

where the left part is the total dropping of network capacity (ΔC_k defined later) after deleting the top $e * n$ nodes according to the closeness centrality. The two coefficients e and r are set to 0.5 and 0.9 in [6], which means that the network contains a CPS if the network capacity drops more than 90% on the deletion of the 50% nodes closest to the network center. This setting can be utilized to determine whether a network contains valid CPSs, because the traditional metric value only indicates a higher or lower similarity and cannot judge whether the CPS partition is valid by the value itself.

The closeness centrality of a node u in a network G is defined as

$$\Upsilon(u) = \frac{1}{\sum_{v \in \mathbb{V}} d_G(u, v)}, \quad (5)$$

s.t. $(u, v) \in \mathbb{P}$.

where $d_G(u, v)$ denotes the length of the shortest path between two nodes u and v in a graph G . The length is counted only if the two nodes u and v are connected, i.e., $(u, v) \in \mathbb{P}$. The closeness centrality of a node u is the reciprocal of the shortest distance from u to all other reachable nodes. If u cannot reach any other nodes then $\Upsilon(u) = 0$. A greater closeness centrality of a node indicates the node is closer to the center of the graph.

To calculate ΔC_k , the nodes in \mathbb{V} are firstly sorted in descending order in terms of closeness centrality and denoted as $\ddot{\mathbb{V}}$. ΔC_k is the dropping of network capacity after the sequential deletion of the k -th node in $\ddot{\mathbb{V}}$, i.e.,

$$\Delta C_k = C(\overline{G}_{k-1}) - C(\overline{G}_k), \quad (6)$$

where \overline{G}_k indicates the remaining network after removing the first k nodes in $\ddot{\mathbb{V}}$.

3.4 MOP for CPSD

The previous three metrics work relying on prior knowledge of the number of core nodes, the perfect model, and/or optimal parameter settings, which might not be available in advance and therefore result in inaccurate detection. In this subsection, we propose to solve CPSD in a multi-objective perspective. To enhance the CPSD accuracy, we identify optimal CPS partition in terms of two conflicting objectives, namely, core-node size and core node capacity. Particularly, the MOP is formulated as follows:

$$\min F(\mathbf{x}) = (f_1(\mathbf{x}), -f_2(\mathbf{x}))^T \quad \text{s.t. } \mathbf{x} \in \Omega,$$

$$f_1(\mathbf{x}) = \|\mathbf{x}\|_1 \quad (7)$$

$$f_2(\mathbf{x}) = C(G) - C(G - \bigsqcup_{\mathbf{x}_i=1} i),$$

where $f_1(\mathbf{x})$ calculates the number of core nodes in the candidate partition \mathbf{x} and $f_2(\mathbf{x})$ denotes the total network capacity loss after deleting the core nodes encoded in \mathbf{x} . Particularly, $C(G)$ is the network capacity of G defined in Eq. (3) and $G - \bigsqcup_{\mathbf{x}_i=1} i$ indicates the new network obtained by deleting the code nodes with $\mathbf{x}_i = 1$ from G . The network capacity loss of a single node i is defined as **node capacity**:

$$\Delta C(i) = C(G) - C(G - i), \quad (8)$$

where $G - i$ indicates the new network obtained by deleting node i from G . $f_2(\mathbf{x})$ can also be referred to as the **core-node capacity**.

According to the definition of CPS, a deletion of a core node tends to cause more reduction of the network capacity. Thereby, the possibility of a node i being a core node can be estimated by the dropping of the network capacity after the deletion of node i , i.e., the $\Delta C(i)$. By minimizing the number of core nodes while maximizing the core-node capacity, we can correctly identify the most likely core nodes in different resolutions (different number of core nodes). For example, the solutions of the MOP $F(\mathbf{x})$ for the network shown in Fig. 2 are depicted in Fig. 4. The user can choose different CPSs with different preference of the number of core nodes. This advantage of MOP makes up for the inaccuracy of traditional single-objective metrics.

On optimizing Eq. (7), a candidate partition \mathbf{x}_a is said to dominate another partition \mathbf{x}_b , denoted as $\mathbf{x}_a \prec \mathbf{x}_b$, if and only if $f_1(\mathbf{x}_a) \leq f_1(\mathbf{x}_b)$, $f_2(\mathbf{x}_a) \geq f_2(\mathbf{x}_b)$ and $F(\mathbf{x}_a) \neq F(\mathbf{x}_b)$. A solution partition $\mathbf{x}^* \in \Omega$ is called Pareto optimal solution if it is not dominated by any other solution. The Pareto optimal set (PS) is denoted as

$$PS := \{\mathbf{x}^* | \neg \exists \mathbf{x} \in \Omega, \mathbf{x} \prec \mathbf{x}^*\}, \quad (9)$$

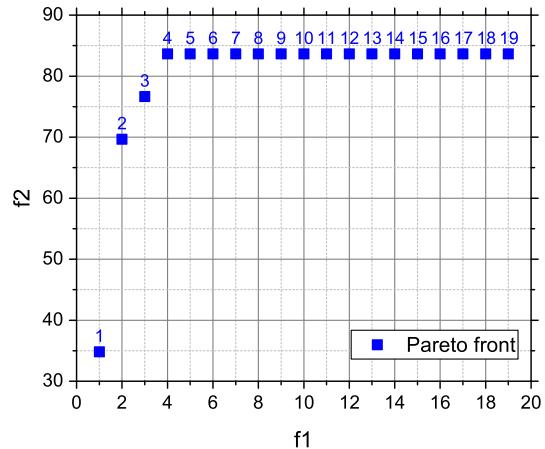


Fig. 4 The corresponding partition PF of the network shown in Fig. 2

and the Pareto front (PF) of the Pareto optimal set is defined as

$$PF := \{F(\mathbf{x}) | \mathbf{x} \in PS\}, \quad (10)$$

An example PF consisting of 12 candidate partitions, i.e., from P_1 to P_{12} , is shown in Fig. 4. The x-axis of PF covers a range of core size number. The y-axis indicates the obtained optimal accumulated core-node capacity (normalized with $C(G)$ for the sake of display) corresponding to the core size.

It should be emphasized that a real core-periphery network has a sharply dropped network capacity after the removal of core nodes, while the phenomenon does not occur in community network in that the remained nodes still have connections. The condition on sharply dropped network capacity ensures that the detected core nodes are from the real CPS and not the community. The judgment criterion on sharp network capacity dropping can be related to the heavy-tailed distribution of node capacity (detailed in Sect. 4.3). To select an appropriate representative partition from the PF to investigate the CPS, the knee point solution is a reasonable option [28], because the knee point solution can locate the greatest network capacity dropping on deleting the core nodes. For example, the knee point solution P_4 in the PF shown in Fig. 4 corresponds to the best partition of the network shown in Fig. 2(b).

It is worth highlighting that the proposed MOP does not optimize the aforementioned three metrics directly. Instead, the metrics are used to evaluate the solutions of the MOP. For example, Fig. 5 shows four partition samples. For the four partitions, Table 2 reports the metric values of the detected CPSs. The first two metrics, i.e., ρ and Q_{cp} , are quantitative measures on the partition quality, whereas the third metric, i.e., $\sum_{k=1}^{0.5*n} \Delta C_k \geq 0.9 * C(G)$ is a qualitative metric on where the network contains CPSs.

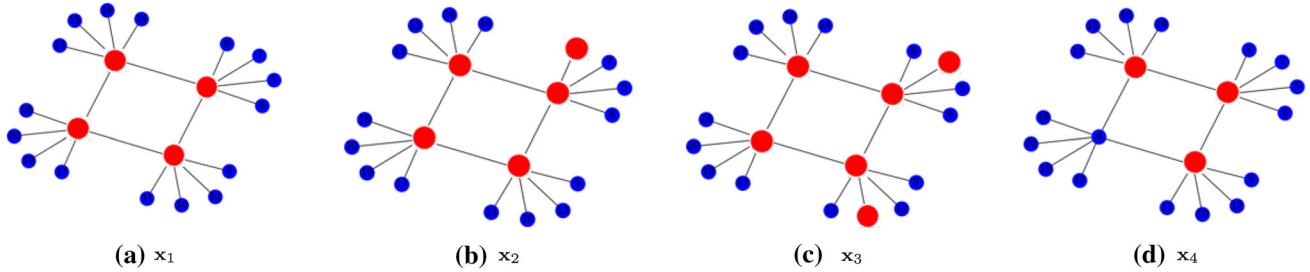


Fig. 5 Partition samples. (a) to (d) are the detected partitions of G where x_1 and x_4 are two non-dominated partitions

Table 2 Partition evaluation using different metrics

Partition	ρ	Q_{cp}	$\sum_{k=1}^{0.5*n} \Delta C_k \geq 0.9 * C(G)$	f_1	f_2
$x_1: \{1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$	0.2857	12.304	yes	4	83.6667
$x_2: \{1,1,1,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0\}$	0.2353	11.54	yes	5	83.6667
$x_3: \{1,1,1,1,0,0,1,0,0,1,0,0,0,0,0,0,0,0,0\}$	0.202	10.868	yes	6	83.6667
$x_4: \{1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0\}$	0.2593	9.2	yes	3	76.6667

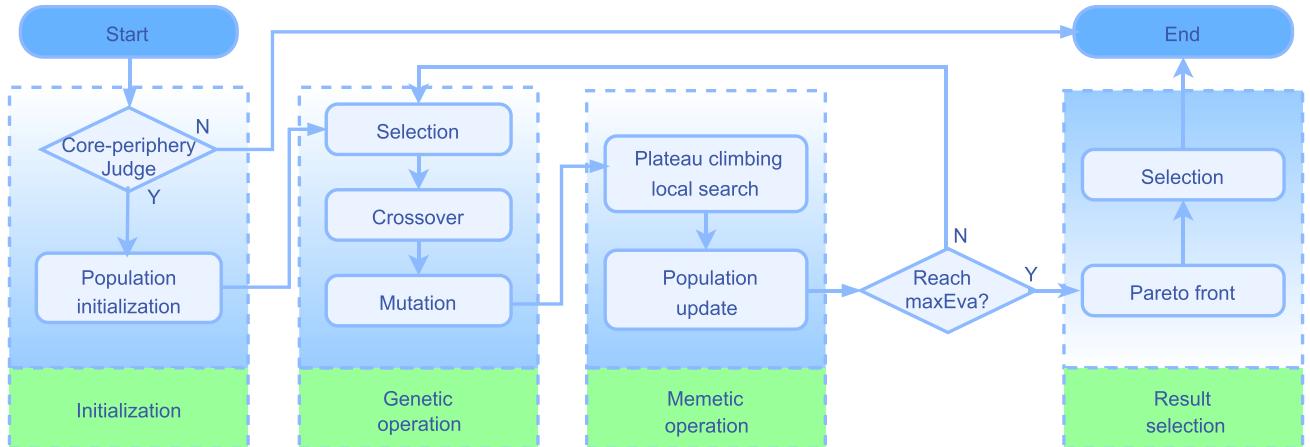


Fig. 6 The procedure of MOMA-PCLS: (1) In the initialization stage, the population is generated if the network is a real core-periphery network, and the nodes are sorted according to their node capacity. (2) In the genetic operation, selection, crossover, and mutation operators are used to evolve the population. (3) In the memetic operation stage, the

plateau-climbing local search is used to fine tune the solutions, and the population is updated if there is an improvement achieved. (4) If the termination condition is reached, a suitable solution in the Pareto front is selected to show the CPS

4 The proposed MOMA-PCLS

In this section, the details of the proposed MOMA-PCLS are provided. The flowchart of MOMA-PCLS is depicted in Fig. 6. MOMA-PCLS consists of four stages. At the first stage, the network is validated with Eq. (4) to test whether the network potentially contains a CPS. If the network passes the test, the nodes are sorted according to their node capacity defined in Eq. (8) and a population of solutions is randomly initialized. At the second stage, genetic operators including selection, crossover, and mutation operators are conducted to evolve the population. At the third stage, the PCLS incorporating the heavy-tailed distribution

of node capacity kicks in to fine tune the solutions. The second and third stages are repeated until the stopping criterion is met and the non-dominated solutions are output. At the last stage, MOMA-PCLS selects a knee point solution from the output solutions to show the CPS. The main design of MOMA-PCLS is the local search method. The main decision in PCLS is to identify the heavy-tailed cutting point. By sorting each node's capacity, the node with a maximum dropped NC is identified as the cutting point between the head part and tail part. Then, among the generated neighborhood individuals, the core nodes are more likely located at the head part, while the periphery nodes tend to be at the tail part. Based on the characteristics of the neighborhood, we

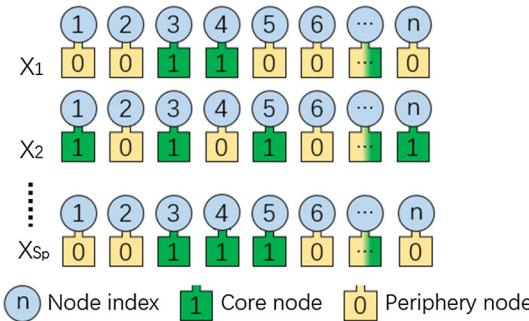


Fig. 7 Initialization: Generating a population of chromosomes. S_p is the number of chromosomes in a population. In a chromosome, each gene represents a network node with 1 indicating a core node and 0 indicating a periphery node

design a plateau climbing local search method to accelerate the convergence. The details of the algorithm are introduced below.

4.1 Initialization and fitness evaluation

In the initialization, the nodes are firstly sorted in descending order according to the node capacity defined in Eq. (8), such that the core nodes are more likely located in the front of the node set. The re-ordering can help the local search to reduce the search space as detailed in Sect. 4.3. Afterward, an individual population of size S_p with each individual encoding a candidate network partition is randomly generated as depicted in Fig. 7. Each individual chromosome is represented as a binary string containing n gene bits, i.e., one for each network node. A ‘1’(‘0’) bit indicates the corresponding node is a core(periphery) node. The fitness of an individual is evaluated based on Eq. (7).

4.2 Genetic operation

After population initialization, the population is evolved with genetic operation including selection, crossover, and mutation.

4.2.1 Selection

In the selection operator, the elitist retention model is used. In each generation, the S_p fitter chromosomes in the combination of parent population and offspring population survive to the next generation. Particularly, the individuals in the parent population are firstly grouped according to the $f_1(\mathbf{x})$ value, i.e., the number of core nodes, defined in Eq. (7). Then the individuals in each group are sorted in descending order according to the $f_2(\mathbf{x})$ value. Secondly, for each individual \mathbf{x}' in the offspring population, the selection operator works according to $f_1(\mathbf{x}')$, i.e.,

- 1) if the group $f_1(\mathbf{x}')$ does not exist in the parent population, the last parent individual in the largest group is replaced by \mathbf{x}' ;
- 2) if the group $f_1(\mathbf{x}')$ exists in the parent population and the last parent individual in this group is worsen than \mathbf{x}' in terms of $f_2(\mathbf{x})$, the last parent individual is replaced by \mathbf{x}' ;
- 3) otherwise, abandon \mathbf{x}' .
- 4) After a parent individual is replaced in a group, the group is sorted again for the next individual replacement.

The previous procedure is repeated until all offspring individuals are considered. The remaining parent population then goes through a new iteration of evolution. Note that since the best individual in each group is persevered, the non-dominated solutions are guaranteed to survive into next generation. The elitist retention model saves partitions in all detected f_1 values, while the traditional selection method based on crowd distance cannot retain all detected values.

4.2.2 Crossover

In each generation, $P_c * S_p$ individuals are randomly chosen to undergo the crossover operation, where P_c is the crossover probability and its value is set in $[0, 1]$. Different crossover operators such as single point, two-point, and uniform point crossover can be used in the proposed algorithm. In the single point operator, the crossover point is randomly selected, then the first half of a chromosome is recombined with the second half of the paired chromosome. In the two-point crossover, the bits are exchanged between the two randomly chosen points. In the uniform crossover, each bit is selected for crossover by tossing a coin. These crossover operators are illustrated in Fig. 8. The effects of using different crossover operators are shown in Sect. 5.3.

4.2.3 Mutation

In the mutation operation, each bit in an individual is selected for mutating with probability of P_m . Different mutation operators can be considered. For example, in a bit flip mutation, each selected bit is flipped; in an exchange mutation, a pair of randomly selected bits exchange their place; and in an adaptive mutation, a strategy of dynamically reducing the P_m is applied. Particularly, after 10% of total generations, the P_m is reduced by 10% of its initial value. The mutation operators are as shown in Fig. 9. The effects of using different mutation operators are also investigated in Sect. 5.3. An appropriate mutation probability helps to jump out of the local optima in the evolution process.

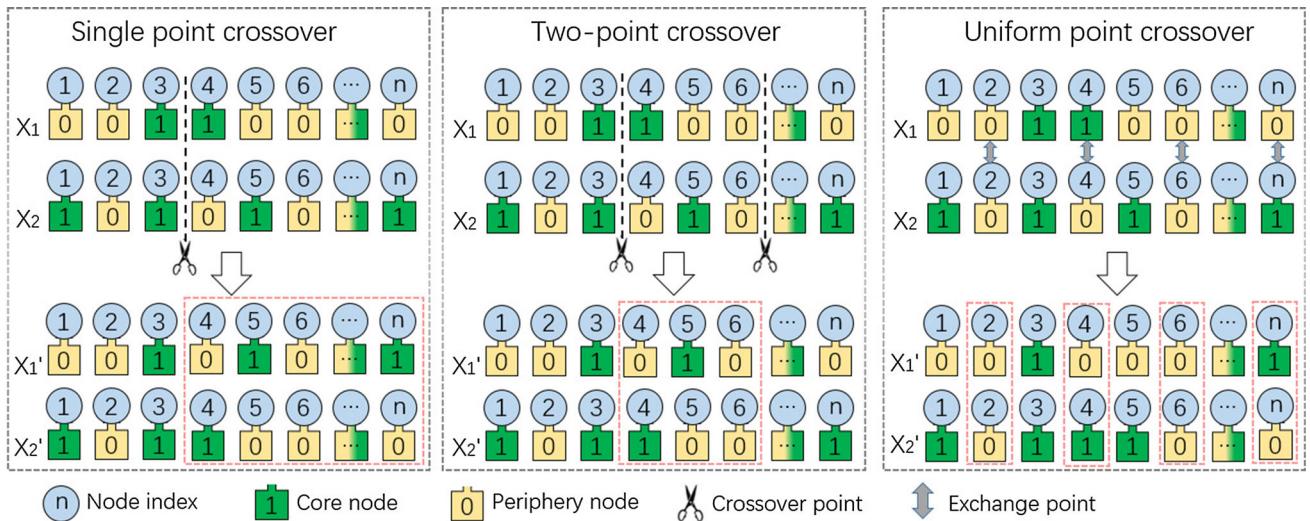


Fig. 8 Crossover: A pair of chromosomes x_1 and x_2 are selected, then a crossover operator is used to generate a pair of offspring chromosomes x_1' and x_2'

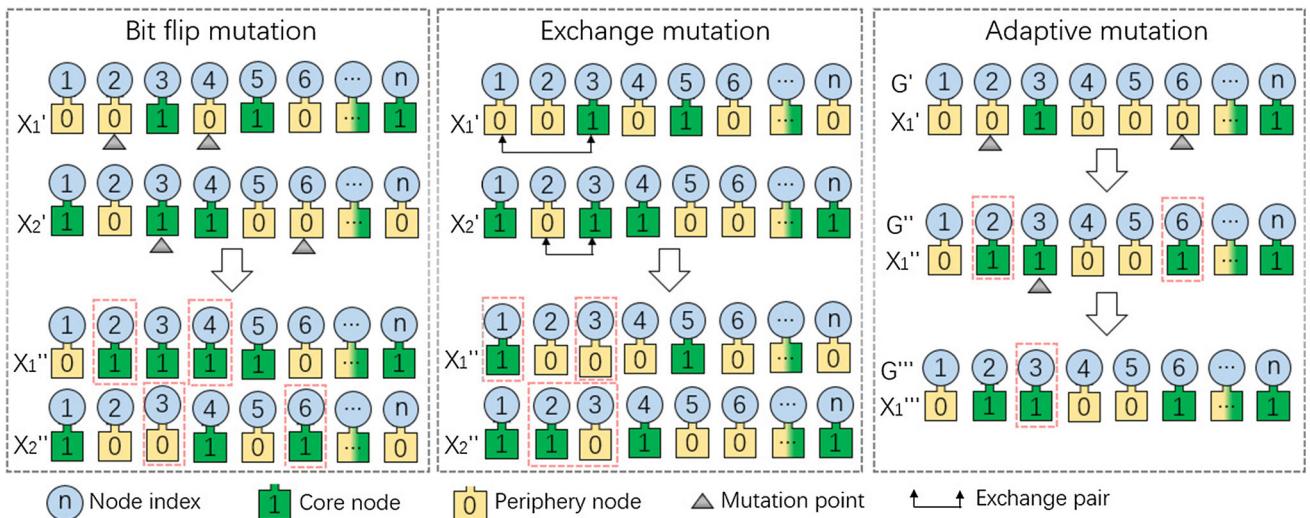


Fig. 9 Mutation: For the chromosomes x_1' and x_2' , a mutation operator is selected then changes its phenotype as x_1'' and x_2''

4.3 Memetic operation

Conventional evolutionary algorithms tend to suffer from slow convergence in local regions. Local search or memetic operation serves as a good solution to this issue by fine-tuning the individuals in local regions. In the memetic operation, we propose a PCLS method based on heavy-tailed distribution of the node capacity to speed up the evolution. A local search probability $P_l \in [0, 1]$ is imposed to determine the proportion of individuals to undergo local search.

As stated in the initialization, the nodes are sorted in descending order according to their node capacity. Since the core nodes tend to have significantly larger node capacity than that of the periphery nodes. The distribution of node capacity is very likely heavy-tailed as shown in Fig. 10(a).

The local search firstly identifies the cut point of the heavy-tailed distribution to separate the potential core nodes and periphery nodes. Next, a new PCLS method improved from the traditional hill-climbing method is used to deal with the combinatory search space. The details of the PCLS are provided as follows.

4.3.1 Heavy-tailed cutting method

The heavy-tailed distribution of node capacity can be used to narrow down the scope of the local search. The node capacity reflects the proportion network capacity of a node. In a CPS, the core nodes occupy an essential part of node capacity, whereas the periphery nodes occupy a much smaller portion of the node capacity. Therefore, the heavy-tailed distribu-

Fig. 10 Heavy-tailed distribution of node capacity. (a) heavy-tailed distribution of node capacity against sorted node index, (b) node capacity dropping on the removal of the network nodes. According to the heavy-tailed distribution, the search space of core nodes can be reduced to the head part

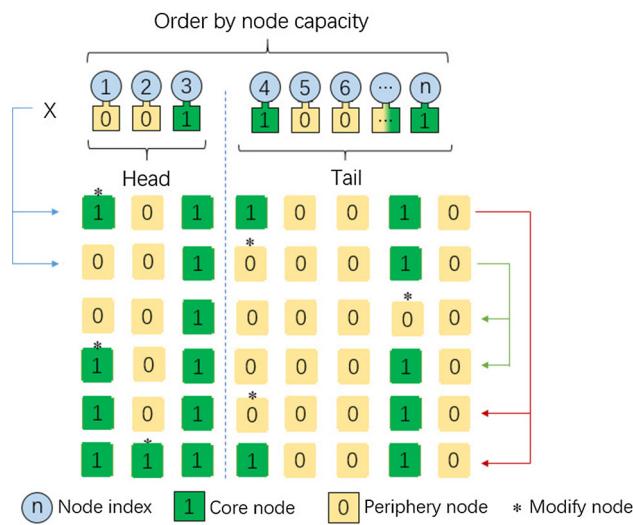
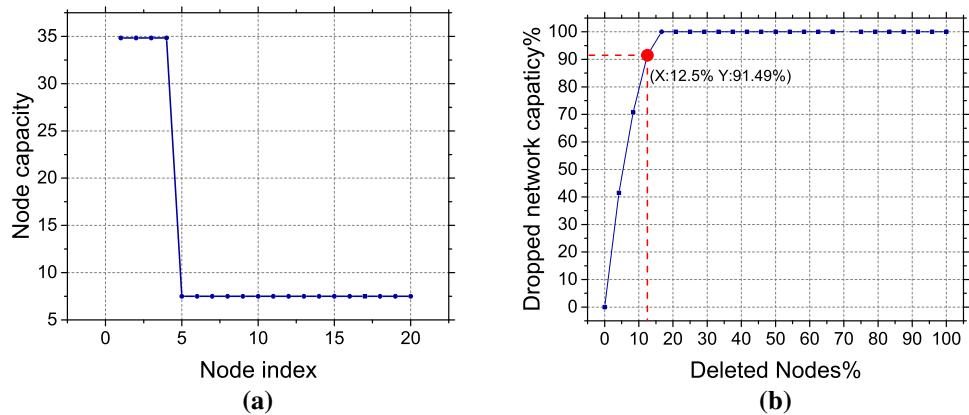


Fig. 11 The generation of neighborhood in local search. The nodes in the head part are more likely to be core nodes, so a neighborhood search space is generated at the head part to improve search efficiency

tion of the node capacity can effectively reflect the core and periphery nodes as shown in Fig. 10(a). For most realistic core-periphery networks, about 90% of the network capacity should be contained in no more than 50% of the network nodes [6]. Hence, the cutting point of the core part and the periphery part can be defined in the place of the 90% node capacity dropping as depicted in Fig. 10(b).

In PCLS, given a solution individual \mathbf{x} and the cutting point α , a neighbor individual \mathbf{x}' is generated according to the following steps.

- Step 1 : Let $\mathbf{x}' = \mathbf{x}$;
- Step 2 : Randomly select a position i in the chromosome;
- Step 3 : If $i \leq \alpha$ and $\mathbf{x}_i == 0$, then $\mathbf{x}'_i = 1$ and output \mathbf{x}' ;
- Step 4 : If $i > \alpha$ and $\mathbf{x}_i == 1$, then $\mathbf{x}'_i = 0$ and output \mathbf{x}' ;
- Step 5 : Repeat Step 2.

For a selected solution individual \mathbf{x} , $\lfloor \sqrt{S_p} \rfloor$ neighbors are firstly generated according to the previous procedure ($\lfloor \cdot \rfloor$ is a floor function to obtain the greatest integer less than or equal to the given input number), and then for each generated neighbor, another $\lfloor \sqrt{S_p} \rfloor$ solutions are generated similarly. Therefore, it results in about S_p new generated individuals to form the neighborhood of \mathbf{x} . α is the cutting point, namely, the number of points reach the largest dropped NC in heavy-tailed distribution.

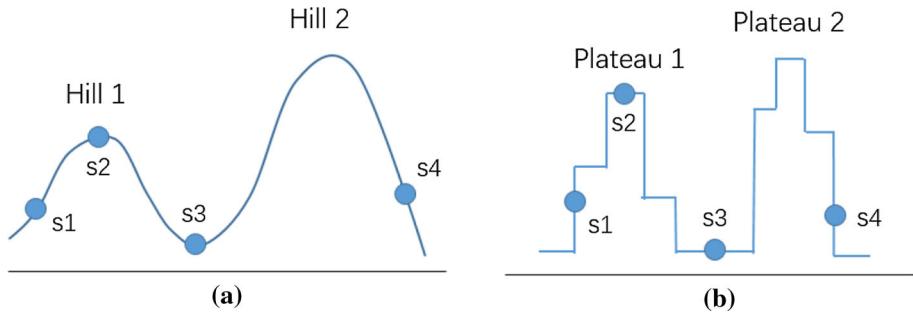
As illustrated in Fig. 11, given an individual \mathbf{x} and suppose $S_p = 6$, two neighbor individuals of \mathbf{x} are firstly generated (the 2nd and 3rd chromosomes), and then four new individuals at the bottom (two for each neighbor) are generated sequentially, resulting in a six-individual neighborhood of \mathbf{x} .

4.3.2 Plateau climbing local search

After the neighborhood search space is determined, a local search method is conducted to effectively search the neighborhood in the hope of improving the individual. The core-periphery network partition is a vector of n discrete-valued decision variables. The neighborhood search space of the vector is composed of many step intervals. Traditional hill-climbing local search methods are more prone to stagnate in a discrete interval. As depicted in Fig. 12(a), a traditional hill-climbing method generates some starting points and then continuously searches for the larger value in the direction of increasing the fitness value by comparing the fitness values with adjacent locations. If there are many interval segments with the same value in the search space, the hill-climbing method may tend to fall into the local optima. As shown in Fig. 12, with traditional local search, all solutions in the continuous search space (shown in Fig. 12(a)) can reach their corresponding local optimal, whereas in discrete stepped search space (shown in Fig. 12(b)), solutions s1, s3 and s4 can hardly get to the local optima.

To solve the aforementioned issue, we design the PCLS method according to the characteristics of the data, as illus-

Fig. 12 Hill-climbing in different search spaces. (a) continuous search space, (b) discrete search space. The traditional hill-climbing method is suitable for the search space where the fitness value is continuously changing, and the plateau-climbing method is suitable for the search space where the fitness values present discrete steps



Algorithm 1: Plateau-climbing local search method

```

Input: Population  $\mathcal{P}$ , local search probability  $P_l$ , local search range  $\beta$ , the number of start points  $N_s$ 
Output: Updated population  $\mathcal{P}_{update}$ 
1  $\mathcal{P}_{subset} \leftarrow \text{SelectSubset}(\mathcal{P}, P_l);$ 
2 for each chromosome  $c_i$  in  $\mathcal{P}_{subset}$  do
3    $C_i^{neighbor} \leftarrow \text{Combination}(c_i);$ 
4   for each chromosome  $c_j$  in  $C_i^{neighbor}$  do
5      $S \leftarrow \text{RandomStartpoint}(N_s);$ 
6     for each startpoint  $S_t$  in  $S$  do
7        $S_t^{init} \leftarrow S_t;$ 
8        $S_t^{best} \leftarrow S_t;$ 
9       while not exceed Boundary do
10         $S_t^{next} \leftarrow \text{nextLocation}(S_t);$ 
11        if  $F(S_t^{next}) < F(S_t^{best})$  then
12           $| S_t^{best} \leftarrow S_t^{next}$ 
13        if  $|F(S_t^{best}) - F(S_t^{init})| * \beta <$ 
14           $|F(S_t^{next}) - F(S_t^{init})|$  then
15           $| \text{break};$ 
16        end
17         $c_t \leftarrow \text{getLocation}(S_t^{best});$ 
18      end
19      if  $c_t \prec c_j$  then
20         $| c_j \leftarrow c_t$ 
21      end
22    end
23    if  $c_j \prec c_i$  then
24       $| c_i \leftarrow c_j$ 
25    end
26  end
27  $\mathcal{P}_{update} \leftarrow \text{Update}(\mathcal{P}, \mathcal{P}_{subset})$ 

```

trated in Algorithm 1. First, the PCLS method generates some starting points N_s like the traditional hill-climbing method. For each starting point, the search begins with a direction, and it would change to an opposite direction when the previous direction cannot be further exploited. Unlike traditional hill-climbing methods, the PCLS method allows for continued search after a local optimum value is obtained, which might cause dropping of the fitness value. The tolerant dropping range is controlled by parameter β . A reasonable setting of β is $\beta = 2$ through our test. As such, the local search is possible to escape from one step interval to another better one. The PCLS method accelerates by increasing the forward search

step if the next value is the same. Finally, if a better individual in terms of dominance is detected in the neighborhood, the population is updated. The detailed dominant comparison is described in Sect. 3.4.

5 Experiments

In this section, experiments on six networks are conducted to demonstrate the effectiveness of the proposed MOMA-PCLS. These networks include social, physical, economic, and biological networks that have typical core-periphery structures. MOMA-PCLS is compared with other state-of-the-art CPSD methods, including LapSgnCore [5], Rombach [30], and KM_config [17], and two baseline multi-objective evolutionary algorithms, i.e., MOGA (multi-objective genetic algorithm without local search) and MOMA (multi-objective memetic algorithm that is combination of MOGA and traditional hill-climbing local search). The effects of the parameters of MOMA-PCLS and PCLS are also investigated. Since the results of each run of the evolutionary algorithms have certain randomness, the results of these algorithms are averaged over 20 independent runs. To ensure a fair comparison, the evolutionary algorithms are terminated if the same maximum number of fitness evaluations is reached.

5.1 Core-periphery network datasets

Six representative core-periphery network datasets are selected to test the proposed algorithm. The details of the datasets are provided in Table 3. To verify whether the networks contain CPSs, the block model based metric is applied at a coarse-grained level as shown in Fig. 13. Using the network capacity based metric, a fine-grained level judgment is conducted in Fig. 14.

The karate club network is a commonly used social network that has been demonstrated to have CPSs in [17]. It contains 34 nodes and 78 edges, where nodes represent the members of a karate club, while edges represent the existence of a relationship between members. The snowflake network is a synthesized network according to the natural form of

Fig. 13 Block model metric.
The Block model metric method first generates adjacency matrices after arranging the network nodes in reverse order according to the closeness centrality, and then compares the block pattern with the idealized CPS model

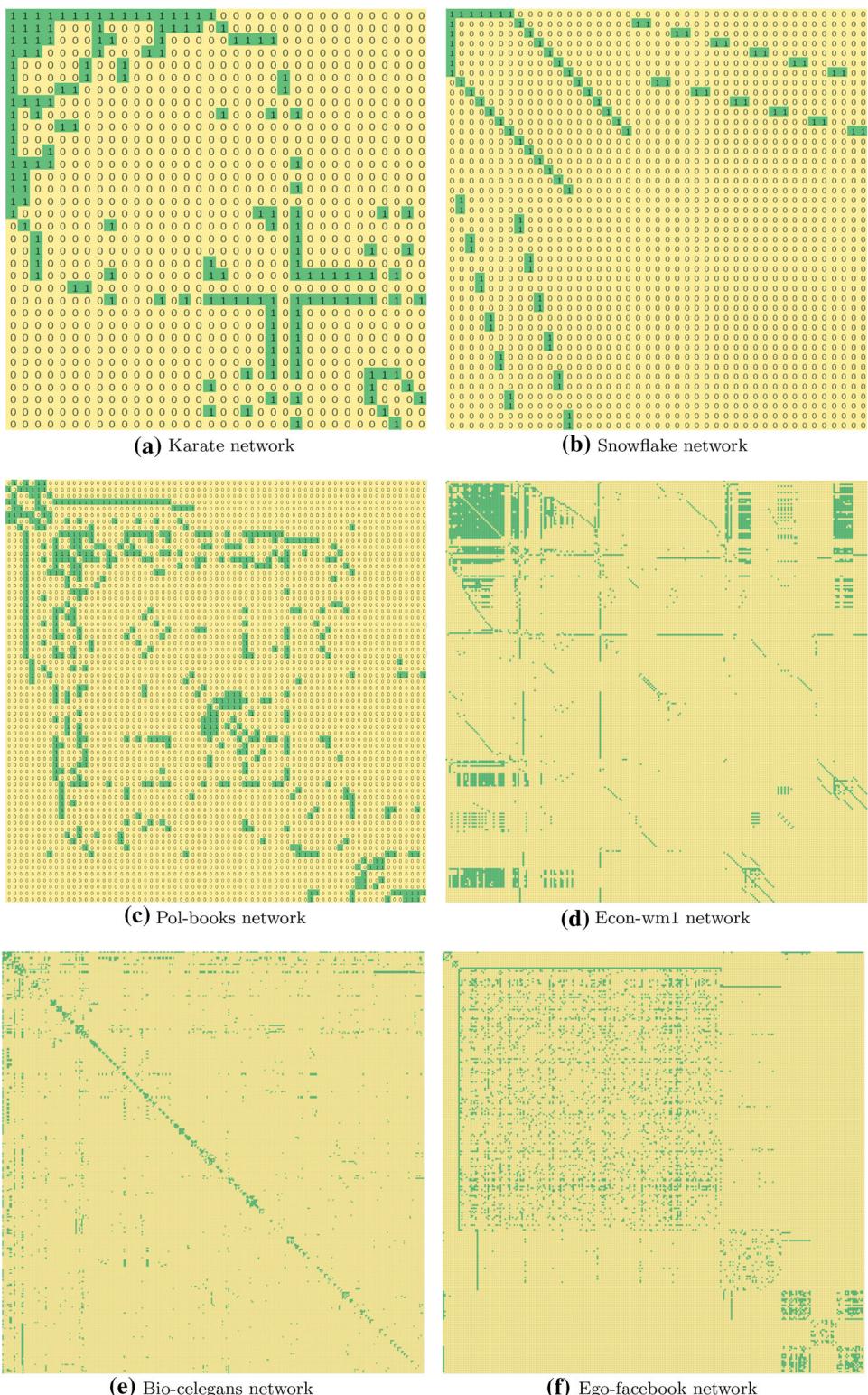


Table 3 The test network dataset

Dataset	Description	Nodes	Edges	Degree	
				Avg.	Max.
Karate	Social network of member relationships in the karate club	34	78	4.59	17
Snowflake	The physical structure of natural snowflake	43	42	1.95	6
Pol-books	The purchasing relationships in a bookseller	105	441	8.4	25
Econ-wm1	Economical problem network in the realworld	258	2,389	18.52	106
Bio-celegans	Biological network of metabolic reactions	453	2,025	8.94	237
Ego-facebook	An example of the Facebook social network	4039	88,234	43.69	1045

snowflake structure that serves as a fantastic core-periphery network. The core nodes occupy the critical network capacity, and the periphery nodes are only attached to core nodes. The Pol-books network represents the purchasing relationship between customers and US political books at an online bookseller [31]. The Econ-wm1 network is an economic network [31] contains 258 nodes and 2, 389 edges. Its nodes are transaction entities, and edges are transaction relationships. The Bio-celegans network contains 453 nodes and 2, 025 edges. It is a biological network of *C. elegans*, where nodes are substrates and edges are metabolic reactions [31]. The Ego-facebook data is a social network contains 4, 039 nodes and 88, 234 edges [34].

According to the block model based metric, the two small networks in Fig. 13(a) and Fig. 13(b) show a clear core-periphery pattern. Fig. 13(d) to Fig. 13(f) are mesoscale networks, and they have many small or large patterns of CPSs. On using the network capacity based metric as depicted in Fig. 14, when the percentage of deleted network nodes reached 50%, the dropped network capacity are all beyond the 90% threshold. The judgment results are in accordance with Eq. (4). As depicted in Fig. 15, these six core-periphery networks all show heavy-tailed distribution of node capacity.

5.2 Effects of the parameters

In MOMA-PCLS, the settings of the probability parameters, i.e., P_c , P_m , and P_l , could affect the algorithm performance. The crossover probability P_c determines the number of chromosomes participating in the crossover; the mutation probability P_m determines the number of mutating bits in each chromosome. Both P_c and P_m affect the global search ability. The local search probability P_l determines the number of chromosomes for local search.

Table 4 reports the number of fitness evaluations for MOMA-PCLS algorithm to obtain the best knee point solution on the Karate network with different parameters, where the exact CPS is known. In the test, each group of parameters is run 20 times. Then, the mean and standard deviation of each group is calculated, and the Friedmen Test is used

to determine whether or not there is a statistically significant difference between all groups. In the first test of P_c , the setting of S_p remains consistent with the number of network nodes, a larger E_{max} value is used to ensure convergence within the specified number of evaluations. The initial values of P_m and P_l are set to 0 to prevent interference. In the test result, the group $P_c = 0.9$ used the least number of evaluations to reach the optimal solution, and the Friedmen Test's P-value shows that all groups are significantly different. In the second test of P_m , the optimal selection of P_c is reserved, and the other parameters continue to be consistent. Similarly, the third test of P_l adopts the above method. The results show that the MOMA-PCLS algorithm obtains the best performance with $P_c = 0.9$, $P_m = 0.1$, and $P_l = 0.2$.

In conventional optimization problems, the number of decision variables is relatively small, and the effect of population size S_p on optimization is not necessarily noticeable. However, in CPSD, the number of decision variables is equal to that of network nodes. The population size could significantly affect the experiment performance and results. Therefore, when the network scale is small, S_p can be set to be roughly the same as n or a size S_{max} that does not affect program performance. When the network scale is relatively large, S_p is set to S_{max} to prevent performance degradation. The E_{max} can be set to $100 * n$ or a value that can ensure convergence.

5.3 Effect of evolutionary operators and local search methods

This subsection investigates the effects of using different crossover operators and mutation operators. The comparison result is shown in Fig. 16. We choose the Karate network as the representative for illustration. The crossover operators contain the single point, two-point and uniform point methods with a group of fixed parameters, where S_p is 50, E_{max} is 5000, P_c is 0.9 and P_m is 0.1. When approaching the minimized optimization objective f_1 , the uniform point operator can detect the maximum f_2 value. In the same way, we compare three different mutation operators and find that adaptive mutation operator is slightly better than the other two muta-

Table 4 Comparison of various probability parameters set. When the parameters are set from 0.1 to 0.9, the number of fitness evaluations required by the MOMA-PCLS algorithm to search for the optimal knee point solution

P_c test	P_c	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Avg.	4100.75(+)	2850.25(+)	2033.05(+)	2214.2(+)	1789.15(+)	1563.9(+)	1284.8(+)	1142.75(+)		924.05
Std.	±1092.2189	±2086.6418	±1299.2543	±1880.617	±931.3723	±1678.2356	±856.1343	±548.1266	±409.0487	
Test parameters: ($S_p=50$, $E_{max}=10000$, $P_c=[0.1, 0.9]$, $P_m=0$, $P_l=0$)										
P_m test	P_m	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Avg.	1959.8	2727.75(+)	3828.55(+)	3265.65(+)	3580.35(+)	3742.45(+)	3227.2(+)	3734.4(+)		4294.65(+)
Std.	±908.5903	±1435.4597	±2212.9468	±1972.6688	±2393.0488	±1981.937	±1868.6261	±2432.132	±2990.4511	
Test parameters: ($S_p=50$, $E_{max}=10000$, $P_c=0.9$, $P_m=[0.1, 0.9]$, $P_l=0$)										
P_l test	P_l	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Avg.	3031.25(+)	2725.3	3959.6(+)	4354.05(+)	4240.85(+)	3264.2(+)	4712.25(+)	4097.05(+)		4996.8(+)
Std.	±2145.1809	±1141.7888	±1804.5663	±2180.2727	±2373.834	±1145.3334	±2047.1687	±2449.0941	±1848.2252	
Test parameters: ($S_p=50$, $E_{max}=10000$, $P_c=0.9$, $P_m=0.1$, $P_l=[0.1, 0.9]$)										

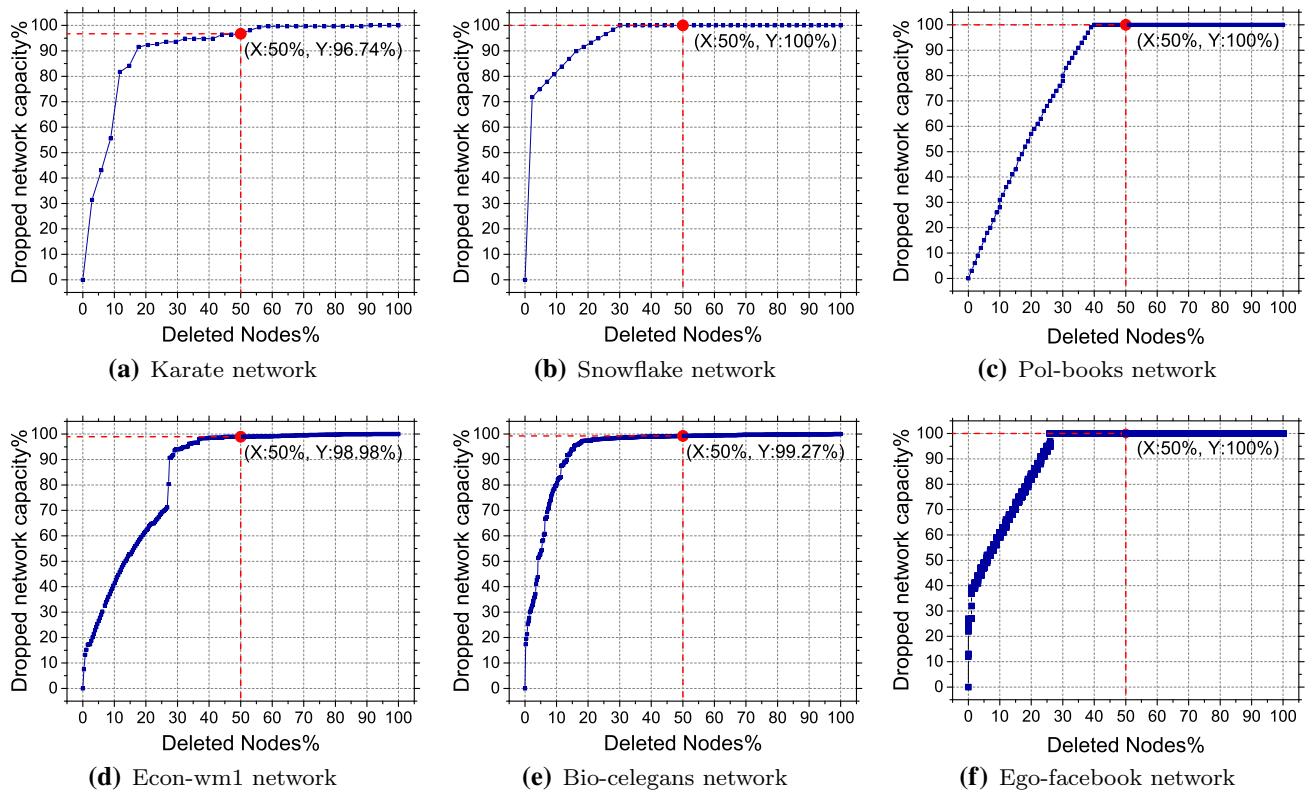


Fig. 14 Network capacity metric. The network capacity metric method first arranges the nodes in reverse order according to the closeness centrality, and then delete the nodes in the network in turn. If deleting 50%

of the nodes, the overall network capacity drops by more than 90%, it means that this network is a core-periphery network

tion operators. The exchange mutation detects nothing in the lower f_1 values.

This subsection also investigates the effects of using different local search methods, as shown in Fig. 17. The local search methods include the hill-climbing, TS, SA, and PCLS. We choose a larger network econ-wm1 for better illustration of the identified PF. It can be seen from the comparison results that the PF of PCLS is non-dominated by that of the other local search methods. Therefore, the performance of the PCLS method is relatively better than the other local search methods.

5.4 Comparison with state-of-the-art algorithms

The chosen algorithms for comparison with our MOMA-PCLS are divided into two types. One is the heuristic single-objective optimization algorithm based on the traditional ρ or Q_{cp} metric, containing the LapSgnCore, Rombach, and Km_config algorithms. For the detected partition of these algorithms, the missing metric results are supplemented for uniform comparison. The other type is multi-objective optimization algorithms based on the number of core nodes and network capacity metrics, including MOGA and MOMA.

5.4.1 Comparison with single objective algorithms

Table 5 reports the comparison of state-of-the-art CPSD algorithms and our MOMA-PCLS algorithm. The parameters setting in this comparison are ($S_p=50$, $E_{max}=5000$, $P_c=0.9$, $P_m=0.1$, $P_l=0.2$) for Karate, Snowflake, and Pol-books networks, and ($S_p=200$, $E_{max}=20000$, $P_c=0.9$, $P_m=0.1$, $P_l=0.2$) for Econ-wm1, Bio-celegans, and Ego-facebook networks. To validate the detected CPSs, the obtained solutions of the compared algorithms are evaluated with the block model metric ρ [2] and the null model metric Q_{cp} [17] as well as the detected f_1 and f_2 solutions. The result obtained by MOMA-PCLS is a knee point solution of the final solution set. The comparison results show that MOMA-PCLS obtains larger ρ and Q_{cp} values, and solution is non-dominated by the other heuristic algorithms in terms of (f_1, f_2). It should be noted that in the Snowflake network, the f_2 value of LapSgnCore, KM-config, and MOMA-PCLS is the same. The reason is that after deleting their core nodes, there are only discrete points in the network, so these f_2 values all equal to its original network capacity 266.25.

The visual comparison results can be intuitively shown in small-scale networks. As illustrated in Fig. 18, there are

Table 5 Comparison of single objective algorithms with MOMA-PCLS. The f_1 and f_2 values of LapSgnCore, Rombach, and KM_config are supplemented based on the detected CPS. The ρ and Q_{cp} of MOMA-PCLS are also supplemented for comparison. ‘-’ represents the program cannot obtain a valid result in a larger network

	LapSgnCore				Rombach				KM-config				MOMA-PCLS			
	ρ	Q_{cp}	f_1	f_2	ρ	Q_{cp}	f_1	f_2	ρ	Q_{cp}	f_1	f_2	ρ	Q_{cp}	f_1	f_2
Karate	0.1686	5.324	9	264.68	0.1333	4.784	12	251.57	0.1213	7.408	14	276.02	0.2186	13.004	6	254.1833
Snowflake	0.067	10.796	19	266.25	0.0877	3.744	8	243.75	0.0761	12.692	16	266.25	0.0897	14.608	13	266.25
Pol-books	0.068	87.6	44	1750.46	0.0723	57	37	1286.09	0.0717	33.8	73	2157.02	0.0736	92.1	35	2013.47
Econ-wml	0.1237	825.6	92	12839.05	0.1144	784.4	103	12307.79	0.1217	813	94	11835.27	0.1303	846	86	12463.4
Bio-celegans	0.029	361.76	142	37763.33	0.02	340.48	363	39410.17	0.0205	332.98	327	40084.07	0.0498	395.48	183	21839.55
Ego-facebook	0.0084	30556.2	1616	2048303	-	-	-	-	-	-	-	-	0.0086	32712.6	1485	2288097

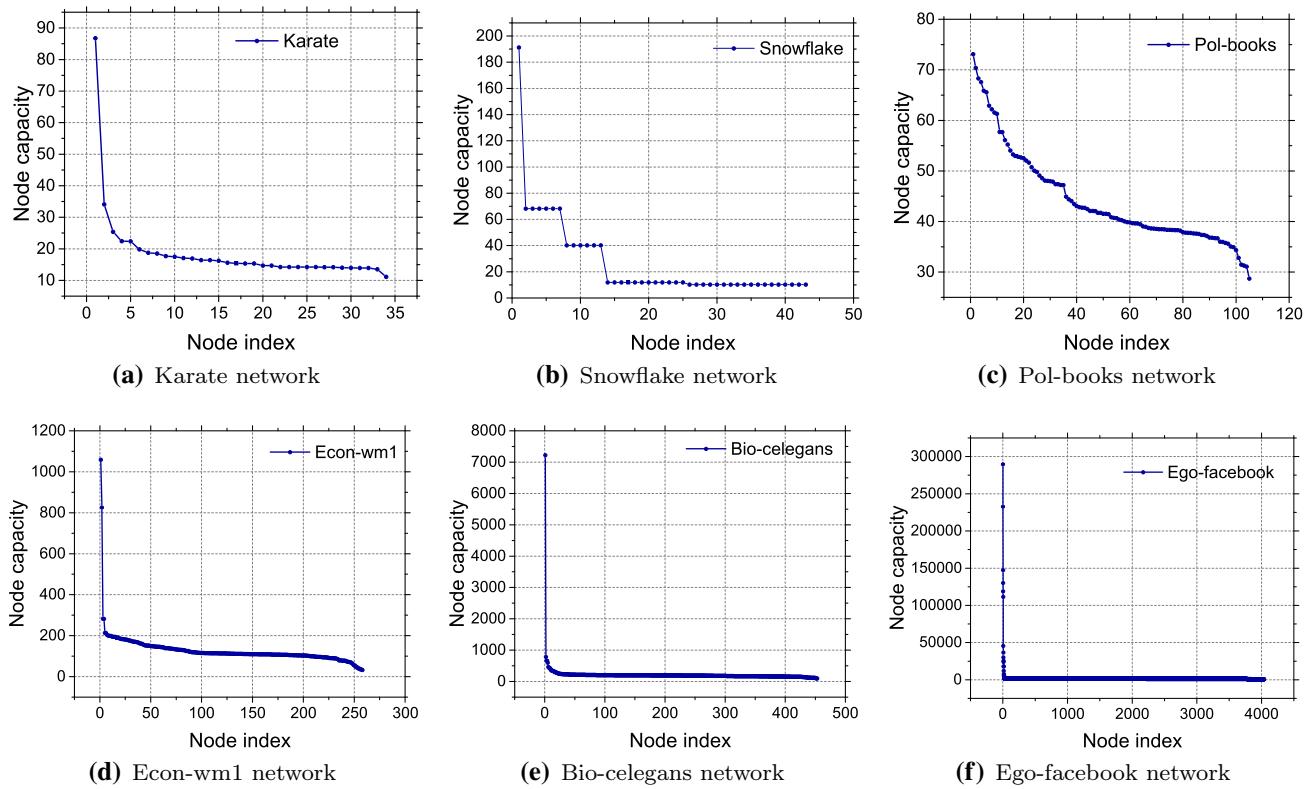
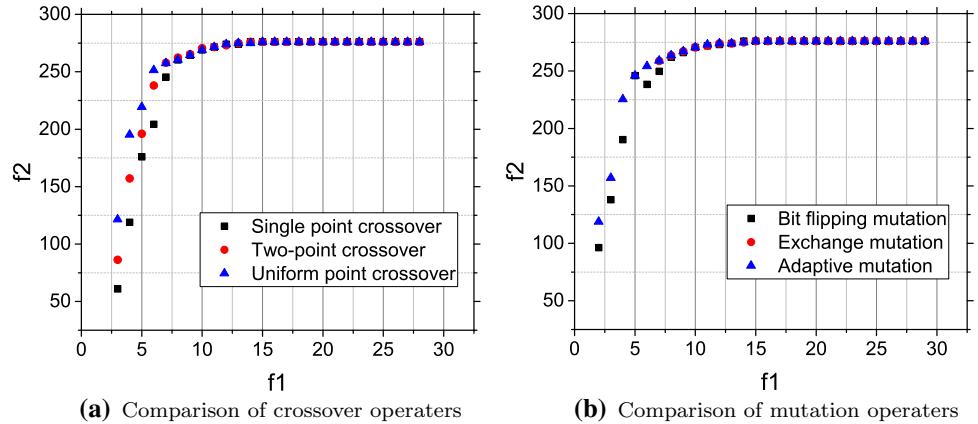


Fig. 15 Heavy-tailed distribution. In real core-periphery networks, arrange the nodes in reverse order of node capacity, and there will be an obvious heavy-tailed distribution

Fig. 16 Comparison of evolutionary operators on karate network. (a) is the comparison of Pareto front among the single point, two-point and uniform point crossover, (b) is the comparison of Pareto front among the bit flipping, exchange and adaptive mutation. The evolutionary parameters both are ($S_p=50$, $E_{max}=5000$, $P_c=0.9$, $P_m=0.1$, $P_l=0.2$)



some deviations in the traditional algorithms, whereas the CPS detected by the MOMA-PCLS algorithm is more accurate, which suggest the better global search capability of the evolutionary algorithm.

5.4.2 Comparison with multi-objective algorithms

Table 6 reports the comparison of other multi-objective algorithms with our MOMA-PCLS algorithm. Each group of tests is run 20 times, and the average values, standard values, and Friedmens Test's P-value are reported. The

parameter setting are ($S_p=50$, $E_{max}=10000$, $P_c=0.9$, $P_m=0.1$, $P_l=0.2$) for Karate, Snowflake, and Pol-books networks, and ($S_p=200$, $E_{max}=20000$, $P_c=0.9$, $P_m=0.1$, $P_l=0.2$) for Econ-wm1, Bio-celegans, and Ego-facebook networks, which are set separately for networks of different sizes. Fewer evaluation times represents faster convergence of the algorithm. MOMA-PCLS with PCLS converges to better solutions than MOGA and MOMA. The meme operator PCLS can accelerate the convergence speed.

Fig. 19 shows the Pareto front of multi-objective algorithms. In Fig. 19, f_1 represents the number of core nodes,

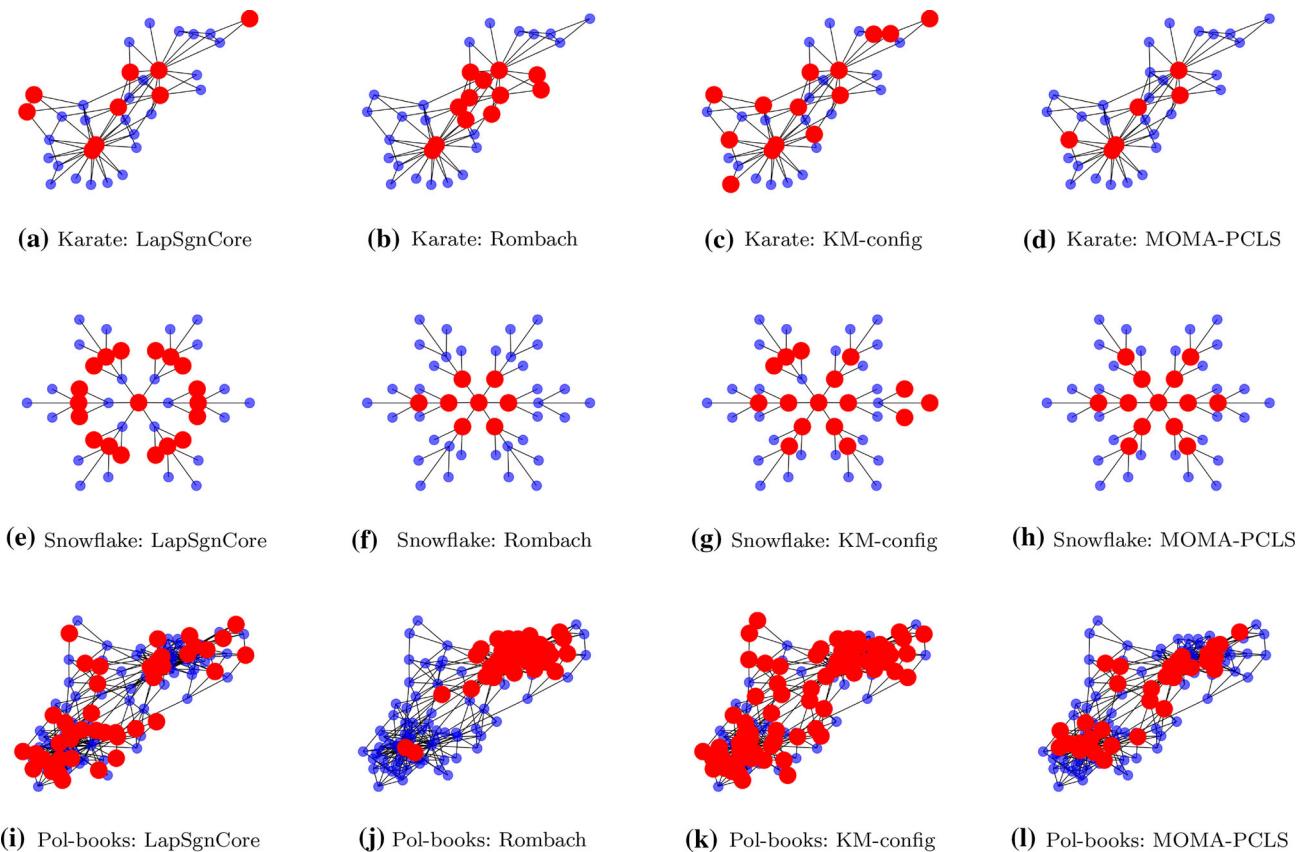


Fig. 18 The visual comparison on small-scale networks. (a) - (d) are the detection results of state-of-the-art algorithms on karate network, (e) - (h) are on snowflake network, and (i) - (l) are on Pol-books network. The

detection results of traditional single objective algorithms exist some deviations, the detection accuracy of the MOMA-PCLS algorithm is higher

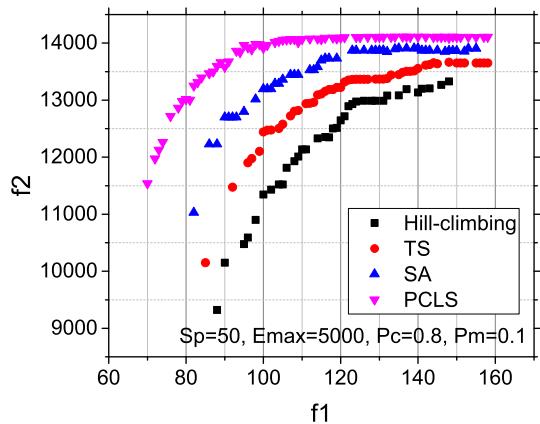


Fig. 17 Comparison of local search methods on econ-wm1 network. The comparison is based on the Pareto fronts of four local search methods: hill-climbing, Tabu search, simulated annealing, and PCLS. The evolutionary parameters are ($S_p=50$, $E_{max}=5000$, $P_c=0.9$, $P_m=0.1$, $P_l=0.2$)

and $f_2/C(G)$ represents the ratio of network capacity occupied by core nodes that need to be maximized. Taking Fig. 19(a) as an example, when selecting 10 core nodes in the

f_1 axis, the $f_2/C(G)$ values for MOGA, MOMA, MOMA-PCLS are 0.35, 0.45, and 0.5, respectively. It can be seen that MOMA-PCLS obtains a better non-dominated set.

The above results illustrate that the MOMA-PCLS algorithm is more accurate and effective than state-of-the-art algorithms in the CPSD. Algorithms based on a single metric do not solve the resolution problem well. In the comparison with the multi-objective algorithms, PCLS can detect an accurate CPS more efficiently. However, we must acknowledge that our code in the current form is impotent to handle the much larger network [21,43] due to the lack of efficient parallelism and the visualization is also difficult to distinguish the core nodes due to the huge network density.

6 Conclusion

The CPS represents the most robust topology of networks against random failures. CPSD can effectively reflect the non-trivial organization of complex networks. This paper formulates CPSD as a multi-objective optimization problem and proposes a multi-objective memetic algorithm called

Table 6 Evaluation statistical result of multi-objective test algorithms over 20 independent trials on six networks. For a detailed comparison, the average values, standard values, P-values are computed based on

	MOGA		MOMA		MOMA-PCLS		P-value
	Avg.	Std.	Avg.	Std.	Avg.	Std.	
Karate	4843.35(+)	±2009.5732	3171.3(+)	±1596.0748	2135.4	±902.6311	2.36E-03
Snowflake	2541.9(+)	±1209.96	1682.1(+)	±1063.4815	1381.55	±721.2354	2.88E-03
Pol-books	8851.5(+)	±1224.1947	8192.5(+)	±1027.5742	7005.92	±906.5456	6.42E-06
Econ-wm1	18575.0(+)	±1147.9874	17100.0(+)	±1049.2855	14930.04	±921.4662	9.21E-08
Bio-celegans	19705.0(+)	±1373.4901	17270.0(+)	±1109.0987	15355.13	±985.6343	9.31E-08
Ego-facebook	19025.0(+)	±1165.2789	18315.0(+)	±1101.9415	17790.0	±966.9023	1.74E-02

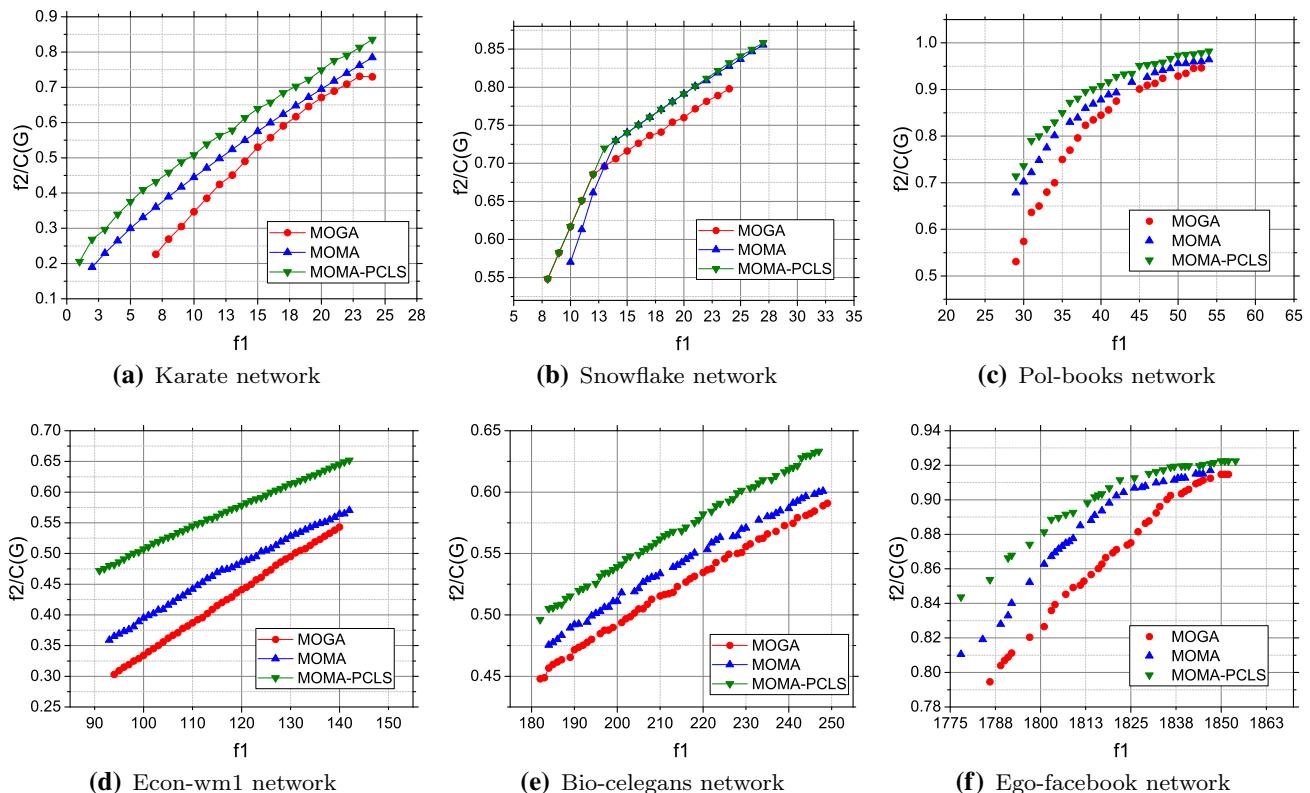


Fig. 19 Comparison of search accuracy of evolutionary algorithms in multiple dimensions. (a) karate Pareto front, (b) snowflake Pareto front, (c) pol-books Pareto front, (d) econ-wm1 Pareto front, (e) bio-celegans

the comparison algorithms. The Friedmen Test's P-value indicates that the results are significantly different if the P-value is less than 0.05

MOMA-PCLS to solve the problem. MOMA-PCLS utilizes the heavy-tailed distribution of node capacity and reduces the core nodes search space. Moreover, according to the feature of combination search space, a PCLS method is designed to speed up the local search efficiency. The experimental results demonstrate that the MOMA-PCLS algorithm is more accurate than other state-of-the-art CPSD algorithms and faster than classical multi-objective evolutionary algorithms. In the future, the MOMA-PCLS algorithm could be extended to a multitasking algorithm for multiple related complex net-

works via knowledge transfer. GPU-accelerated methods can be introduced to solve the performance degradation problem in large-scale networks. The source code of MOMA-PCLS is available via <http://csse.szu.edu.cn/staff/zhuzz/MOMA-PCLS>.

Funding This work was supported in part by National Natural Science Foundation of China [61871272 and 61803269], in part by the Guangdong Provincial Key Laboratory [2020B121201001 and 2020A1515010790], in part by the Shenzhen Fundamental Research Program [JCYJ20190808173617147 and JCYJ20190808174801673],

and in part by the BGI-Research Shenzhen Open Funds [BGIRSZ 20200002].

References

- Amaya JE, Cotta C, Fernández AJ, García-Sánchez P (2020) Deep memetic models for combinatorial optimization problems: application to the tool switching problem. *Memetic Comput* 12(1):3–22
- Borgatti SP, Everett MG (2000) Models of core/periphery structures. *Social Netw* 21(4):375–395
- Chen X, Ong YS, Lim MH, Tan KC (2011) A multi-facet survey on memetic computation. *IEEE Trans Evol Comput* 15(5):591–607
- Cheng F, Cui T, Su Y, Niu Y, Zhang X (2018) A local information based multi-objective evolutionary algorithm for community detection in complex networks. *Appl Soft Comput* 69(1):357–367
- Cucuringu M, Rombach P, Lee SH, Porter MA (2016) Detection of core-periphery structure in networks using spectral methods and geodesic paths. *Eur J Appl Math* 27(6):846–887
- Da Silva MR, Ma H, Zeng AP (2008) Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. *Proc IEEE* 96(8):1411–1420
- Fasino D, Rinaldi F (2020) A fast and exact greedy algorithm for the core-periphery problem. *Symmetry* 12(1):94
- Gabardo AC, Berretta R, Moscato P (2020) M-link: a link clustering memetic algorithm for overlapping community detection. *Memetic Comput* 12(2):87–99
- Glover F (1986) Future paths for integer programming and links to artificial intelligence. *Comput Oper Res* 13(5):533–549
- Gong M, Chen C, Xie Y, Wang S (2020) Community preserving network embedding based on memetic algorithm. *IEEE Trans Emerg Top Comput Intell* 4(2):108–118
- Gu S, Xia CH, Ceric R, Moore TM, Gur RC, Gur RE, Satterthwaite TD, Bassett DS (2020) Unifying the notions of modularity and core-periphery structure in functional brain networks during youth. *Cereb Cortex* 30(3):1087–1102
- Gupta A, Ong YS (2018) Memetic computation: the mainspring of knowledge transfer in a data-driven optimization era, vol 21. Springer, New York, NY, USA
- Ibrahim AO, Shamsuddin SM, Abraham A, Qasem SN (2019) Adaptive memetic method of multi-objective genetic evolutionary algorithm for backpropagation neural network. *Neural Comput Appl* 31(9):4945–4962
- de Jeude JV, Caldarelli G, Squartini C (2019) Detecting core-periphery structures by surprise. *Europhys Lett* 125(6):68001
- Jia J, Benson AR (2019) Random spatial network models for core-periphery structure. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining WSDM, Feb 11–15, 2019, Melbourne, Australia, pp. 366–374
- Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
- Kojaku S, Masuda N (2018) Core-periphery structure requires something else in the network. *New J Phys* 20(4):43012–43012
- Kojaku S, Xu M, Xia H, Masuda N (2019) Multiscale core-periphery structure in a global liner shipping network. *Sci Rep* 9(1):1–15
- Li M, Liu J, Wu P, Teng X (2020) Evolutionary network embedding preserving both local proximity and community structure. *IEEE Trans Evol Comput* 24(3):523–535
- Li W, Qiao M, Qin L, Zhang Y, Chang L, Lin X (2020) Scaling up distance labeling on graphs with core-periphery properties. In: Proceedings of the 2020 International Conference on Management of Data SIGMOD, June 14–19, 2020, Portland, OR, USA, pp. 1367–1381. ACM
- Lierde HV, Chow TWS, Chen G (2020) Scalable spectral clustering for overlapping community detection in large-scale networks. *IEEE Trans Knowl Data Eng* 32(4):754–767
- Ma L, Li J, Lin Q, Gong M, Coello CAC, Ming Z (2019) Cost-aware robust control of signed networks by using a memetic algorithm. *IEEE Trans Cybern* 50(10):4430–4443
- Ma L, Li J, Lin Q, Gong M, Coello CAC, Ming Z (2019) Reliable link inference for network data with community structures. *IEEE Trans Cybern* 49(9):3347–3361
- Ma X, Li X, Zhang Q, Tang K, Liang Z, Xie W, Zhu Z (2019) A survey on cooperative co-evolutionary algorithms. *IEEE Trans Evol Comput* 23(3):421–441
- NetworkX: Configuration model (2021). Accessed 30 March 2021. https://networkx.org/documentation/stable/reference/generated/networkx.generators.degree_seq.configuration_model.html
- Ong YS, Lim MH, Zhu N, Wong KW (2006) Classification of adaptive memetic algorithms: a comparative study. *IEEE Trans Syst Man Cybern B* 36(1):141–152
- Pizzuti C, Socievole A (2019) Multiobjective optimization and local merge for clustering attributed graphs. *IEEE Trans Cybern* 49(1):1–13
- Qiu J, Liu M, Zhang L, Li W, Cheng F (2019) A multi-level knee point based multi-objective evolutionary algorithm for AUC maximization. *Memetic Comput* 11(3):285–296
- Riaza R (2018) Twin subgraphs and core-semiperiphery-periphery structures. *Complexity* 2018(1):1–17
- Rombach P, Porter MA, Fowler JH, Mucha PJ (2017) Core-periphery structure in networks. *SIAM Rev* 59(3):619–646
- Rossi RA, Ahmed NK (2015) The network data repository with interactive graph analytics and visualization. In: Proceedings of the Twenty-Ninth Conference on Artificial Intelligence AAAI, January 25–30, 2015, Austin, Texas, USA, pp. 4292–4293
- Ruiz LGB, Capel MI, Pegalajar M (2019) Parallel memetic algorithm for training recurrent neural networks for the energy efficiency problem. *Appl Soft Comput* 76:356–368
- Sarkar S, Sikdar S, Bhowmick S, Mukherjee A (2018) Using core-periphery structure to predict high centrality nodes in time-varying networks. *Data Min Knowl Disc* 32(5):1368–1396
- SNAP: Ego-facebook network (2021). Accessed 30 March 2021. <http://snap.stanford.edu/data/ego-Facebook.html>
- Tang W, Zhao L, Liu W, Liu Y, Yan B (2019) Recent advance on detecting core-periphery structure: a survey. *CCF Trans Perva Comput Interact* 1(3):175–189
- Tudisco F, Higham DJ (2019) A nonlinear spectral method for core-periphery detection in networks. *SIAM J Math Data Sci* 1(2):269–292
- Wang G, Cai X, Cui Z, Min G, Chen J (2020) High performance computing for cyber physical social systems by using evolutionary multi-objective optimization algorithm. *IEEE Trans Emerg Top Comput* 8(1):20–30
- Wang R, Liu G, Wang C (2019) Identifying protein complexes based on an edge weight algorithm and core-attachment structure. *BMC Bioinform* 20(1):1–20
- Wang S, Gong M, Liu W, Wu Y (2020) Preventing epidemic spreading in networks by community detection and memetic algorithm. *Appl Soft Comput* 89:106118
- Wang S, Liu J (2019) Community robustness and its enhancement in interdependent networks. *Appl Soft Comput* 77:665–677
- Xiang BB, Bao ZK, Ma C, Zhang X, Chen HS, Zhang HF (2018) A unified method of detecting core-periphery structure and community structure in networks. *Chaos: Interdiscipl J Nonlinear Sci* 28(1):013122
- Yang J, Zhang M, Shen KN, Ju X, Guo X (2018) Structural correlation between communities and core-periphery structures in social networks: evidence from twitter data. *Expert Syst Appl* 111(1):91–99

43. Zhang X, Zhou K, Pan H, Zhang L, Zeng X, Jin Y (2020) A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks. *IEEE Trans. Cybern.* 50(2):703–716
44. Zhou Z, Xiao Z, Deng W (2020) Improved community structure discovery algorithm based on combined clique percolation method and k-means algorithm. *Peer Peer Netw. Appl.* 13(6):2224–2233
45. Zhu Z, Ong YS, Dash M (2007) Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recogn.* 40(11):3236–3248
46. Zhu Z, Ong YS, Dash M (2007) Wrapper-filter feature selection algorithm using a memetic framework. *IEEE Trans. Syst. Man Cyber. B* 37(1):70–76

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