Highlights

**Network Collaborator: Knowledge Transfer Between Network Reconstruction and Community Detection**

Kai Wu, Chao Wang, Junyuan Chen, Jing Liu

- At the problem level, we propose joint optimization of NR and CD tasks from dynamics in a multitask setting. This is the first work that focuses on effectively transferring beneficial knowledge of these two tasks to achieve high accuracy.
- We innovatively propose a solution to the multitasking network reconstruction and community detection problems, which existing evolutionary multitasking algorithms cannot solve directly.
Network Collaborator: Knowledge Transfer Between Network Reconstruction and Community Detection

Kai Wu\textsuperscript{a}, Chao Wang\textsuperscript{a}, Junyuan Chen\textsuperscript{b} and Jing Liu\textsuperscript{b,\textast}}

\textsuperscript{a}School of Artificial Intelligence, Xidian University, Xi’an, 710071, China
\textsuperscript{b}Guangzhou Institute of Technology, Xidian University, Guangzhou, 510555, China

\begin{abstract}
Exact network structures and community partitions are valuable tools for modeling and analyzing complex systems. However, deriving them from the dynamics of complex systems is challenging. Although many approaches have been devised to address network reconstruction (NR) and community detection (CD) independently, none of them consider explicit shareable knowledge across these two tasks. NR and CD from dynamics are natural synergistic tasks that motivate the proposed evolutionary multitasking NR and CD framework, called Network Collaborator. In the method, the NR task explicitly shares the network structure with the CD task. And the CD task explicitly transfers the community partitions to assist the NR task. Moreover, to share knowledge from the NR task to the CD task, Network Collaborator models the study of CD from dynamics to find communities in the dynamic network and then considers whether to transfer knowledge across tasks. To verify the performance of Network Collaborator, we design a test suite of multitasking NR and CD problems based on various synthetic and real-world networks. The experimental results have demonstrated that joint NR with CD has a synergistic effect. Specifically, community partitions can be employed to improve the reconstruction accuracy of the network. Conversely, the network structure also better helps the discovery of community partitions. The code can be accessed at https://github.com/xiaofangxd/EMTNRCD.
\end{abstract}

1. Introduction

Complex networks play an essential role in synchronizing and controlling complex dynamical systems Strogatz (2001), which have received extensive attention in many fields. Moreover, uncovering communities in networks complex offers coarse-graining relations between entities. However, in many natural complex systems, the network structure between entities and the complex systems’ nodal dynamics is unobservable. Instead, we may observe interdependent signals from the nodes in a complex network to infer these relationships. Thus, this paper focuses on two tasks: one for reconstructing network structure from dynamics; another one for discovering communities from dynamics.

To solve the first task, a series of network reconstruction (NR) methods has been proposed to reconstruct network structure from dynamics, which are divided into two classes: model-free methods Braunstein, Pagnani, Weigt and Zecchina (2008); Margolin, Nemenman, Bassol, Wiggins, Stolovitzky, Dalla Favera and Califano (2006); Huynh-Thu, Irrthum, Wehenkel and Geurts (2010); Marbach, Costello, Küffner, Vega, Prill, Camacho, Allison, Kellis, Collins and Stolovitzky (2012) and model-based methods Han, Shen, Wang and Di (2015); Wang, Lai, Grebogi and Ye (2011); Nitzan, Casadiego and Timme (2017). In model-free methods, the strength of a link among nodes is obtained by measuring the dependence from their dynamics in terms of correlationsDe La Fuente, Bing, Hoeschele and Mendes (2004), maximum entropy distributions Braunstein et al.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{The differences between our proposal and current methods. Left: CEMO-NR Wu et al. (2021) uses community partitions to aid the network reconstruction task. Middle: current methods first reconstruct the precise network structure from dynamics and then perform community detection on the obtained network. Right: the proposed multitasking network reconstruction and community detection problem.}
\end{figure}

\*\*Corresponding author
\*\*\*neouma@mail.xidian.edu.cn (J. Liu)
\*\*\*\*ORCID(s):
the similarity of any pair of factors in a complex system described above; second, convert the resemblance to a dense weighted network or a binary network Chan, Park, Savalia, Petersen and Wig (2014). After determining the underlying network, the community detection methods are employed to discover clusters of the network, such as modularity Newman (2006b), evolutionary algorithms (EAs) Zhang, Zhou, Pan, Zhang, Zeng and Jin (2018); Folino and Pizzuti (2013); Gong, Cai, Chen and Ma (2013); Liu, Xue, Wu, Zhou, Hu, Paris, Nepal, Yang and Philip (2020b), and deep learning Liu et al. (2020b). These approaches assume that the edges have been observed accurately Fortunato (2010). However, these approaches do not consider knowledge transfer between NR and CD tasks, which may promote the accuracy of the two studies. We also find that a more precise network structure may promote the accuracy of community discovery, and better communities may promote the performance of the NR task. Thus, can the joint optimization of the NR and CD tasks obtain better performance?

This paper proposes an evolutionary multitasking NR and CD framework, termed Network Collaborator (NC), to utilize the information of the network structure and community partition obtained from dynamics to improve each other. The phenomenon inspires this evolutionary multitasking framework that a more precise network structure can improve the accuracy of community discovery, and better communities can promote the performance of the NR task Wu et al. (2021). Fig. 1 shows the difference between our proposal and current methods. NC is inspired by evolutionary multitasking optimization Gupta, Ong and Feng (2016a), a new paradigm for solving multiple tasks by taking advantage of the parallelism mechanism of evolutionary algorithms. Evolutionary multitasking optimization has been successfully applied to overcome many practical challenges in recent years because of its parallelism and easy scalability Gupta, Zhou, Ong, Chen and Hou (2022); Tan, Feng and Jiang (2021); Wang, Liu, Wu and Ying (2021). These studies show that efficient knowledge transfer across tasks can improve the method’s convergence characteristics Feng, Zhou, Zhong, Gupta, Ong, Tan and Qin (2019); Wang, Liu, Wu and Wu (2022). Moreover, NR and CD tasks are often used to analyze complex systems simultaneously, which is a natural multitasking optimization problem. Thus, to share knowledge across the NR and CD tasks, we first establish a multitasking framework, where the community partition obtained from the CD task is explicitly transferred to improve the performance of the NR task, and the better network structures acquired from the NR task are explicitly transferred to enhance the performance of the CD task from dynamics.

However, existing evolutionary multitasking optimization methods cannot be employed to handle this problem because of the specificity of the transferred knowledge. Moreover, the gene coding of the NR task is continuous, and the gene coding of the CD task is discrete. Thus, we need to design a new evolutionary multitasking framework to handle this problem. Current EA-based CD methods assume that the network structure is known. Therefore they cannot directly address the CD task from dynamics. Thus, we design a preprocessing stage to obtain the initial network structure before conducting the CD task. Moreover, communities are discovered from the given network in traditional CD tasks. In our framework, we transfer the new network structure obtained from the NR task to the CD task, which leads to the fact that the CD task differs from discovering communities from the static network. We first model this process as a dynamic CD problem that needs to infer communities from a dynamic network to overcome this issue. To assist the NR task, we design two local search strategies to utilize the inter-community and intra-community structural information transferred from the CD task.

To validate the performance of NC, we design a test suite on four real-world networks and four synthetic networks by considering the evolutionary game (EG) model Nowak and May (1992); Szabó and Fath (2007) and resistor network (RN) model Wang et al. (2011); Wu et al. (2016). Moreover, three state-of-the-art multiobjective EAs for the NR task and a state-of-the-art population-based CD algorithm of the dynamic network are embedded in NC. As shown in the experimental results, joining these two tasks has a synergistic effect, whereby the discovery of communities significantly increases the reconstruction accuracy, which in turn improves the performance of the CD task compared to performing these tasks in isolation.
The contributions of the NC are summarized as follows:

1. The current network reconstruction and community detection tasks are handled separately. We found that the network reconstruction and community detection tasks can facilitate each other. The knowledge that can be transferred between the two tasks is the network structure in the network reconstruction task and the community structure in the community detection task.

2. We design a multitask community detection and network reconstruction framework to exploit the discovered knowledge for transfer between the two tasks, overcoming the shortcomings of current evolutionary multitask optimization frameworks that cannot use this knowledge.

The rest of this paper is organized as follows. Section 2 reviews EA-based NR methods, EA-based CD methods, and evolutionary multiobjective multitasking optimization. In Section 3, we introduce the problem formulation. The details of NC are shown in Section 4. Section 5 gives the experimental results of the designed test suite to illustrate the effectiveness of our methods. Finally, Section 6 summarizes the work in this paper and discusses the potential directions.

2. Related Work

2.1. EA-based NR Methods

The NR task aims to reconstruct the links between each pair of entities. Various EA-based methods were proposed to overcome NR problems. Here, several traditional methods are introduced to show our motivation for NC. These approaches have a similar inference pattern in handling the task of NR from dynamics. The inference model, such as fuzzy cognitive maps Wu et al. (2022b); Wu, Liu, Liu and Yang (2020); Wu and Liu (2017), S-system Kimura, Ide, Kashiwara, Kano, Hatakeyama, Masui, Nakagawa, Yokoyama, Kuramitsu and Konagaya (2005); Palafox et al. (2012), recurrent neural network Xu, Wunsch II and Frank (2007), is used to model the observed data obtained from dynamics. Then the EA is employed to optimize the parameters of the inference model. Finally, the designed inference model is used to obtain the network structure of the complex system. More accurate models and high-performance optimizers are the keys to these approaches. Several approaches are proposed to reconstruct the network structure based on the complex behavior Wu et al. (2016, 2019). However, these methods do not consider the influence of community partition on the NR task, which may improve the NR task’s performance, especially for the high-dimensional NR problems. The most related work (CEMO-NR) proposed in Wu et al. (2021) employed community partition information to aid the NR task by decomposing the original problem into several low-dimensional subproblems. However, CEMO-NR does not consider the CD task nor transfer helpful knowledge to improve the performance of the CD task performed alone. Moreover, both tasks were not optimized simultaneously.

2.2. EA-based CD Methods

A series of EA-based approaches have been proposed to discover the community partition from the different networks Pizzuti (2017), such as large-scale networks Zhang et al. (2018); Gong et al. (2013), dynamic networks Folino and Pizzuti (2013); Zeng, Wang, Chen and Yen (2019); Ma, Wu, Wang and Liu (2023), attributed networks Teng, Liu and Li (2019), multilayer NetworkGao, Yin, Wang, Li and Li (2023); Lyu, Shi, Sun and Lin (2022); Zhao and Cheong (2023), and signed social networks Liu, Liu and Jiang (2014). Moreover, several works are proposed to handle the more difficult task of discovering overlapping communities Wen, Chen, Lin, Gu, Zhang, Li, Yin and Zhang (2016); Zhang, Pan, Su, Zhang and Niu (2017); Teng et al. (2019). However, they are not able to discover communities from dynamics. These works assume that the exact network structure has been obtained before performing the CD. Our proposed NC is the first EA-based framework to overcome this challenge.

In general, most methods need to perform the following two steps to infer communities from dynamics: First, one NR method is chosen to reconstruct network structure from dynamics. Second, the EA-based CD methods are employed to detect network clusters obtained by the first step. This process is described in Fig. 1. This type of approach assumes that the edges are accurately observed. Meanwhile, the error suffered by the CD will enlarge the error of the whole process. Unlike the above methods, Hoffmann et al. Hoffmann, Peel, Lambiotte and Jones (2020) proposed a Bayesian hierarchical model to directly discover communities from time series without considering the process of the NR. However, neither of these approaches attempts to perform the NR together with the CD, and none of them considers knowledge transfer among these two tasks, which may promote the accuracy of the CD from dynamics.

Using the NR and CD tasks characteristics, the proposed NC employs the community partition obtained by the CD task to improve reconstruction accuracy and operates the better network obtained by the NR task to find a better community partition.

2.3. Evolutionary Multiobjective Multitasking Optimization

The NR and CD tasks are modeled as multiobjective multitasking optimization problems in this paper. Therefore, we review the existing methods. Inspired by the multifactorial inheritance across organisms and the parallelism of population-based search, a multiobjective multifactorial evolutionary algorithm (MO-MFEA) was proposed in Gupta, Ong, Feng and Tan (2016b) to optimize multiple tasks simultaneously in a single population. MO-MFEA designed a fixed-parameter rmp to control the knowledge transfer simply, which does not consider the relationship between tasks. Recently, Bali et al. Bali, Gupta, Ong and Tan (2020) introduced MO-MFEA-II to overcome this issue. In this method, an online transfer parameter estimation maintained a transfer parameter matrix RMP to adaptively
control the degree of knowledge. Besides, some methods with adaptive knowledge transfer capabilities have also attracted great attention. An explicit multiobjective multitasking evolutionary algorithm is proposed by Feng et al. (2019), in which each task is assigned an independent solver. Then for every two tasks, a denoising autoencoder is designed to learn linear mappings across tasks. Lin et al. (2019) introduced a novel evolutionary multiobjective multitasking framework based on incremental Naive Bayes classifiers to find helpful knowledge (solutions) during the multitasking search. To improve the convergence rate, Liang et al. (2020) proposed a method to control the degree of knowledge. Besides, some methods based on generative adversarial networks and inertial differential evolution to produce transferable knowledge and knowledge across component layers to improve reconstruction framework for multilayer NR, which exploits the shared components with dense intra-community links and sparse inter-community links from Newman (2003). A network can be regarded as a graph $G = (V, E)$, where the nodes (vertices) in $V$ represent the concepts, and the edges in $E$ represent the relationship among individuals and $E = \{(u, v) \mid u, v \in V, u \neq v\}$ for undirected networks. The network structure $X$ between nodes is defined as follows:

$$X = \begin{bmatrix}
  x_{11} & \cdots & x_{1N} \\
  \vdots & \ddots & \vdots \\
  x_{N1} & \cdots & x_{NN}
\end{bmatrix}$$

where $x_{ij} \in \{0, 1\}$ is the connection across nodes $i$ and $j$ and $N$ represents the number of nodes. Let $Y$ and $h(X, Y)$ be the observed data and the inference model simulation from the candidate network structure, respectively. The goal of the NR is to infer the connections across each pair of nodes according to $Y$ and $h(X, Y)$. In general, the NR problem can be expressed as follows: (Newman (2003))

$$\min_X F(X) = (h(X, Y), g(X))$$

subject to $X \in \{0, 1\}^{N \times N}$

The definition of $h(X, Y)$ and $g(X)$ is determined by the complex network’s inference model. The detailed model can be found in Wu et al. (2022c). We introduce two common NR problems in Appendix.

### 3. Problem Formulation

The main symbols of this paper are summarized in Table 1. This section introduces the problem formulation of multiobjective multitasking optimization, NR problems, and CD problems. Then we discuss how to model NR problems and CD problems as multitasking NR and CD.

#### 3.1. Multiobjective Multitasking Optimization

For $K$ minimization tasks, the multiobjective multitasking optimization problem is mathematically formulated as follows:

$$\begin{aligned}
\min_{\mathbf{x}_i} F_i(\mathbf{x}_i) &= (f_i^1(\mathbf{x}_i), f_i^2(\mathbf{x}_i), \ldots, f_i^{m_i}(\mathbf{x}_i)) \\
\text{s.t.} \mathbf{x}_i &= [x_i^1, x_i^2, \ldots, x_i^{N_i}] \in D_i, i = 1, 2, \ldots, K
\end{aligned}$$

where $F_i(\mathbf{x}_i)$ is the $i$-th multiobjective optimization task and $D_i$ is the search space for optimization task $i$. $n_i$ and $m_i$ are the number of objective functions and the dimensionality of $x_i$ in the $k$-th task, respectively. Suppose that $x_i^{(1)}$ and $x_i^{(2)}$ be two solutions for the $i$-th multiobjective optimization task, $x_i^{(1)}$ is said to Pareto dominate $x_i^{(2)}$, if and only if $f_i^j(x_i^{(1)}) \leq f_i^j(x_i^{(2)}), \forall j \in \{1, 2, \ldots, m_i\}$ and there exists at least one objective $f_i^k(k \in \{1, 2, \ldots, m_i\})$ satisfying $f_i^k(x_i^{(1)}) < f_i^k(x_i^{(2)})$. $x_i^*$ is Pareto optimal if there is no $x_i$ such that $x_i$ dominates $x_i^*$. The set of all Pareto optimal solutions is named the Pareto set. The projection of the Pareto set in the objective space is called the Pareto front. In summary, multiobjective multitasking optimization aims to find a set of solutions approximating the Pareto front for each multiobjective optimization task, which should be as close as possible to the Pareto front and distributed evenly and widely over the Pareto front (Bali et al., 2020).

#### 3.2. Network Reconstruction

A network can be regarded as a graph $G = (V, E)$, where the nodes (vertices) in $V$ represent the concepts, and the edges in $E$ represent the relationship among individuals $\{v_1, v_2, \ldots, v_N\}$ is a set of $N$ nodes and $E = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j\}$ for undirected networks. The network structure $X$ between nodes is defined as follows:

$$X = \begin{bmatrix}
  x_{11} & \cdots & x_{1N} \\
  \vdots & \ddots & \vdots \\
  x_{N1} & \cdots & x_{NN}
\end{bmatrix}$$

where $x_{ij} \in \{0, 1\}$ is the connection across nodes $i$ and $j$ and $N$ represents the number of nodes. Let $Y$ and $h(X, Y)$ be the observed data and the inference model simulation from the candidate network structure, respectively. The goal of the NR is to infer the connections across each pair of nodes according to $Y$ and $h(X, Y)$. In general, the NR problem can be expressed as follows: (Newman (2003))

$$\min_X F(X) = (h(X, Y), g(X))$$

subject to $X \in \{0, 1\}^{N \times N}$

The definition of $h(X, Y)$ and $g(X)$ is determined by the complex network’s inference model. The detailed model can be found in Wu et al. (2022c). We introduce two common NR problems in Appendix.

#### 3.3. Community Detection

It is difficult to handle the task of detecting communities from dynamics. We also introduce the traditional CD from network methods, which will be employed in our framework. CD aims to divide all network nodes into multiple communities with dense intra-community links and sparse inter-community links from Newman (2006b). Let $C = \{C_i \subseteq V, C_i \neq \emptyset, i = 1, 2, \ldots, S\}$ be a set of $S$ communities obtained from $G$. Thus, it must satisfy the following conditions:

$$\begin{aligned}
\bigcup_{i=1}^{S} C_i &= V \\
\forall i \neq j, C_i \neq C_j, i, j \in \{1, 2, \ldots, S\}
\end{aligned}$$
For CD problems, modularity $Q$ is one of the most well-known functions for evaluating the quality of network partitions, which can be expressed as follows:

$$Q = \sum_{s=1}^{S} \left[ \frac{l_s}{e} - \left( \frac{d_s}{2e} \right)^2 \right]$$

where $e$ denotes the total number of edges; $l_s$ and $d_s$ represent the number of edges and the sum of the degree of nodes in the $s$-th community, respectively. The larger the $Q$ value, the better the CD results of the network.

Normalized mutual information (NMI) is a commonly used function to measure the similarity of two community partitions. Let $B_1$ and $B_2$ be two partitions of a network and $A$ be a confusion matrix, where element $A_{ij}$ is the number of nodes that belong to both the $i$-th community of $B_1$ and the $j$-th community of $B_2$. Then NMI can be expressed as follows:

$$\text{NMI}(B_1, B_2) = \frac{-2 \sum_{i=1}^{S_1} \sum_{j=1}^{S_2} A_{ij} \log \left( \frac{A_{ij} \cdot N}{A_i \cdot A_j} \right)}{\sum_{i=1}^{S_1} A_i \log \left( \frac{A_i \cdot N}{N} \right) + \sum_{j=1}^{S_2} A_j \log \left( A_j \cdot N \right)}$$

where $S_1$ and $S_2$ are the number of communities in $B_1$ and $B_2$, respectively; $A_i$ and $A_j$ are the sum of the elements of $A$ in the $i$-th row and $j$-th column, respectively. Generally speaking, the larger the NMI value, the more similar $B_1$ and $B_2$.

### 3.4. Multitasking Network Reconstruction and Community Detection

The NR and CD tasks are two hot topics in researching complex network systems. The former aims to infer the network structure from the limited observed data, while the latter aims to explore the complex interactions and relationships among nodes from dynamics. As is shown in Fig. 1, we can generally reconstruct the network structure from dynamics first and then detect the community partition.

The work proposed in Blondel, Guillaume, Lambiotte and Lefebvre (2008) shows that social networks naturally tend to cluster into groups or communities. The nodes in the same group link more densely than nodes outside the group. Inspired by this, the explicit knowledge transfer across the CD task and NR task is considered to improve each task’s performance, modeled as a multiobjective multitasking optimization problem, as shown in Fig. 1. In the optimization process, the community partition obtained from the CD task can be used to help reconstruct the connection between nodes. Similarly, the network structure obtained from the NR task can also be transferred to assist the CD task. The detailed procedure is shown in Section IV.

### 4. Proposed Method

This section introduces the outline of the proposed method first. Then the explicit knowledge transfer from the NR task to the CD task is presented. Next, we also provide the optimization process of the NR task, including the knowledge transfer from the CD task to the NR task. Finally, the framework of the proposed method is presented.

#### 4.1. Overview

The goal of NC is to simultaneously reconstruct the network structure and detect the community partition from the given limited observed data by transferring knowledge across the NR and CD tasks. The outline of the proposed method is shown in Fig. 2, consisting of two different optimization stages: pre-optimization and normal optimization. In the pre-optimization stage, we first optimize the NR task with a fixed number of function evaluations by employing the general population-based NR algorithm. Then, the initialized network structure is selected from the optimized population. Next, based on the initial network structure, we optimize the CD task with a fixed number of function evaluations by employing the general population-based CD algorithm. After the pre-optimization stage, we obtain the initial network and community partitions used in the subsequent stages. The NR and CD tasks are optimized simultaneously with explicit knowledge transfer in the normal optimization stage. In optimizing each generation of the CD task, the network structures obtained from the NR task in the current generation and the previous generation can be considered the snapshots of a dynamic network at two consecutive time steps, viewed as prior knowledge to assist the CD task. Thus, we model this process as the dynamic CD problem and then solve the CD task using the general population-based CD algorithm of a dynamic network. In optimizing each generation of the NR task, the same population-based NR algorithm as the pre-optimization stage is employed. Since the nodes in the same community link more densely than nodes outside the community, the community partition obtained from the CD task is transferred to help reconstruct the links between nodes in the NR task, which is inspired by the work Wu et al. (2021). Next, we introduce the procedure of explicit knowledge transfer across the NR task and CD task.

#### 4.2. Knowledge Transfer from the NR Task to CD Task

We usually need prior knowledge about the network structure $X$ when solving the CD task. In the evolutionary process of NC, more and more precise network structures are obtained with the development of the optimization of the NR task. Inspired by this, the network structures obtained from the NR task at consecutive generations can be viewed as multiple snapshots of a dynamic network to assist the CD task. To realize the above ideas, the following two key issues need to be resolved: 1) How to obtain the network structure $X(t)$ from optimized population $P_{NR}^{(t)}$ of the NR task in the $t$-th generation? 2) How to take full advantage of the network structure $X(t)$ to assist the CD task?

1) How to obtain the network structure $X(t)$ from optimized population $P_{NR}^{(t)}$ of the NR task in the $t$-th generation?
How to select the network structure \( X^{(t)} \) has a significant effect on the performance of NC. The NR task aims to obtain a set of Pareto-optimal network structures, which represent a trade-off between multiple objectives such as reconstruction performance and network sparsity. Unlike single-objective optimization, the unique best network structure cannot be directly obtained from \( P_{NR}^{(t)} \). In this paper, network structure \( X(t) \) is selected by applying the crowding distance metric to the first nondominated front of the population \( P_{NR}^{(t)} \) reported in Deb, Pratap, Agarwal and Meyarivan (2002). The pseudocode of the selection operator is given in Algorithm 1. Firstly, the fast nondominated sorting method Deb et al. (2002) is employed to partition \( P_{NR}^{(t)} \) into \( L \) nondominated fronts \( F_1, \ldots, F_L \), where \( F_1 \) is the first nondominated front, and \( F_L \) is the last one. The network structures in \( F_1 \) are individuals with good convergence in \( P_{NR}^{(t)} \). Next, the crowding distance of each network structure in \( F_1 \) is calculated. The network structure \( X_{(t)} \) in \( F_1 \) with the largest crowding distance value is selected to assist the CD task. This unique network structure is far away from other network structures in \( F_1 \). If there are two extreme network structures in \( F_1 \), we choose one at random. Therefore, the selection process comprehensively considers the diversity and convergence of the network structure set.

2) How to take full advantage of the network structure \( X^{(t)} \) to assist the CD task?

Algorithm 1 Selection

Require: \( P_{NR}^{(t)} \): Optimized population of the NR task in the \( t \)-th generation.
Ensure: Output \( X^{(t)} \): Network structure obtained from the NR task in the \( t \)-th generation.

1: \( \{ F^1, \ldots, F^L \} \leftarrow Fast \ Nondominated \ Sorting \ (P_{NR}^{(t)}); \)
2: count \( \leftarrow \) Count the number of network structures in \( F^1; \)
3: if count > 2 then
4: \( X^{(t)} \leftarrow \arg \max_{X \in F^1} Crowding \ Distance (F^1); \)
5: else
6: \( X^{(t)} \leftarrow \) Randomly choose a network structure from \( F^1; \)
7: end if

After obtaining the network structure \( X^{(t)} \) from the optimized population \( P_{NR}^{(t)} \), we need to consider how to apply the information of network structure \( X^{(t)} \) to assist the CD task. The evolution analysis of the community partition of dynamic networks is also one of the current research hotspots in network sciences. In the evolutionary process of NC, the network structures \( X^{(t)} \) and \( X^{(t-1)} \) obtained from the NR task for two consecutive generations can be regarded as the snapshots of a dynamic network \( G \) in two consecutive time steps. For convenience, a dynamic network \( G \) is denoted as \( G = \{ G^{(t)} = (V, E^{(t)}), t = 0, 1, \ldots, T \} \), where \( G^{(t)} \)
represents a snapshot of $G$ in the $t$-th generation of NC. Let $C^{(t)} = \{C^{(t)}_j\}_{j=1}^{|V|}$ be the community partition containing $S^{(t)}$ communities obtained from $C^{(t)}$ and $C^{(t)} \cap C^{(t)} = \emptyset$ for any $C^{(t)}_j, C^{(t)}_j \in C$. Then, the CD in dynamic networks aims to find the communities $C = \{C^{(0)}, C^{(1)}, \cdots, C^{(T)}\}$. To uncover dynamic networks’ evolutionary behavior, a framework named temporal smoothness is proposed by Chakrabarti et al. Chakrabarti, Kumar and Tomkins (2006), emphasizing that the network should not shift significantly from one timestep to the next. A cost function $f^{(t)}$ combining cost $f^{(t)}_1$ and temporal cost $f^{(t)}_2$ at generation $t$ is described as follows:

$$f^{(t)}_1(C^{(t)}) , t = 0$$
$$\alpha \cdot f^{(t)}_1(C^{(t)}) + (1 - \alpha) \cdot f^{(t)}_2(C^{(t)}, C^{(t-1)}) , t = 1, T$$

where $\alpha \in [0, 1]$ is a balance parameter that controls the trade-off between $f^{(t)}_1$ and $f^{(t)}_2$. The first term $f^{(t)}_1$ evaluates the quality of the community partition at generation $t$, and the second term $f^{(t)}_2$ evaluates the similarity of community partitions between generation $t$ and $t-1$. To automatically obtain the best trade-off between the above two items, Folino et al. Folino and Pizzuti (2013) transformed the CD in dynamic networks into a multiobjective optimization problem, which can be expressed as follows:

$$\max_{C^{(0)}} f^{(t)}_1(C^{(t)}), t = 0$$
$$\max_{C^{(1)}} (f^{(t)}_1(C^{(t)}) + f^{(t)}_2(C^{(t)}, C^{(t-1)})), t = 1, \cdots, T$$

Based on the reported results in Liu, Wu, Zhou and Yang (2019); Liu, Wu, Xue, Zhou, Yang and Sheng (2020a), modularity $Q$ in Eq. 5 and NMI in Eq. 6 are used as the two optimization objectives for finding communities of dynamic networks in this paper. The former evaluates the quality of community partitions, while the latter evaluates the similarity of community partitions in two consecutive generations. Then we can optimize Eq. 8 by the general population-based CD algorithm of the dynamic network to obtain communities $C = \{C^{(0)}, C^{(1)}, \cdots, C^{(T)}\}$. Based on Eq. 8, the CD task of a dynamic network composed of the network structure obtained by two consecutive generations $t$ and $t-1$ in this paper is shown as follows:

$$\max_{C^{(0)}, C^{(1)}} \left( Q \left(C^{(0)}, N MI \left(C^{(0)}, C^{(0-1)}\right) \right) \right.$$  
$$s.t. C^{(0)} = \left\{\begin{array}{l} C^{(t)} \in V, C^{(t)} \neq \emptyset, j = 1, 2, \cdots, S^{(t)} \\
 \forall C^{(t)}_j \subseteq C^{(t)}, C^{(t)}_j \cap C^{(t)}_j = \emptyset \end{array}\right\}$$

4.3. Knowledge Transfer from the CD Task to NR Task

There may be multiple communities in the real network, and the connections between nodes in one community are generally tight. Inspired by this phenomenon, the community partition $C^{(t)}$ from the CD task in the $t$-th generation is explicitly transferred to the NR task to take advantage of the internal tightness of the community. The knowledge transfer from the CD task to the NR task is proposed to realize the above ideas, as shown in Algorithm 2. Firstly, the solution $X'$ is obtained from the $P^{(t)}_{NR}$ by the same selection method as algorithm 1. Then we perform two local searches $l_{S_1}$ and $l_{S_2}$ on $X'$ by utilizing the link information within and between communities, respectively. Further, the environment selection procedure that is the same as the procedure of the population-based NR algorithm in the normal optimization stage is performed on the union set $P^{(t)}_{NR}, l_{S_1}, l_{S_2}$ to obtain the new population $P^{(t)}_{NR}$ with $N_1$ chromosomes after knowledge transfer. Next, we introduce how to perform two local search operators in detail. Moreover, the example of these two operators can be found in Fig. 3.

A community $C'$ is randomly selected from $C^{(t)}$, and then a local search operator $l_{S_1}$ (lines 3-8 in Algorithm 2) is proposed by utilizing the links information within a community $C'$. Firstly, each link relationship within the community $C'$ in the network $X'$ is changed with a probability of 0.8, while other link relationships remain unchanged. Then an initial population $P_{\alpha}$ with $\alpha$ chromosomes can be obtained by repeating this operator (line 4 in Algorithm 2). In the main loop, $\alpha$ individuals are generated by repeating the following operation. Two solutions from the population $P_{\alpha}$ are selected randomly, and then the single-point crossover

Algorithm 2 Knowledge Transfer from the CD Task to NR Task

Require: $N_1$: Population size for the NR task, $P^{(t)}_{NR}$: Population obtained from the normal optimization stage of the NR task in $t$-th generation,$C^{(t)} = \{C^{(t)}_j, t = 1, 2, \cdots, S^{(t)}\}$: Network structure obtained from the NR task in the $t$-th generation, $a$: Population size for the knowledge transfer from the CD task to NR task, $t_1$: Number of function evaluations for the knowledge transfer from the CD task to NR task.

Ensure: Output $P^{(t)}_{NR}$: Population obtained from the CD task to NR task.

1: $X' \leftarrow$ Selection($P^{(t)}_{NR}$); // Algorithm 1;
2: // local search operator $l_{S_1}$ on $X'$ by utilizing the structural information within a community.
3: $C' \leftarrow$ Randomly choose a community from $C^{(t)}$;
4: $P_{in} \leftarrow$ Initialization within($a, X', C'$);
5: while the used number of function evaluations $\leq t_1$ do
6: $O \leftarrow$ Offspring Generation within($P_{in}$);
7: $P_{env} \leftarrow$ Environment Selection($P_{env}, O, a$);
8: end while
9: // local search operator $l_{S_2}$ on $X'$ by utilizing the structural information between communities.
10: $P_{out} \leftarrow$ Initialization between($a, X', C'$);
11: while the used number of function evaluations $\leq t_1$ do
12: $O \leftarrow$ Offspring Generation between($P_{out}$);
13: $P_{env} \leftarrow$ Environment Selection($P_{env}, O, a$);
14: end while
15: $P^{(t)}_{NR} \leftarrow$ Environment Selection($P^{(t)}_{NR}, P_{in}, P_{out}, N_1$);
and bitwise mutation operator are performed on the decision variables corresponding to the link relationship within the community $C'$ (line 6 in Algorithm 2). Finally, the environment selection procedure that is the same as the population-based NR algorithm procedure in the normal optimization stage is used to select solutions that survive to the next generation (line 7 in Algorithm 2).

Besides, a local search operator $ls_2$ (lines 10-14 in Algorithm 2) is proposed by utilizing the links information between communities, and its difference from $ls_1$ includes two operations: Initialization and offspring generation. We only need to modify the link information within a community $C'$ used in the above two operators of $ls_1$ to the link information between communities to obtain $ls_2$. In summary, as prior knowledge, the link information within and between communities is embedded in two local search operators to optimize the NR task.

### 4.4. Framework of Network Collaborator

The general framework of NC, including two optimization stages, is presented in Algorithm 3. In the pre-optimization step of the NR task, a population $P^{0}_{NR}$ for the NR task is initialized randomly, and the general population-based NR algorithm $Optimizer_{NR}$ can be employed to pre-optimize the NR task with a fixed number of function evaluations ($1 - c \times TFE_1$) (lines 1-2 in Algorithm 3). Then an initial network structure $X^{(0)}$ is selected from $P_{NR}^{(0)}$ by Algorithm 1, and the NR task on $X^{(0)}$ is pre-optimized by the general population-based CD algorithm $Optimizer_{CD_{pre}}$ with a fixed number of function evaluations ($1 - c \times TFE_2$) (line 3-4 in Algorithm 3). After the pre-optimization stage, the initial network structure $X^{(0)}$ and community partition $C^{(0)}$ can be obtained. Next, a normal optimization stage is performed to get more accurate solutions by utilizing the common knowledge across the NR and CD tasks. The same algorithm as $Optimizer_{NR}$ is employed in the main loop to optimize the NR task with a fixed number of function evaluations $N_1$ in the $t$ generation of NC (line 8 in Algorithm 3). Then a general population-based CD algorithm of dynamic network $Optimizer_{CD_{normal}}$ is performed to optimize 9 with a fixed number of function evaluations $t_2$ by utilizing the knowledge (network structure) acquired from the NR task for two consecutive generations (lines 10-11 in Algorithm 3). Next, knowledge transfer from the CD task to the NR task is employed to improve the convergence and diversity of the population $P^{(t)}_{NR}$ (line 12 in Algorithm 3). Finally, the decision-maker has selected the network structure $X^*$ and community partition $C^*$ (lines 17-18 in Algorithm 3).

### 5. Experiments

This section verifies the effectiveness of NC on the proposed test suite. Firstly, we give the experimental settings in Section V.A. Then, the performance of our proposal is tested in Section V.B. Next, we analyze the effect of knowledge transfer of NC in Section V.C. Finally, in Section V.D, the parameter analysis is also given. We perform all experiments on a PC with Windows and Intel Core i7-8700 CPU at 3.20 GHz and 16GB RAM.

#### 5.1. Experimental Setup

**Test Suite** We employ the commonly used networks to construct the test suite, and these networks represent most types of networks. In this section, four real social networks, football Newman (2006a), polbooks, dolphin Newman (2006a), and ZK Zachary (1977) are employed to construct the test suite of MNRCDPs. Meanwhile, synthetic networks, such as Erdős–Rényi random networks (ER) Erdős, Rényi et al.
Wang and Jin (2020), three state-of-the-art multiobjective EAs, are embedded in NC as OptimizerNR to form NC-NSGA-II, NC-SPEA2, and NC-SparseEA, respectively. Then ECD Liu et al. (2020a), a state-of-the-art EA of the CD method in the dynamic network, and PNGACD Liu et al. (2019), a state-of-the-art EA of the CD method that has the same base search operator as ECD, are considered as OptimizerCDnormal and OptimizerCDpre in NC-NSGA-II, NC-SPEA2, and NC-SparseEA respectively. All experiments are implemented on the multiobjective optimization platform PlatEMO Tian, Cheng, Zhang and Jin (2017).

In general, we should compare NC with the alternative of solving the tasks with a parallel method but not sharing any knowledge. To verify the performance of our proposal in the experiments, a comparison algorithm called NR2CD is designed, which follows the general process of the NR task and the CD task.

1) Single-task NR Methods. In NR2CD, we first reconstruct the network structure from the observed data and then detect the community partition where knowledge transfer does not occur across these two tasks. NSGA-II and SPEA2 are embedded in NR2CD as the base optimizer for the NRP to form NR2CD-NSGA-II and NR2CD-SPEA2, respectively. Because the first stage of NR2CD is the same as the single-task NR method, the results of the NR task in NR2CD can be used to finish this comparison. It should be highlighted that this paper aims to examine the potential for mutual promotion between NR and CD tasks, not to attain competitive network reconstruction performance. Therefore, we do not compare with those state-of-the-art approaches.

2) Single-task CD Methods. As shown in Fig. 1, none of the current methods can detect communities from dynamics

### Algorithm 3 Network Collaborator

**Require:** Z: MNRCDDPs; \( N_1 \): Population size for the NR task; \( N_2 \): Population size for the CD task; \( TFE_1 \): Number of function evaluations for the NR task; \( TFE_2 \): Number of function evaluations for the CD task; \( \lambda, a, t_1, t_2 \): Key parameters; OptimizerNR: Optimizer for the NR task; OptimizerCDnormal: Optimizer for the CD task in pre-optimization stage; OptimizerCDpre: Optimizer for the CD task in normal optimization stage.

**Ensure:** Output learned network structures \( X^* \) and community partition \( C^* \).

1. \( p_{NR}^{(0)} \leftarrow \text{Initialization}(N_1, Z); \)
2. \( \text{// pre-optimization stage of the NR task} \)
3. \( p_{NR}^{(0)} \leftarrow \text{OptimizerNR}(N_1, Z, p_{NR}^{(0)}(1-\lambda) \times TFE_1); \)
4. \( X^{(0)} \leftarrow \text{Selection}(p_{NR}^{(0)}); // \text{Algorithm 1} \)
5. \( \text{// pre-optimization stage of the CD task} \)
6. \( C^{(0)} \leftarrow \text{OptimizerCDpre}(N_2, Z, (1-\lambda) \times TFE_2); \)
7. \( t \leftarrow 1; \)
8. while termination criterion not fulfilled do
9. \( \text{if used } FEl \leq \lambda \times TFE_1 \text{ then} \)
10. \( \text{// normal optimization stage of the NR task} \)
11. \( p_{NR}^{(t)} \leftarrow \text{OptimizerNR}(N_1, Z, p_{NR}^{(t)}(1-\lambda) \times TFE_1); \)
12. \( \text{if used } FEl \leq \lambda \times TFE_2 \text{ then} \)
13. \( X^{(t)} \leftarrow \text{Selection}(p_{NR}^{(t)}); // \text{Algorithm 1} \)
14. \( \text{// normal optimization stage of the CD task} \)
15. \( C^{(t)} \leftarrow \text{OptimizerCDnormal}(N_2, Z, X^{(t)}, C^{(t-1)}); \)
16. \( p_{NR}^{(t)} \leftarrow \text{KTransfer}_{CD-NR}(N_1, p_{NR}^{(t)}(1-\lambda) \times TFE_1)(N_2, Z); // \text{Algorithm 2} \)
17. end if
18. end if
19. \( t \leftarrow t + 1; \)
20. end while
21. \( X^* \leftarrow \text{Selection}(p_{NR}^{(t)}); // \text{Algorithm 1} \)
22. \( C^* \leftarrow C^{(t)} \)

### Table 2

Test suite of MNRCDDPs.

| ProblemID | Network | \( N_1 - L \) | Type | \( D \) | \( N \) | \( S_1 \) |
|-----------|---------|--------------|------|------|------|------|
| EG1       | ZK      | 1156         | 34   | 3    |
| EG2       | polbooks| 11025        | 105  | 3    |
| EG3       | football| 13255        | 115  | 12   |
| EG4       | dolphin | 3844         | 62   | 2    |
| EG5       | ZK      | 1156         | 34   | 2    |
| EG6       | polbooks| 11025        | 105  | 3    |
| EG7       | football| 13255        | 115  | 12   |
| EG8       | dolphin | 3844         | 62   | 2    |
| EG9       | BA      | 2500         | 50   | -    |
| EG10      | ER      | 2500         | 50   | -    |
| EG11      | NW      | 2500         | 50   | -    |
| EG12      | WS      | 2500         | 50   | -    |
| RN1       | ZK      | 1156         | 34   | 2    |
| RN2       | polbooks| 11025        | 105  | 3    |
| RN3       | football| 13255        | 115  | 12   |
| RN4       | dolphin | 3844         | 62   | 2    |
| RN5       | ZK      | 1156         | 34   | 2    |
| RN6       | polbooks| 11025        | 105  | 3    |
| RN7       | football| 13255        | 115  | 12   |
| RN8       | dolphin | 3844         | 62   | 2    |
| RN9       | BA      | 2500         | 50   | -    |
| RN10      | ER      | 2500         | 50   | -    |
| RN11      | NW      | 2500         | 50   | -    |
| RN12      | WS      | 2500         | 50   | -    |

(1960), Barabási–Albert scale-free networks (BA) Barabási and Albert (1999), Newman–Watts small-world networks (NW) Newman and Watts (1999), and Watts–Strogatz small-world networks (WS) Watts and Strogatz (1998) with the average degree 6, are employed to enrich the property of the designed benchmark. These synthetic networks may not have a good network division. The details of those networks are presented in Table 2, including the type of variables for the NR task, the number of variables \( D \) for the NR task, the number of nodes \( N \), the number of links \( L \), and the number of communities \( S_1 \). In terms of \( D \), these problems are high-dimensional. In the test suite, two types of response sequences are generated from the given networks by the methods described in the Appendix. The first one is the case of 5 response sequences with ten rounds each (\( N_S=5, L=10 \)). The second is the case of 20 response sequences with ten rounds each (\( N_S=20, L=10 \)). In Table 2, EG1–EG12 and RN1–RN12 represent the EG and RN tasks, respectively. This test suite is widely used in Han et al. (2015); Wang et al. (2011).
directly. Thus, we cannot compare our proposal with the single task of the CD from dynamics.

Different multitasking environments are created with synergistic and non-synergistic tasks to see the efficiency of solving simultaneous tasks. We consider the following two cases.

3) CD Methods Aided by the NR Task. For a fair comparison, PNGACD is considered the base optimizer for the CD task in NR2CD-NSGA-II and NR2CD-SPEA2 and detects communities from the networks obtained by single-task NR methods. The results of the CD task in NR2CD can be used to finish this comparison.

4) NR methods Aided by Communities. CEMO-NR Wu et al. (2021) employed community partition to aid the task of network reconstruction. In NC, the idea of transferring the best communities for the NR task is inspired by CEMO-NR. The performance of NC matches that of CEMO-NR, and then we do not show the NR results of CEMO-NR.

5) Evolutionary Multitasking Methods. None of the current evolutionary multitasking methods can handle multitasking network reconstruction and community detection, especially their knowledge transfer strategies. The proposed NC is a unique design for this problem. Thus, we cannot compare our proposal with the state-of-the-art evolutionary multitasking methods.

Table 3

| ID | Type of Tasks | NR2CD-NSGA-II | NC-NSGA-II | NR2CD-SPEA2 | NC-SPEA2 | NR2CD-SparseEA | NC-SparseEA |
|----|---------------|---------------|------------|-------------|-----------|----------------|-------------|
| EG1| CD Task       | 7.90e-01 (2.31e-02) | 7.90e-01 (2.31e-02) | 8.37e-01 (2.33e-02) | 8.48e-01 (2.36e-02) | 7.88e-01 (2.39e-02) | 7.88e-01 (2.35e-02) |
| EG2| CD Task       | 3.03e-01 (9.33e-03) | 3.03e-01 (9.33e-03) | 3.52e-01 (9.52e-03) | 3.27e-01 (9.70e-03) | 8.75e-02 (1.51e-02) | 3.52e-01 (3.45e-04) |
| EG3| CD Task       | 5.14e-01 (5.08e-03) | 5.05e-01 (5.05e-03) | 5.05e-01 (5.05e-03) | 5.05e-01 (5.05e-03) | 5.05e-01 (5.05e-03) | 5.05e-01 (5.05e-03) |
| EG4| CD Task       | 7.77e-01 (8.06e-03) | 7.37e-01 (8.16e-03) | 7.37e-01 (8.16e-03) | 7.37e-01 (8.16e-03) | 7.37e-01 (8.16e-03) | 7.37e-01 (8.16e-03) |
| EG5| CD Task       | 8.55e-01 (3.26e-02) | 8.44e-01 (3.27e-02) | 8.44e-01 (3.27e-02) | 8.44e-01 (3.27e-02) | 8.44e-01 (3.27e-02) | 8.44e-01 (3.27e-02) |
| EG6| CD Task       | 7.57e-01 (1.38e-01) | 8.57e-01 (1.38e-01) | 8.57e-01 (1.38e-01) | 8.57e-01 (1.38e-01) | 8.57e-01 (1.38e-01) | 8.57e-01 (1.38e-01) |
| EG7| CD Task       | 5.16e-01 (7.06e-03) | 5.10e-01 (7.06e-03) | 5.10e-01 (7.06e-03) | 5.10e-01 (7.06e-03) | 5.10e-01 (7.06e-03) | 5.10e-01 (7.06e-03) |
| EG8| CD Task       | 7.19e-01 (5.39e-03) | 7.10e-01 (5.39e-03) | 7.10e-01 (5.39e-03) | 7.10e-01 (5.39e-03) | 7.10e-01 (5.39e-03) | 7.10e-01 (5.39e-03) |
| RN1| CD Task       | 5.73e-01 (9.07e-03) | 9.33e-01 (9.52e-03) | 9.33e-01 (9.52e-03) | 8.92e-01 (9.71e-03) | 8.92e-01 (9.71e-03) | 8.92e-01 (9.71e-03) |
| RN2| CD Task       | 5.06e-01 (7.90e-03) | 5.03e-01 (7.90e-03) | 5.03e-01 (7.90e-03) | 5.01e-01 (7.90e-03) | 5.01e-01 (7.90e-03) | 5.01e-01 (7.90e-03) |
| RN3| CD Task       | 3.63e-01 (9.97e-03) | 3.63e-01 (9.97e-03) | 3.63e-01 (9.97e-03) | 3.63e-01 (9.97e-03) | 3.63e-01 (9.97e-03) | 3.63e-01 (9.97e-03) |
| RN4| CD Task       | 6.51e-01 (9.06e-03) | 6.51e-01 (9.06e-03) | 6.51e-01 (9.06e-03) | 6.51e-01 (9.06e-03) | 6.51e-01 (9.06e-03) | 6.51e-01 (9.06e-03) |
| RN5| CD Task       | 4.80e-01 (3.85e-02) | 7.00e-01 (3.85e-02) | 7.00e-01 (3.85e-02) | 7.00e-01 (3.85e-02) | 7.00e-01 (3.85e-02) | 7.00e-01 (3.85e-02) |
| RN6| CD Task       | 3.91e-01 (2.58e-02) | 7.66e-01 (2.58e-02) | 7.66e-01 (2.58e-02) | 7.66e-01 (2.58e-02) | 7.66e-01 (2.58e-02) | 7.66e-01 (2.58e-02) |
| RN7| CD Task       | 1.78e-01 (3.06e-01) | 6.56e-01 (3.06e-01) | 6.56e-01 (3.06e-01) | 6.56e-01 (3.06e-01) | 6.56e-01 (3.06e-01) | 6.56e-01 (3.06e-01) |
| RN8| CD Task       | 6.94e-01 (4.90e-03) | 6.94e-01 (4.90e-03) | 6.94e-01 (4.90e-03) | 6.94e-01 (4.90e-03) | 6.94e-01 (4.90e-03) | 6.94e-01 (4.90e-03) |

Evaluation Metrics

Two measurement indices are employed to evaluate the performance of our proposal, the Matthews correlation coefficient (MCC) and NMI. MCC is employed to measure the accuracy of the reconstructed network structure, and the NMI is employed to measure the quality of community partitions. For each case, all comparison algorithms are run 20 times independently to obtain statistical results, and the Wilcoxon rank-sum test is employed to test the significance of the experimental results.

Parameter Settings

NSGA-II, SPEA2, and SparseEA used the same single-point crossover and bit-wise mutation operators for the NR task. For the CD task, ECD and PNGACD use the same crossover, mutation, and migration operators in Liu et al. (2020a). The population size for the NR task and CD task is set to 100, and both the maximum number of function evaluations for the NR and CD tasks are assigned to 200000 in all experiments. In NC, λ and t1 are set to 0.5 and 1000, respectively. Then the number of time steps for the CD task in NC, marked as T, is \[ T = \frac{\lambda \times T F E}{(N_t + 2 \times t_1)} \] and the number of function evaluations for the CD task in each time step in NC, marked as t2, is \( T F E / T \). The Sub-population size \( \alpha \) for the knowledge transfer from the CD task to the NR task in NC is set to 20. Details of parameter settings are listed in Table 4.
5.2. Results and Discussion

Table 3 lists the mean and standard deviation of MCC (NR Task) and NMI (CD Task) obtained by different methods on problems EG1-EG8 and RN1-RN8 over 20 independent runs, where symbols “−”, “≈” and “+” imply that the NR2CD-Alg is significantly worse, similar and better than NC-Alg on the Wilcoxon rank-sum test with 95% confidence level, respectively. We mark the best results in boldface.

In Table 3, NC-Alg has shown excellent performance in terms of MCC and NMI in the test suite. Specifically, compared to the single NR task, NC-Alg performs better on 42, ties 6, and loses once out of 48 tasks in terms of MCC (NR Task), which demonstrates that the transferred community information in our method may be helpful. NC-Alg exceeds the NR2CD-Alg in 46 and loses 2 out of 48 cases in terms of NMI, which shows our proposal is better than the CD methods aided by the NR task. In NC-Alg, useful knowledge is transferred across the NR and CD tasks to improve each task’s performance. Since NC-Alg has the same basic evolutionary solver of the NR task and CD task as NR2CD-Alg, knowledge transfer’s effectiveness in NC-Alg is confirmed. For the problems with the polbooks network, NC-Alg and NR2CD-Alg have similar performances in terms of the NR task. Since the community partition in the reconstructed imprecise network may have a large gap from the community partition in the existing network, knowledge transfer may fail in large-scale networks. Moreover, it is found that the number of communities is much larger than the real number of communities. The size of transferred communities is too small to offset its consumption of computational resources. Table 3 shows that the quality of both networks and communities, except for ZK, is not impressive. NMI is often below 0.5, which means that the approach cannot recover the true communities, but the MCC is not high. It can be found that the accuracy of the obtained network structure is not high, and the community detected on this network may deviate from the real community partition. This phenomenon appears since we do not design the specified operator for these two tasks, which may improve the performance violently. These experiments aim to verify the effectiveness of the designed knowledge transfer operator and whether NC can handle the task of CD and NR from dynamics jointly. We think the experiments have achieved these goals.

In the worst case, one network contains a community. In this case, NC can be considered as the single task of reconstructing the network alone. Moreover, it is challenging to create non-synergistic tasks since the more accurate the given network is, the more accurate the detection results are. Also, due to the limited ability of evolutionary algorithms, the performance of the NR decreases with bigger networks, as shown in Table 3. The core of the experiments is to verify whether the proposed knowledge transfer strategy is effective and whether NC can handle multitasking network reconstruction and community detection.

To better illustrate the performance of NC-Alg, we visualize the network structure and community partition obtained by NC-NSGA-II on the ZK network. In Fig. 4, we mark the different communities with different colors. According to the connection between every two nodes obtained by knowledge transfer from the NR task to the CD task, NC-NSGA-II divides all nodes into two communities, which is consistent with the actual community division as shown in Table 2. Our proposal organizes the network into two divisions according to the connection density between nodes, where closely connected nodes are divided into the same community.

5.3. Verification of Knowledge Transfer

This section further illustrates the effectiveness of the knowledge transfer operators from the CD task to the NR task and from the NR task to the CD task in NC and takes the algorithm NC-NSGA-II and the problem EG1 as an example.

Knowledge transfer from the CD task to the NR task is valuable. Fig. 5 shows the obtained non-dominated solution set before and after the knowledge transfer from the CD task to the NR task when the number of function evaluations for the NRP is 102100 and 198700. As can be observed in Fig. 5a, in the mid-stage of evolution, compared with the
the number of function evaluations for the NRP is 102100. (b) The number of function evaluations for the NRP is 198700.

Figure 5: The obtained non-dominated solution set before (grey) and after (red) the knowledge transfer from the CD task to the NR task. (a) The number of function evaluations after the knowledge transfer from the CD task to the NR task positively impacts the performance shown in the early stage of evolution, but the computational budget at the last stage, but the knowledge transfer still uses the computational budget. We can decrease this negative knowledge transfer by reducing the frequency of knowledge transfer.

Knowledge transfer from the NR task to the CD task is valuable. Besides, Fig. 6 shows a curve of values of NMI over generations, in which the NMI values are calculated based on community partitions at each generation and real community partitions.

non-dominated solution set (grey) before knowledge transfer from the CD task to the NR task, the new non-dominated solution set (red) has better overall performance in terms of convergence and diversity, which illustrates the effectiveness of knowledge transfer from the CD task to NR task. As shown in Fig. 5b, in the later stages of evolution, the overall performance of the non-dominated solution set before and after the knowledge transfer is not much different in terms of convergence and diversity. Since the network structure obtained by the NR task has fully utilized the community partition obtained by the CD task, the knowledge transfer process from the CD task to the NR task in the later stage of evolution is not apparent for improving the non-dominated solution set.

Figure 6: A curve of the NMI values over generations, in which the NMI values are calculated based on community partitions at each generation and the real community partitions.

Table 5
Experimental results of NC-NSGA-II and NR2CD-NSGA-II on EG9-EG12 and RN9-RN12 in terms of MCC (NR task) and Q (CD task).

| ProblemID | Task (Metric) | NR2CD-NSGA-II | NC-NSGA-II |
|-----------|---------------|---------------|------------|
| EG9       | NR(MCC)       | 6.22e-01(3.05e-01) | 7.92e-01(2.36e-02) |
|           | CD(Q)         | 3.18e-01(1.71e-01) | 3.89e-01(0.00e+00) |
| EG10      | NR(MCC)       | 6.90e-01(3.41e-01) | 9.46e-01(2.50e-02) |
|           | CD(Q)         | 5.75e-01(2.66e-01) | 7.05e-01(3.82e-02) |
| EG11      | NR(MCC)       | 8.47e-01(1.36e-02) | 9.73e-01(1.48e-02) |
|           | CD(Q)         | 7.16e-01(4.39e-02) | 7.60e-01(5.07e-02) |
| EG12      | NR(MCC)       | 7.11e-01(3.48e-01) | 9.58e-01(1.57e-02) |
|           | CD(Q)         | 6.66e-01(3.36e-01) | 7.95e-01(2.57e-02) |
| RN9       | NR(MCC)       | 6.70e-01(3.28e-01) | 9.30e-01(7.99e-03) |
|           | CD(Q)         | 3.74e-01(1.89e-01) | 5.14e-01(5.94e-02) |
| RN10      | NR(MCC)       | 7.27e-01(3.57e-01) | 9.29e-01(2.50e-02) |
|           | CD(Q)         | 6.44e-01(3.17e-01) | 8.22e-01(5.44e-02) |
| RN11      | NR(MCC)       | 8.66e-01(2.51e-02) | 9.45e-01(1.57e-02) |
|           | CD(Q)         | 6.76e-01(9.35e-02) | 7.53e-01(3.39e-02) |
| RN12      | NR(MCC)       | 7.23e-01(3.54e-01) | 9.43e-01(2.61e-02) |
|           | CD(Q)         | 6.80e-01(3.45e-01) | 7.58e-01(3.42e-02) |

5.4. Testing on Synthetic Networks

We test the performance of the proposed algorithm under different characteristic networks, and the datasets used are shown in Table 2 as EG9-EG12 and RN9-EG12. Since these problems do not have precise community partition, this section uses Q as the performance metric for the CD task. See Table 4 for other settings. In Table 5, it can be found that on eight problems, NC-NSGA-II performs better than NR2CD-NSGA-II. It can be shown that when the community partition is not apparent, our scheme is still effective and can also be adapted to a wide range of scenarios.

5.5. Parameter Sensitivity

The effects of two critical parameters are analyzed in this section, and we take the algorithm NC-NSGA-II and the problem EG1 as an example. Two key parameters are shown as follows: 1) \( \lambda \), the share of the TFE used for the normal optimization stage in NC; 2) \( t_f \), the number of function evaluations for the knowledge transfer from the CD task to the NR task. To analyze one parameter visually, values of other parameters are fixed.

Effect of \( \lambda \) on NC. Fig. 7 shows the MCC and NMI versus varying \( \lambda \). The value of \( \lambda \) is set to 0.1, 0.3, 0.5, and 0.7. In Fig. 7a, the value of MCC increases with \( \lambda \) until \( \lambda > 0.3 \). Then with increasing \( \lambda \), the median value of MCC decreases. Fig. 7b shows that when \( \lambda \) is 0.5, the value of NMI is the most stable in all cases. With the increase of \( \lambda \), the NMI and MCC first increase and then decrease. This case appears...
obtained by the CD task to aid the NR task and share the better network structure obtained by the NR task to aid the CD task. The experimental results on 24 cases show that this strategy is effective and supports the claim that joining these two tasks has a synergistic effect. The discovery of communities significantly improves the reconstruction accuracy, which finds a better community partition to perform these tasks in isolation. However, in terms of large-scale complex systems, the NR task’s low accuracy will decrease the CD task’s performance. In the future, more robust NR methods should be further studied. Besides, applying our proposal to find overlapping communities is a promising research topic in future work.

6. Conclusions

This paper answers the question: could the joint optimization of NR and CD tasks effectively improve these two tasks’ performance? The answer is Yes, which is ensured by the proposed Network Collaborator. The core is to determine what knowledge should be transferred across two tasks. Benefiting from the proposed evolutionary multitasking framework, we explicitly transfer the better community partition obtained by the CD task to aid the NR task and share the better network structure obtained by the NR task to aid the CD task. The experimental results on 24 cases show that this strategy is effective and supports the claim that joining these two tasks has a synergistic effect. The discovery of communities significantly improves the reconstruction accuracy, which finds a better community partition to perform these tasks in isolation. However, in terms of large-scale complex systems, the NR task’s low accuracy will decrease the CD task’s performance. In the future, more robust NR methods should be further studied. Besides, applying our proposal to find overlapping communities is a promising research topic in future work.

Figure 7: The MCC and NMI versus varying $\lambda$. (a) MCC. (b) NMI.

Figure 8: The MCC and NMI versus varying $t_1$. (a) MCC. (b) NMI.

because the accuracy of initializing the network impacts knowledge transfer in the normal optimization stage. With the increase of $\lambda$, the $TFE$ used for the normal optimization stage decreases, which results in the inability to fully utilize the knowledge between the NR task and the CD task.

Effect of $t_1$ on NC. Fig. 8 shows the MCC and NMI versus varying $t_1$. The value of $t_1$ is set to 200, 400, 600, 800, and 1000. In Fig. 8a, with increasing $t_1$, NC obtains the greater value of MCC. In Fig. 8b, when $t_1$ is 1000, the median value of NMI is 1; that is, the community partition in the EG1 problem is completely identified. With increasing $t_1$, the transferred knowledge (network structure) can be more fully utilized to assist the NR task.

It can be seen from the observation that these parameters significantly impact the performance of NC. These parameters affect the frequency of knowledge transfer and the convergence and diversity of the obtained non-dominated solution. Since different NR and CD problems have unique properties, the collection of parameters in the problem EG1 may not suit the others.

CRediT authorship contribution statement

Kai Wu: Conceptualization, Supervision, Writing - original draft. Chao Wang: Methodology, Software, Validation, Writing - original draft. Junyuan Chen: Formal analysis, Writing - review & editing. Jing Liu: Writing - review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62206205, in part by the Guangdong High-level Innovation Research Institution Project under Grant 2021B0909050008, and in part by the Guangzhou Key Research and Development Program under Grant 202206030003.

Appendix

EG Problem. The evolutionary game (EG) model has commonly been used to model node-to-node interactions in complex systems. In each round of the EG model, two players should choose a cooperation strategy or defection. Then the payoffs of these players are determined by the game’s strategy and payoff matrix. The prisoner’s dilemma games Szabó and Fath (2007) are used in this paper. Its payoff matrix is denoted as follows:

$$P = \begin{pmatrix} 1 & 0 \\ 1.2 & 0 \end{pmatrix}$$

(10)

If both choose different strategies, the defector gets a reward of 1.2, and the cooperator receives a prize of 0. And if both decide to cooperate (or defect), all players get rewards
1 (or 0). Formally, for player $i$ in round $t$, its payoff is as follows:
\[
Y_i(t) = \sum_{j=1}^{N} x_{ij} S_i^T(t) P S_j(t)
\]  
(11)
where $S_i(t)$ is the strategy of player $i$ in the $t$-th round and $T$ represents “transpose”, $x_{ij} = 1$ if players $i$ and $j$ are connected and $x_{ij} = 0$ otherwise. To maximize their payoff at the next round, players adjust strategies according to their payoff and their neighbors after each round of the game. Fermi rule Nowak and May (1992) is adopted to update the payoff data of all players, and the second goal is to ensure the sparsity of the learned EG network. The simulation of the EG problem can be expressed as:
\[
W(S_i \leftarrow S_j) = \frac{1}{1 + \exp\left(\frac{(Y_i - Y_j)}{\kappa}\right)}
\]  
(12)
where $\kappa = 0.1$. To find the connections among players, the EG problem can be reformulated as:
\[
\min_{X,Y} F = \left( h(X, Y) = \sum_{i=1}^{N} \left\| U_i X_i - Y_i \right\|_2^2, g(X) = \| X \|_0 \right)
\]
\[
\text{s.t.}
\begin{align*}
X &= [x_{11}, x_{21}, \ldots, x_{1N}]^T \in \{0, 1\}^N \\
Y &= [Y(1), Y(2), \ldots, Y(L)]^T \\
U_i &= \begin{pmatrix}
S_i^T(1) P S_i(1) & \cdots & S_i^T(1) P S_N(1) \\
\vdots & \ddots & \vdots \\
S_i^T(L) P S_i(L) & \cdots & S_i^T(L) P S_N(L)
\end{pmatrix}
\end{align*}
\]  
(13)
where $L$ is the number of rounds. The first goal is to minimize the difference between the real payoff data and the generated payoff data of all players, and the second goal is to ensure the sparsity of the learned EG network. The simulation of the EG is described as follows Wang et al. (2011):

1. Input an RN network;
2. Each node state is obtained from a random number $\Delta w_i \in [0, 20]$;
3. Calculate the voltage of node $i$ by $V_i = V^* \sin((w + \Delta w_i)t)$;
4. Calculate the electrical current of the node $i$ by (14);
5. Repeat Step 3) to Step 4) $T$ times.

For this dynamic process, the strategies and the payoffs of all players in different rounds are recorded as observational data. For this dynamic process, the strategies and the payoffs of all players in different rounds are recorded as observational data.

**RN Problem** Han et al. (2015). Resistor network dynamics is a standard circuit system considering current transportation in the resistor. The resistance of a resistor across nodes $i$ and $j$ is denoted as $r_{ij}$. For simplicity, $r_{ij} = 1$ if $i$ and $j$ are directly connected by a resistor and $r_{ij} = \infty$ otherwise. According to Kirchhoff’s laws at different periods, assuming the voltages at the nodes and resistances of connections are known, the currents at the nodes are as follows:
\[
I_i(t) = \sum_{j=1}^{N} \frac{1}{r_{ij}} \left( V_i(t) - V_j(t) \right)
\]  
(14)
where $I_i$ is the total current at node $i$ and $V_i = V \sin((w + \Delta w_i)t)$ is the voltage. In this paper, $V^* = 1$ is the voltage peak, $w = 10^3$ is the frequency, and $\Delta w_i \in [0, 20]$ is the perturbation. Assume that only the voltages and currents at the nodes are measurable, and the resistor network can be reconstructed as follows:
\[
\min_{X} F = \left( h(X, Y) = \sum_{i=1}^{N} \left\| R_i X_i - Y_i \right\|_2^2, g(X) = \| X \|_0 \right)
\]  
\[
\text{s.t.}
\begin{align*}
X_i &= [x_{11}, x_{21}, \ldots, x_{1N}]^T \in \{0, 1\}^N \\
Y_i &= [I(1), I(2), \ldots, I(L)]^T \\
R_i &= \begin{pmatrix}
V_i(1) - V_1(1) & \cdots & V_i(1) - V_N(1) \\
\vdots & \ddots & \vdots \\
V_i(L) - V_1(L) & \cdots & V_i(L) - V_N(L)
\end{pmatrix}
\end{align*}
\]  
(15)
where $L$ is the rounds of the observation data. The first goal is to minimize the difference between the real current data and the generated current data, and the second goal is to ensure the sparsity of the RN. The simulation of the RN is described as follows:

1. Input an RN network;
2. Each node state is obtained from a random number $\Delta w_i \in [0, 20]$;
3. Calculate the voltage of node $i$ by $V_i = V^* \sin((w + \Delta w_i)t)$;
4. Calculate the electrical current of the node $i$ by (14);
5. Repeat Step 3) to Step 4) $T$ times.

The voltages and the currents at the nodes simultaneously are recorded as observational data for this dynamic process.

**References**
Bali, K.K., Gupta, A., Ong, Y.S., Tan, P.S., 2020. Cognizant multitasking in multiobjective multifactorial evolution: Mo-mfea-ii. IEEE Transactions on Cybernetics 51, 1784–1796.
Barabási, A.L., Albert, R., 1999. Emergence of scaling in random networks. Science 286, 509–512.
Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008, P10008.
Braunstein, A., Pagnini, A., Weigt, M., Zecchina, R., 2008. Inference algorithms for gene networks: a statistical mechanics analysis. Journal of Statistical Mechanics: Theory and Experiment 2008, P12001.
Casadiego, J., Nitzan, M., Hallerberg, S., Timme, M., 2017. Model-free inference of direct network interactions from nonlinear collective dynamics. Nature Communications 8, 1–10.
Chakrabarti, D., Kumar, R., Tomkins, A., 2006. Evolutionary clustering, in: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 554–560.
Chan, M.Y., Park, D.C., Savalia, N.K., Petersen, S.E., Wig, G.S., 2014. Decreased segregation of brain systems across the healthy adult lifespan. Proceedings of the National Academy of Sciences 111, E4997–E5006.
De La Fuente, A., Bing, N., Hoeschele, I., Mendes, P., 2004. Discovery of meaningful associations in genomic data using partial correlation coefficients. Bioinformatics 20, 3565–3574.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nega-II. IEEE Transactions on Evolutionary Computation 6, 182–197.

Erdős, P., Rényi, A., et al., 1960. On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci 5, 17–60.

Feng, L., Zhou, L., Zhong, J., Gupta, A., Ong, Y.S., Tan, K.C., Qin, A.K., 2019. Evolutionary multitasking via explicit autoencoding. IEEE Transactions on Cybernetics 49, 3457–3470.

Folino, F., Pizzuti, C., 2013. An evolutionary multiobjective approach for community discovery in dynamic networks. IEEE Transactions on Knowledge and Data Engineering 26, 1838–1852.

Fortunato, S., 2010. Community detection in graphs. Physics Reports 486, 75–174.

Gao, C., Yin, Z., Wang, Z., Li, X., Li, X., 2023. Multilayer network inference of s-system models of genetic networks using a cooperative coevolutionary algorithm. Bioinformatics 21, 1154–1163.

Gong, M., Cai, Q., Chen, X., Ma, L., 2013. Complex network clustering by eigenvectors of matrices. Physical review E 74, 036104.

Huynh-Thu, V.A., Irrthum, A., Wehenkel, L., Geurts, P., 2010. Inferring regulatory networks from expression data using a tree-based methods. PloS One 5, e12776.

Han, X., Shen, Z., Wang, W.X., Di, Z., 2015. Robust reconstruction of complex networks from sparse data. Physical Review Letters 114, 028701.

Hoffmann, T., Peel, L., Lambiotte, R., Jones, N.S., 2020. Community detection in networks without observing edges. Science Advances 6, eaav1478.

Huynh-Thu, V.A., Iridum, A., Wehenkel, L., Geurts, P., 2010. Inferring regulatory networks from expression data using tree-based methods. PloS One 5, e12776.

Kimura, S., Ide, K., Kashiwara, A., Kano, M., Hatakeyama, M., Masui, R., Nakagawa, N., Yokoyama, S., Kuramitsu, S., Konagaya, A., 2005. Inference of s-system models of genetic networks using a cooperative coevolutionary algorithm. Bioinformatics 21, 1154–1163.

Liang, Z., Zhu, Y., Wang, X., Li, Z., Zhu, Z., 2022. Evolutionary multitasking for multi-objective optimization based on generative strategies. IEEE Transactions on Evolutionary Computation 1, 1–1doi:10.1109/TEVC.2022.3184988.

Lin, J., Liu, H.L., Xue, B., Zhang, M., Gu, F., 2019. Multiobjective multi-tasking optimization based on incremental learning. IEEE Transactions on Evolutionary Computation 24, 824–838.

Liu, C., Liu, J., Jiang, Z., 2014. A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. IEEE Transactions on Cybernetics 44, 2274–2287.

Liu, F., Wu, J., Xue, S., Zhou, C., Yang, J., Sheng, Q., 2020a. Detecting the evolving community structure in dynamic social networks. World Wide Web 23, 715–733.

Liu, F., Wu, J., Zhou, C., Yang, J., 2019. Evolutionary community detection in dynamic social networks, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, pp. 1–7.

Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Yang, J., Philip, S.Y., 2020b. Deep learning for community detection: progress, challenges and opportunities, in: 29th International Joint Conference on Artificial Intelligence, IJCAI 2020, pp. 4981–4987.

Lyu, C., Shi, Y., Sun, L., Lin, C.T., 2022. Community detection in multiplex networks based on evolutionary multi-task optimization and evolutionary clustering ensemble. IEEE Transactions on Evolutionary Computation, 1–1doi:10.1109/TEVC.2022.3184988.

Ma, H., Wu, K., Wang, H., Liu, J., 2023. Higher-order knowledge transfer for dynamic community detection with great changes. IEEE Transactions on Evolutionary Computation, 1–1doi:10.1109/TEVC.2023.3257563.

Marbach, D., Costello, J.C., Küffner, R., Vega, N.M., Prill, R.J., Camacho, D.M., Allison, K.R., Kellis, M., Collins, J.J., Stolovitzky, G., 2012. Wisdom of crowds for robust gene network inference. Nature Methods 9, 796–804.

Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Dalla Fava, R., Califano, A., 2006. Aracne: an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. BMC Bioinformatics 7, 1–15.

Newman, M.E., 2003. The structure and function of complex networks. SIAM Review 45, 167–256.

Newman, M.E., 2006a. Finding community structure in networks using the eigenvectors of matrices. Physical review E 74, 036104.

Newman, M.E., 2006b. Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103, 8577–8582.

Newman, M.E., Watts, D.J., 1999. Renormalization group analysis of the small-world network model. Physics Letters A 263, 341–346.

Nitzan, M., Casadiego, J., Timme, M., 2017. Revealing physical interaction networks from statistics of collective dynamics. Science Advances 3, e1600396.

Nowak, M.A., May, R.M., 1992. Evolutionary games and spatial chaos. Nature 359, 826–829.

Palafox, L., Noman, N., Iba, H., 2012. Reverse engineering of gene regulatory networks using dissipative particle swarm optimization. IEEE Transactions on Evolutionary Computation 17, 577–587.

Pizzuti, C., 2017. Evolutionary computation for community detection in networks: A review. IEEE Transactions on Evolutionary Computation 22, 464–483.

Strogatz, S.H., 2001. Exploring complex networks. Nature 410, 268–276.

Szabó, G., Fath, G., 2007. Evolutionary games on graphs. Physics Reports 446, 97–216.

Tan, K.C., Feng, L., Jiang, M., 2021. Evolutionary transfer optimization—a new frontier in evolutionary computation research. IEEE Computational Intelligence Magazine 16, 22–33.

Teng, X., Liu, J., Li, M., 2019. Overlapping community detection in directed and undirected attributed networks using a multiobjective evolutionary algorithm. IEEE Transactions on Cybernetics 51, 138–150.

Tian, Y., Cheng, R., Zhang, X., Jin, Y., 2017. Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. IEEE Computational Intelligence Magazine 12, 73–87.

Tian, Y., Zhang, X., Wang, C., Jin, Y., 2020. An evolutionary algorithm for large-scale sparse multiobjective optimization problems. IEEE Transactions on Evolutionary Computation 24, 380–393. doi:10.1109/TEVC.2019.2918148.

Wang, C., Liu, J., Wu, K., Wu, Z., 2022. Solving multitask optimization problems with adaptive knowledge transfer via anomaly detection. IEEE Transactions on Evolutionary Computation 26, 304–318. doi:10.1109/TEVC.2021.3068157.

Wang, C., Liu, J., Wu, K., Yang, C., 2021. Learning large-scale fuzzy cognitive maps using an evolutionary many-task algorithm. Applied Soft Computing 108, 107441.

Wang, W.X., Lai, Y.C., Grebogi, C., Ye, J., 2011. Network reconstruction based on evolutionary-game data via compressive sensing. Physical Review E 83, 021021.

Wang, X., Dong, Z., Tang, L., Zhang, Q., 2023. Multiobjective multitask optimization-neighborhood as a bridge for knowledge transfer. IEEE Transactions on Evolutionary Computation 27, 155–169. doi:10.1109/TEVC.2022.3154416.

Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of ‘small-world’networks. nature 393, 440–442.

Wen, X., Chen, W.N., Lin, Y., Gu, T., Zhang, H., Li, Y., Yin, Y., Zhang, J., 2016. A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. IEEE Transactions on Evolutionary Computation 21, 363–377.

Wu, K., Hao, X., Liu, J., Liu, P., Shen, F., 2022a. Online reconstruction of complex networks from streaming data. IEEE Transactions on
Wu, K., Liu, J., 2017. Learning large-scale fuzzy cognitive maps based on compressed sensing and application in reconstructing gene regulatory networks. IEEE Transactions on Fuzzy Systems 25, 1546–1560.

Wu, K., Liu, J., Chen, D., 2019. Network reconstruction based on time series via memetic algorithm. Knowledge-Based Systems 164, 404–425.

Wu, K., Liu, J., Hao, X., Liu, P., Shen, F., 2021. An evolutionary multiobjective framework for complex network reconstruction using community structure. IEEE Transactions on Evolutionary Computation 25, 247–261.

Wu, K., Liu, J., Liu, P., Shen, F., 2022b. Online fuzzy cognitive map learning. IEEE Transactions on Fuzzy Systems 29, 1885–1898.

Wu, K., Liu, J., Liu, P., Yang, S., 2020. Time series prediction using sparse autoencoder and high-order fuzzy cognitive maps. IEEE Transactions on Fuzzy Systems 28, 3110–3121.

Wu, K., Liu, J., Wang, S., 2016. Reconstructing networks from profit sequences in evolutionary games via a multiobjective optimization approach with lasso initialization. Scientific Reports 6, 1–11.

Zhang, X., Zhou, K., Pan, H., Zhang, L., Zeng, X., Jin, Y., 2018. A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks. IEEE Transactions on Cybernetics 50, 703–716.

Zhao, J., Cheong, K.H., 2023. Obfuscating community structure in complex network with evolutionary divide-and-conquer strategy. IEEE Transactions on Evolutionary Computation, 1–1 doi:10.1109/TEVC.2023.3242051.

Zitzler, E., Laumanns, M., Thiele, L., 2001. Spea2: Improving the strength pareto evolutionary algorithm. TIK-report 103.