Cost-aware Robust Control of Signed Networks by Using a Memetic Algorithm

Lijia Ma, Jianqiang Li, Qiuzhen Lin, *Member, IEEE,* Maoguo Gong, *Senior Member, IEEE,* Carlos A. Coello Coello, *Fellow, IEEE,* and Zhong Ming

***Abstract*—The robust controllability (RC) of a complex system tries to select a set of dominating entities for the functional control of this entire system without uncertain disturbances, and the research on RC will help to understand the system's underlying functions. In this paper, we introduce the control cost in signed networks and present a cost-aware robust control (CRC) problem in this scenario. The aim of CRC is to minimize the cost to control a set of dominating nodes and transform a set of unbalanced links into balanced ones, such that the signed network can be robustly controlled without uncertain unbalanced factors (like nodes and links). To solve this problem, we first model CRC as a constrained combination optimization problem, and then present a memetic algorithm with some problem-specific knowledge (like the neighbors of nodes, the constraints of CRC, and the fast computation of the cost under each optimization) to solve this problem on signed networks. Extensive experiments on both real social and biological networks assess that our algorithm outperforms several state-of-the-art RC algorithms.**

***Index Terms*—Memetic algorithm, controllability, minimum dominating set, structural balance, signed networks.**

I. Introduction

THE robust controllability (RC) in complex systems has been attracting great attention in recent years due to its availability on supervising the systems' underlying functions and behaviors [1]–[3]. RC tries to determine a set of entities allowing the functional control of an entire system without uncertain disturbances [4]. The study of RC is driven by a deep understanding of the functionalities and behaviors of systems in biology, information, communication and society [3], [5]. For instance, RC governs undesirable behaviors (like the outbreak of disease, economic collapse, cascading failure, malicious attack and social unbalance) in social systems and discovers the function and activity of bimolecules (like genes, complexes and proteins) in biological systems [2].

This work was supported by the Joint Funds of the National Natural Science Foundation of China under Key Program under Grant U1713212, the National Natural Science Foundation of China under Grant 61803269, Grant 61876110, Grant 61672358, Grant 61572330, Grant 61772393 and Grant 61836005. Prof. C.A. Coello Coello gratefully acknowledges support from CONACyT project 1920 (Fronteras de la Ciencia) and from a SEP-Cinvestav 2018 project (application no. 4).

L. Ma, J. Li, Q. Lin and Z. Ming are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China.

M. Gong is with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi Province 710071, China.

C.A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN, México, D.F., 07360, México.

*Corresponding authors: J. Li (e-mail: lijq@szu.edu.cn) and Z. Ming (email: mingz@szu.eu.cn).

Advances in networking provide a significant boost on the studies of the RC in complex systems, providing a quantitative framework to model a complex system [6], [7]. Based on the network model, this RC in complex systems can be well shifted into the detection of modules or behaviors in networks. One representative work is the shift of the RC into a maximum matching problem which can be solved in polynomial time [1], aiming to discover a maximum set of matched links without sharing start or end nodes. However, this work is only applied to some complex systems modeled as unsigned networks with directed links. Following this work [7], Nacher and Akutsu [8] proposed a general mapping work for unsigned systems which transforms the RC into a minimum dominating set (MDS) problem [9], trying to select a MDS such that each non-dominating node is linked with at least one dominating node. For each node i, let $\mathbf{x}_i \in \{0,\ 1\}$ be its dominating label and set $\mathbf{x}_i = 1$ to a dominating node. For an unsigned network with n nodes, this MDS problem can be formulated as the following 0-1 quadratic problem [4], [10].

$$\begin{aligned} \min \quad & F(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{x}_i, \\ \text{s.t.} \quad & \mathbf{x}_i + \sum_{j \in \mathcal{T}_i} \mathbf{x}_j \geq 1, i = 1, 2, \ldots n, \\ & \mathbf{x} \in \{0,1\}^n. \end{aligned} \tag{1}$$

where $F(\mathbf{x})$ is the number of dominating nodes, and $\mathcal{T}_i$ denotes the set of nodes linked with node i. The constraint $\mathbf{x}_i + \sum_{j \in \mathcal{T}_i} \mathbf{x}_j \geq 1$ ensures that i is either selected as a dominating node or linked to at least one dominating node. It has been proved that the MDS problem is a reduction of a set covering problem and a graph partition problem, both of which are nondeterministic polynomial (NP)-hard [11].

To solve the MDS problem in unsigned networks, many algorithms have been proposed, including exact algorithms (e.g., the approximation algorithms [9] and integer linear programming [4]) and (meta)heuristic algorithms (e.g., the genetic algorithms (GAs) [12], greedy algorithms [13]–[16], ant colony optimizations [17] and memetic algorithms (MAs) [10]). Systematic experiments have shown their performance, especially regarding MAs. One representative MA for the MDS problem is HMA [10], which combines a GA with a tabu local search to discover high-quality solutions. Generally, an MA is a combination of global search (to facilitate exploration) and local search (to promote exploitation) [18]–[20]. The global search uses some heuristic algorithms to optimize a population of solutions while the local search adopts a problem-specific

greedy algorithm to refine a solution [18]–[20]. Moreover, MAs usually incorporate some problem-specific knowledge and learning techniques, which enables them to tackle large-scale problems [21], [22]. By integrating the advantages of exploration and exploitation, MAs have become a very promising alternative in artificial intelligence for solving discrete NP-hard problems in networks and tasks [20], [23], e.g., the MDS [10], balance transformation [21], big knapsack problem [22], influence maximization [24], maximum cover set [25], data-driven optimization [26], [27], influential nodes discovery [28], [29], network dismantling [30], multitask optimization [31], capacitated arc routing [32], balance transformation [21] and community robustness improvement [33].

Although some progress of the RC in unsigned networks has been made recently, the RC in signed networks is still a challenging problem, due to the existence of a balance constraint (i.e., the links between the dominating nodes and the other ones should be balanced [34], [35]). The aim of this RC is to find a minimum balanced dominating set which follows the dominating constraint in (1) and the balance constraint. A signed network has both positive and negative links, which reflects the conflicting relationships in systems, like cooperation/competition and support/oppose [36]. These conflicting links will increase the unbalance of networks [37]. As known from structural balance theory [21], [38], [39], the balance of links is determined by the most balanced clustering of a network having the minimum imbalance. However, this clustering is usually unknown *a priori* and its detection is also complicated. None of the existing algorithms [4], [10], [12]–[15], [17] can tackle the MDS problem together with the clustering problem for the RC in signed networks.

In practical applications, the RC in signed networks is generally solved by the construction of dominating nodes [4], [10] and the transformation of unbalanced links into balanced ones [21], [40], both of which will take some cost (e.g., money, time, bandwidth, place, material, etc.) [40]. For instance, constructing a financial regulatory agency takes some time and effort in trade systems, while migrating virtual machines from overloaded computers to low-loaded computers requires some energy and bandwidth in data center networks. Moreover, the scale of real signed systems (like the voting systems in Wiki) is large with thousands of nodes and millions of links, and it is challenging for (meta)heuristic algorithms to tackle these systems. Motivated by these scenarios, in this paper, we consider the control cost in signed networks, and present a cost-aware robust control (CRC) problem. To better solve this CRC problem in signed networks, we propose a memetic algorithm (called as MCRC) with problem-specific knowledge. Our main contributions are summarized as follows.

1) We present the CRC problem in signed networks which considers the control cost. This problem aims to control a signed network by constructing dominating nodes and transforming unbalanced links into balanced ones at a minimum cost.
2) We model the CRC problem as a constrained combination optimization problem, and then present the MCRC algorithm for solving this problem. For the presented MCRC algorithm to tackle large-scale networks, some problem-specific knowledge (like the neighbors of n-odes, the constraints of CRC, and the fast computation of the cost under each optimization) are incorporated.
3) Extensive experiments on two small-scale social networks, four medium-scale biological networks and three large-scale social networks show that MCRC has a better performance than the state-of-the-art algorithms for the control of signed networks.

The rest of the paper is organized as follows. Section II provides the notation adopted in the paper as well as a review of the previous related work on the MDS problem in networks. In Section III, we provide the system model and the formulation of the CRC problem. Section IV introduces the details of our algorithm. Our experimental results are presented in Section V, and we provide our concluding remarks and some possible paths for future work in Section VI.

II. Background

A. Notations and Definitions

1) Notations: We use italic lower-case letters, decorated letters and block upper-case letters to denote scalars, sets and matrixes, respectively. Bold italic lower-case letters are adopted to denote variables with unknown values *a priori*, and upper-case letters are used to represent criteria. Let $\mathcal{S}$ and $|\mathcal{S}|$ be a set and the number of elements in this set, respectively.

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-\}$ be a signed network with sets of nodes $\mathcal{V}$, positive links $\mathcal{E}^+$ and negative links $\mathcal{E}^-$. This $\mathcal{G}$ can be formulated as an adjacent matrix $\mathrm{A} = [a_{ij}]_{i,j\in\mathcal{V}}$, and each of its elements is represented as follows.

$$a_{ij} = \begin{cases} +1 & \text{if } e_{ij} \in \mathcal{E}^+, \\ -1 & \text{if } e_{ij} \in \mathcal{E}^-, \\ 0 & \text{if nodes } i \text{ and } j \text{ are not linked.} \end{cases}$$

For each node i, its neighbor $\mathcal{T}_i = \{j \in \mathcal{V} | a_{ij} \neq 0\}$ is defined as the set of nodes linked with i.

2) Definitions: Some definitions in signed networks, including the balance theory and the balanced dominating set, are given next.

Let the links marked with '$+$' and '$-$' denote a positive link and a negative link, respectively. In a triad signed network, the four possible types of relations are interpreted from the perspective of social psychology as follows [41]:

- $+++$: "my friend's friend is also my friend";
- $++-$: "my friend's friend is my enemy";
- $--+$: "my enemy's enemy is my friend";
- $---$: "my enemy's enemy is also my enemy".

Definition 1: Balance theory of Heider [41] for a signed triad: In signed triads, the relations $+++$, $--+$ and $---$ are balanced whereas the relation $++-$ is unbalanced.

Definition 2: Balance theory of Easley [39] for an arbitrary signed network: An arbitrary signed network is balanced if a clustering can be found such that all positive links are in the same cluster, whereas all negative links are located in different clusters. In this case, all triads in the network are balanced.

In reality, most systems are imbalanced. To evaluate their unbalance degree, the frustration $H(\mathbf{y})$, which computes the

number of unbalanced links, was proposed by Facchetti [40]. Let $\mathbf{y}_i = q$ be the cluster label of node i with q, and let $\mathbf{y} = \{\mathbf{y}_i\}_{i\in\mathcal{V}}$. This frustration $H(\mathbf{y})$ is computed as follows.

$$H(\mathbf{y}) = \sum_{e_{ij}\in\mathcal{E}^+\bigcup\mathcal{E}^-, i<j} h_{ij}(\mathbf{y}_i, \mathbf{y}_j),$$

where $h_{ij}(\mathbf{y}_i, \mathbf{y}_j)$ denotes the balance of a link e_{ij}, and is computed as follows.

$$h_{ij}(\mathbf{y}_i, \mathbf{y}_j) = \begin{cases} 0 & \text{if } a_{ij} = 1, \mathbf{y}_i = \mathbf{y}_j \text{ or } a_{ij} = -1, \mathbf{y}_i \neq \mathbf{y}_j, \\ 1 & \text{otherwise.} \end{cases}$$

Here, an edge e_{ij} is referred as a balanced link when $h_{ij}(\mathbf{y}_i, \mathbf{y}_j) = 0$.

Facchetti et al. [40] have also pointed out that i) the balanced clustering of a signed network has the minimum $H(\mathbf{y})$ and ii) an unbalanced network can be transformed into a balanced one by changing the label of its unbalanced links with a cost.

Definition 3: Balanced dominating set and RC in signed networks: Given a signed network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-\}$, a subset $\mathcal{D} \in \mathcal{V}$ is a balanced dominating set if each node $u \in \mathcal{V} - \mathcal{D}$ is linked with at least one node in $\mathcal{D}$, and meanwhile there exists a balanced clustering $\mathbf{y}$ such that all edges between the dominating nodes ($\mathcal{D}$) and the other ones ($\mathcal{V}-\mathcal{D}$) are balanced. The RC in signed networks aims to discover such a minimum balanced dominating set [34], [35].

B. Related Work for the RC in Unsigned Networks

The RC problem in a system can be shifted into a maximum matching problem and a MDS problem in unsigned networks with directed links and undirected links, respectively. The unmatched nodes in the maximum matching and the nodes in the dominating set are chosen as driver nodes to control the system through the external signals from controllers. Those models are widely used in some real controllability problems, like exact controllability [42] and target controllability [43]. Recent studies have shown that the maximum matching problem can be solved in polynomial time by a linear algorithm in [1], whereas the MDS problem is NP-hard [1], [8], [42], [43]. To solve this NP-hard problem, many algorithms, including the centrality based greedy algorithms (C-Greedy), evolutionary algorithms (EAs) and MAs, have been proposed.

1) C-Greedy: They normally begin with an empty set, and then choose the node with the maximum centrality value into the dominating set iteratively. Classical centralities include betweenness [4], backbone cost [13], degree [14], permutation [15], H-index [44], collective intelligence [45], etc. Generally, these algorithms benefit exploitation but lack of exploration.

2) EAs: They begin with a combinational optimization model with an objective (e.g., the minimum number of dominating nodes), followed by evolving a population of solutions using bio-inspired learning [46] iteratively to optimize the objective. The GAs and ant co-optimization algorithms are used to solve the MDS problem in [12], [46] with the objective in (1), which design novel genetic operators and ants' cooperative strategies to discover good solutions, respectively. Generally, these algorithms benefit exploration but lack of exploitation.

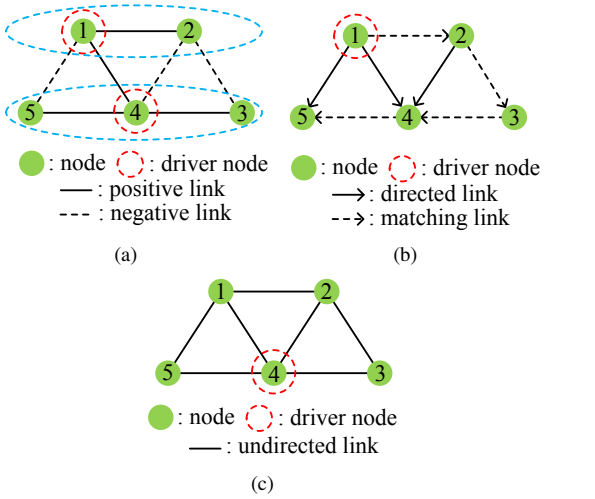


Fig. 1. Illustrations of the RC in (a) a toy signed network, (b) a toy unsigned network with directed links and (c) a toy unsigned network with undirected links. The nodes in an ellipse are divided into the same cluster, and the links between the driver nodes and the others are balanced.

3) MAs: Similar to EAs, MAs optimize an objective by evolving a population of solutions simultaneously, and benefit exploration. Contrary to EAs, they also possess a local search capability, thus benefiting exploitation [10], [20], [23].

Classical MAs for the MDS problem include HGA [12], A-COLS [12] and HMA [10]. HGA and ACOLS adopt a GA and ACO as the global search, respectively, and they use a greedy algorithm as their local search engine. HGA and ACOLS showed a better performance than their simplified versions (GA and Raka-ACO) without the local search, respectively, while HGA performs better than ACOLS. HMA first models the MDS problem as an unconstrained 0-1 programming problem with an adaptive penalty function, and then presents a GA combined with a tabu search to solve this problem. In HMA, a greedy rule in the tabu search is integrated to speed up the convergence, while a population updating strategy is adopted to maintain the diversity. The experiments in [10] showed the superiority of HMA over HGA and ACOLS.

Although some progress of the RC in unsigned networks has been made, the RC in signed networks is still a challenging problem, due to the existence of balance constraints (as shown in **Definition 3**). Fig. 1 provides schematic illustrations of the RC in different network models with 5 nodes. As shown in Fig. 1(a), to control the toy signed network, a minimum number of driver nodes ('1' and '4') are found while the balanced clustering ({{'1', '2'}, {'3', '4', '5'}}) of the network is detected. In this case, all nodes are controlled by at least one driver node, and the edges between the driver nodes ('1' and '4') and the other nodes ('2', '3' and '5') are balanced. As shown in Fig. 1(b), to control the toy directed network, a maximum number of matching links (e_{12}, e_{23}, e_{34} and e_{45}) are found, while the unmatched node '1' is chosen as the driver node. In this case, all nodes are controlled by the input signals on node '1' as there are directed control paths from node '1' to all matched nodes. Fig. 1(c) shows that the toy undirected network is fully controlled when a minimum number of driver

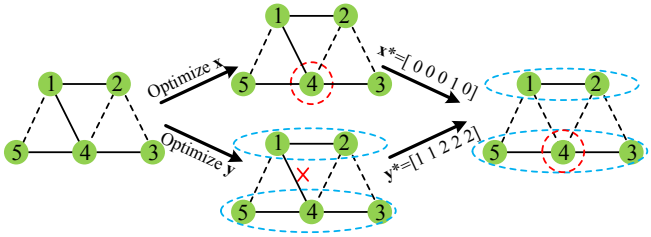


Fig. 2. Schematic illustration of controlling a toy system with 5 nodes and 7 links. The solid lines and dotted lines denote the positive and negative links, respectively. The nodes in an ellipse are divided into the same cluster and the node surrounding by a dotted red circle is the chosen driver node. The line marked with $\times$ is the unbalanced link. $\mathbf{x}^*$ and $\mathbf{y}^*$ are the optimal solution of $\mathbf{x}$ and $\mathbf{y}$, respectively. This system is fully controlled if we choose node 4 as the driver node and transform the unbalanced positive link e_{14} into a negative one.

nodes ('4') are chosen. In this case, all nodes are controlled by node '4' as they directly link with node '4'. Fig. 1 presents that the RC in the signed network is the most difficult as it needs to simultaneously find a minimum number of driver nodes and a balanced clustering.

III. System Model and Problem Formulation

In this section, the system model is first given, and then the CRC problem in signed networks is formulated.

A. System Model

We consider a control system with a signed network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-\}$ and an outside controller. In this system, the controller can control the signals of the dominating nodes and their balanced neighbors, like the traffic light in transportation networks, the concentration of transcription factor in gene regulatory networks, etc [1]. This system aims to provide an optimal solution or strategy by constructing dominating nodes $\mathcal{D}$ and transforming unbalanced links $\mathcal{E}_u$ into balanced ones for the functional control of a signed system.

In this system, each solution or strategy can be encoded into a pair of variables $(\mathbf{x}, \mathbf{y})$, which is detailed as follows.

$$(\mathbf{x}, \mathbf{y}) = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n; \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$$
$$\forall i \in \{1, 2, \ldots, n\},\ \mathbf{x}_i \in \{0, 1\},\ \mathbf{y}_i \in \{1, 2, \ldots, n\}.$$

Recall that $\mathbf{x}_i$ and $\mathbf{y}_i$ are the dominating label and the clustering label of node i, respectively. Moreover, given a solution $(\mathbf{x}, \mathbf{y})$, the sets of dominating nodes $\mathcal{D}$ and unbalanced links $\mathcal{E}_u$ are determined as follows.

$$\begin{aligned}\mathcal{D} =& \{i \in \mathcal{V} | \mathbf{x}_i = 1\}.\\ \mathcal{E}_u =& \{e_{ij} \in \mathcal{E}^+ | i \in \mathcal{D}, j \in \mathcal{V} - \mathcal{D}, \mathbf{y}_i \neq \mathbf{y}_j\} \bigcup \\ & \{e_{ij} \in \mathcal{E}^- | i \in \mathcal{D}, j \in \mathcal{V} - \mathcal{D}, \mathbf{y}_i = \mathbf{y}_j\}.\end{aligned}$$

There are many ways for controlling a signed system. For instance, all nodes in the system are selected as the driver nodes, which is time consuming and expensive. Here, we introduce the control cost (e.g., money, time, bandwidth, place, material, etc.) of constructing the dominating nodes and transforming unbalanced links, with the assumption that a higher cost is taken to dominate a node. Fig. 2 provides a schematic illustration to robustly control a toy system.

B. Problem Formulation

Given a signed system $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-\}$, the cost of constructing a driver node (we set its value as 1), and the cost c of transforming an unbalanced link, our CRC problem tries to simultaneously discover the optimal $\mathbf{x}$ and $\mathbf{y}$, so as to minimize the cost to robustly control the entire signed system by constructing driver nodes and transforming unbalanced links to balanced ones. This CRC problem can be formulated as the following constrained combinational optimization problem.

$$\begin{aligned}\min\ F(\mathbf{x}, \mathbf{y}) &= \sum_{i=1}^{n} \mathbf{x}_i + c \cdot \sum_{i \in \mathcal{D}} \sum_{j \in \mathcal{V} - \mathcal{D}} h_{ij}(\mathbf{y}_i, \mathbf{y}_j),\\ &= \sum_{i=1}^{n} \mathbf{x}_i + c \cdot \sum_{i=1}^{n} \sum_{j=1}^{n} h_{ij}(\mathbf{y}_i, \mathbf{y}_j) \cdot \mathbf{x}_i \cdot (1 - \mathbf{x}_j)\\ \text{s.t.}\quad & \mathbf{x}_i + \sum_{j \in \mathcal{T}_i} \mathbf{x}_j \geq 1, i = 1, 2, \ldots n,\\ & \mathbf{x} \in \{0, 1\}^n, \mathbf{y} \in \{1, 2, \ldots, n\}^n,\end{aligned} \tag{2}$$

where $F(\mathbf{x}, \mathbf{y})$ is the control cost to construct dominating nodes and transform unbalanced links. Recall that $\mathcal{D} = \{i \in \mathcal{V} | \mathbf{x}_i = 1\}$. Here, we mainly consider $0 \leq c \leq 1$ as a higher cost is taken to dominate a node generally. This formulation enables a comprehensive understanding of the effects of both conflicting links and their balance on the RC of signed networks.

Proposition. *The CRC problem is NP-hard.*

Proof: See Appendix. ■

IV. Our Solution

In this section, the MCRC algorithm is proposed for solving the CRC problem in (2) which is NP-hard.

A. General Framework of MCRC

MCRC adopts a framework of MAs, including initialization, selection, genetic operation and local search, as shown in **Algorithm 1**. MCRC begins with an initial population $\mathcal{P}(0)$ generated by the operation **Initialization()** described in Section IV-B, followed by evolving the population using the operations **Genetic_Operator()** and **Local_Search()** detailed in Section IV-C and Section IV-D, respectively. **Genetic_Operator()** executes the genetic operation for exploration, while **Local_Search()** performs the local search for exploitation. The evolutionary operations in lines 4-7 will terminate when the number of generations t reaches a maximum (predefined) number g_m.

Contrary to the MAs available [10], [21], MCRC enables to optimize $\mathbf{x}$ and $\mathbf{y}$ simultaneously. In the following subsections, the proposed MCRC are detailed.

B. Signed Label Propagation based Initialization

Initialization is used to generate a population of initial solutions $\mathcal{P}(0) = \{(\mathbf{x}^l(0), \mathbf{y}^l(0))\}$, $l = 1, 2, \ldots, n_{\mathcal{P}}$, where $n_{\mathcal{P}}$ is the size of the population, which is essential to find the search regions and promote the convergence of MAs [20].

Algorithm 1 Framework of MCRC

1: **Input**: population size: $n_{\mathcal{P}}$; parent population size: $n_{\mathcal{O}}$; crossover probability: p_c; mutation probability: p_m; maximum number of generations: g_m.
2: $\mathcal{P}(0) \leftarrow$ **Initialization**$(n_{\mathcal{P}})$.
3: **for** $(t = 1$ to $g_m)$ **do**
4: $\quad \mathcal{P}_P(t) \leftarrow$ Select $n_{\mathcal{O}}$ solutions from $\mathcal{P}(t-1)$ using a tournament selection.
5: $\quad \mathcal{P}_O(t) \leftarrow$ **Genetic_Operation**$\big(\mathcal{P}_P(t), n_{\mathcal{O}}, p_c, p_m\big)$, and find the solution $(\mathbf{x}^l(t), \mathbf{y}^l(t))$ with the minimum F in $\mathcal{P}_O(t)$.
6: $\quad (\mathbf{x}^l(t), \mathbf{y}^l(t)) \leftarrow$ **Local_Search**$\big((\mathbf{x}^l(t), \mathbf{y}^l(t))\big)$.
7: $\quad \mathcal{P}(t+1) \leftarrow$ Choose $n_{\mathcal{P}}$ solutions with low F values from $\mathcal{P}_P(t)$ and $\mathcal{P}_O(t)$.
8: **end for**

The label propagation algorithm (LPA) assigns the labels of nodes as those of their neighbors, which is widely used to detect the communities of networks due to its advantages with the low computational complexity, fast convergence and little prior knowledge about the structures required [47]. Following the rules of LPA, we present a constrained signed LPA (SLPA) (as shown in **Algorithm 2**), which uses the dominating label propagation and clustering label propagation to generate the $\mathbf{x}$ and $\mathbf{y}$ of the initial population $\mathcal{P}(0)$, respectively. For each solution $(\mathbf{x}^l, \mathbf{y}^l) \in \mathcal{P}(0)$ with $\mathbf{x}_i^l = 1$ and $\mathbf{y}_i^l = i$, $i = 1, 2, \ldots, n$, the dominating label propagation and the clustering label propagation work as follows.

1) Dominating Label Propagation: The dominating labels in $\mathbf{x}^l$ are propagated to each node $i \in \mathcal{V}$ as follows.

$$\mathbf{x}_i^l \leftarrow \begin{cases} 0 & \text{if } \sum\limits_{j \in \mathcal{T}_i} \mathbf{x}_j^l > 0, \text{ s.t. } \forall j \in \mathcal{T}_i, \sum\limits_{k \in \mathcal{T}_j} \mathbf{x}_k^l > 0 \text{ with } \mathbf{x}_i^l = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{3}$$

As known from (3), the dominating label is oppositely propagated from the nodes in $\mathcal{T}_i$ to node i. Specifically, if there is at least one neighbor of node i with dominating label 1, $\mathbf{x}_i^l \leftarrow 0$, and $\mathbf{x}_i^l \leftarrow 1$ otherwise. To satisfy the dominating constraint in (2), SLPA subjects to the constraint $\forall j \in \mathcal{T}_i, \sum_{k \in \mathcal{T}_j} \mathbf{x}_k^l > 0$ for each propagation $\mathbf{x}_i^l \leftarrow 0$.

2) Clustering Label Propagation: The clustering labels in $\mathbf{y}^l$ are propagated to each node $i \in \mathcal{V}$ as follows.

$$\mathbf{y}_i^l \leftarrow \underset{\mathbf{y}_j^l, j \in \mathcal{T}_i}{\arg\min} \sum_{k \in \mathcal{T}_i} h_{ik}(\mathbf{y}_j^l, \mathbf{y}_k^l). \tag{4}$$

In (4), the clustering label is propagated from the nodes in $\mathcal{T}_i$ to node i, which is subject to the minimum number of unbalanced links around node i.

Given an initial population $\mathcal{P}(0) = \{(\mathbf{x}^l(0), \mathbf{y}^l(0))\}$, $l = 1, 2, \ldots, n_{\mathcal{P}}$, SLPA in **Algorithm 2** works as follows. For each solution $(\mathbf{x}^l(0), \mathbf{y}^l(0))$, the $\mathbf{x}_i^l(0)$ and $\mathbf{y}_i^l(0)$ of each node i following a random order are updated by (3) and (4), respectively (in lines 7-10). Those steps are executed until $\mathbf{x}^l(0)$ and $\mathbf{y}^l(0)$ remain unchanged or the number of iterations reaches a maximum number g_t (in line 6). Here, we set g_t to 10 as the LPA algorithm can converge within 10 generations generally [47].

C. Genetic Operators

The genetic operators, including crossover and mutation, are used to guide MAs for exploration. Specifically, crossover

Algorithm 2 Initialization()

1: **Input**: Size of population: $n_{\mathcal{P}}$, and the maximum number of iterations: g_t.
2: **Output**: Initial solutions $\mathcal{P}(0)$.
3: Set $\mathcal{P}(0) = \{(\mathbf{x}^l(0), \mathbf{y}^l(0))\}$ with $\mathbf{x}_i^l(0) = 1$ and $\mathbf{y}_i^l(0) = i$, $i = 1, 2, \ldots, n$ and $l = 1, 2, \ldots, n_{\mathcal{P}}$.
4: **for** $(l = 1$ to $n_{\mathcal{P}})$ **do**
5: $\quad t \leftarrow 1$.
6: $\quad$ **while** $[(\mathbf{x}^l(0), \mathbf{y}^l(0))$ is not converged and $t \le g_t]$ **do**
7: $\quad\quad$ **for** (each node i in a random order) **do**
8: $\quad\quad\quad$ **Dominating label propagation**: Update $\mathbf{x}_i^l(0)$ based on (3).
9: $\quad\quad\quad$ **Clustering label propagation**: Update $\mathbf{y}_i^l(0)$ based on (4).
10: $\quad\quad$ **end for**
11: $\quad\quad t \leftarrow t + 1$.
12: $\quad$ **end while**
13: **end for**

creates offspring solutions by recombing and inheriting the features of parent solutions, while mutation encourages genetic diversity by tuning the solutions' structures.

1) Crossover: The two-way crossover [23] is widely used for community detection in networks due to its simplicity and good inheritance of clusters from parent solutions. Given two parent solutions, a node v is first randomly chosen, and then the nodes with the same cluster label as v in one parent solution propagate their cluster labels to those in the other parent solution. Here, a parallel two-way crossover is presented for the CRC problem whose solution contains both $\mathbf{x}(t)$ and $\mathbf{y}(t)$, and the nodes j with $\mathbf{x}_j(t) = 1$ are limited to execute the crossover of dominating labels for satisfying the dominating constraint. Specifically, given two parent solutions $((\mathbf{x}^a(t), \mathbf{y}^a(t))$ and $(\mathbf{x}^b(t), \mathbf{y}^b(t))$, and a node v, the parallel two-way crossover generates offspring solutions as follows.

$$\begin{aligned} &\mathbf{x}_j^a(t) \leftarrow \mathbf{x}_v^b(t), \ \{\forall j | \mathbf{y}_j^b(t) = \mathbf{y}_v^b(t), \mathbf{x}_j^a(t) \neq 1\}, \\ &\mathbf{y}_j^a(t) \leftarrow \mathbf{y}_v^b(t), \ \{\forall j | \mathbf{y}_j^b(t) = \mathbf{y}_v^b(t)\}, \\ &\mathbf{x}_k^b(t) \leftarrow \mathbf{x}_v^a(t), \ \{\forall k | \mathbf{x}_k^a(t) = \mathbf{y}_v^a(t), \mathbf{x}_k^b(t) \neq 1\}, \\ &\mathbf{y}_k^b(t) \leftarrow \mathbf{y}_v^a(t), \ \{\forall k | \mathbf{y}_k^a(t) = \mathbf{y}_v^a(t)\}. \end{aligned} \tag{5}$$

Fig. 3 gives a schematic illustration about the crossover. As shown in Fig. 3, this crossover enables to simultaneously exchange both the cluster labels and the dominating labels of a cluster of nodes, while preserving both the dominating constraint in (2). Moreover, it generates new offspring solutions which inherit some cluster and dominating labels from their parent solutions, thus improving the exploration of MCRC.

2) Mutation: Similar to the crossover operator, the neighbor-based mutation is also extended for solving the CRC problem. It works on each solution $(\mathbf{x}^l(t), \mathbf{y}^l(t))$ generated by the crossover. For each node $i \in \mathcal{V}$, its labels $(\mathbf{x}_i^l(t), \mathbf{y}_i^l(t))$ are updated with those of its neighbors with a mutation probability p_m. Specifically,

$$\begin{aligned} &\mathbf{x}_i^l(t) \ \leftarrow \ 1 - \mathbf{x}_v^l(t), \ \mathbf{y}_i^l(t) \ \leftarrow \ \mathbf{y}_v^l(t), \\ &\text{s.t. } \forall j \in \mathcal{T}_i, \quad \mathbf{x}_j^l(t) + \sum_{k \in \mathcal{T}_j} \mathbf{x}_k^l(t) > 0. \end{aligned} \tag{6}$$

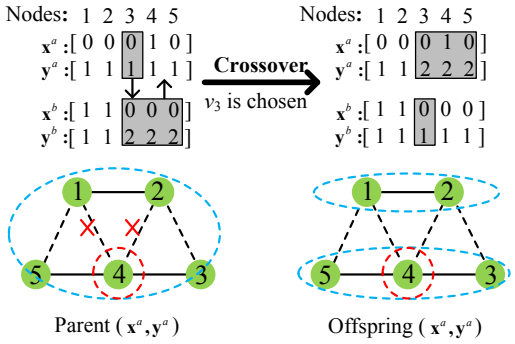


Fig. 3. Schematic illustration of the crossover on the parent solutions $(\mathbf{x}^a, \mathbf{y}^a)$ and $(\mathbf{x}^b, \mathbf{y}^b)$. The nodes surrounding by a dotted red circle is the chosen driver node while the line marked with $\times$ is the unbalanced link. After the crossover, a new solution $(\mathbf{x}^a, \mathbf{y}^a)$ with 1 dominating node and 0 unbalanced link is generated.

Algorithm 3 Genetic_Operator()

Input: Parent solutions: $\mathcal{P}_P(t)$, crossover probability: p_c and mutation probability p_m.
Output: Offspring solutions: $\mathcal{P}_O(t)$.
for [each pair of solutions in $\mathcal{P}_P(t)$] **do**
  Choose a node v randomly.
  if (A randomly generated value $r \leq p_c$) **then**
    Generate two offspring solutions based on (5), and put them into $\mathcal{P}_O(t)$.
  end if
end for // Crossover
for [each i of each solution $(\mathbf{x}^l(t), \mathbf{y}^l(t))$ in $\mathcal{P}_O(t)$] **do**
  if (A randomly generated value $r \leq p_m$) **then**
    Choose v from $\mathcal{T}_i$ randomly.
    Update $\mathbf{x}_i^l(t)$ and $\mathbf{y}_i^l(t)$ based on (6).
  end if
end for // Mutation

In (6), the constraint $\forall j \in \mathcal{T}_i, \quad \mathbf{x}_j^l(t) + \sum_{k \in \mathcal{T}_j} \mathbf{x}_k^l(t) > 0$ ensures that the generated solution follows the dominating constraint in the CRC problem.

The procedures of the genetic operators are summarized in **Algorithm 3**.

D. Local Search

In MAs, the local search is essential for exploitation and convergence. For the CRC problem, the refinement of a solution needs to optimize both $\mathbf{x}$ and $\mathbf{y}$, which is intractable due to their different search spaces and the correlation in the optimization processes. To solve this intractable problem, a serial optimization is first presented, and then a dominating label refinement and a clustering label refinement are proposed to optimize $\mathbf{x}$ and $\mathbf{y}$, respectively.

Serial optimization. Serial optimization is the most common strategy for solving the optimization problems with multiple variables [48]. In this way, one variable is first refined to be the optimum, and then the other ones are further refined.

To accelerate the convergence of the local search, a greedy strategy is used in the dominating and clustering label refinements, in which the label of each node is updated with that of its neighbor having the minimum F. However, this strategy has a high computational complexity as all the possible solutions for each refinement need to be considered and each refinement takes $O(m+n)$ time to compute F. To reduce its computational complexity, MCRC incorporates the following problem-specific knowledge into the local search for tackling large-scale networks.

1) **Constraint of CRC**. The dominating label of each node i is only updated with the one under the constraint of CRC, which reduces the number of possible solutions of $\mathbf{x}_i$ from 2 to $s \leq 2$.
2) **Neighbors of nodes**. The clustering label of each node i is only updated with that of its neighbors, which reduces the number of possible solutions of $\mathbf{y}_i$ from n to k_i, where k_i is the degree of node i.
3) **Computation of F under each refinement**. For each refinement on a node i, the F value is only calculated in its change part, i.e., $\mathbf{x}_i$ and $\sum_{j \in \mathcal{T}_i} h_{ij}(\mathbf{y}_i, \mathbf{y}_j) \cdot (\mathbf{x}_i \odot \mathbf{x}_j)$, where $\mathbf{x}_i \odot \mathbf{x}_j = 1$ if $\mathbf{x}_i \neq \mathbf{x}_j$, and $\mathbf{x}_i \odot \mathbf{x}_j = 0$ otherwise, which reduces the complexity of computing F from $O(m+n)$ to $O(|\mathcal{T}_i|)$.

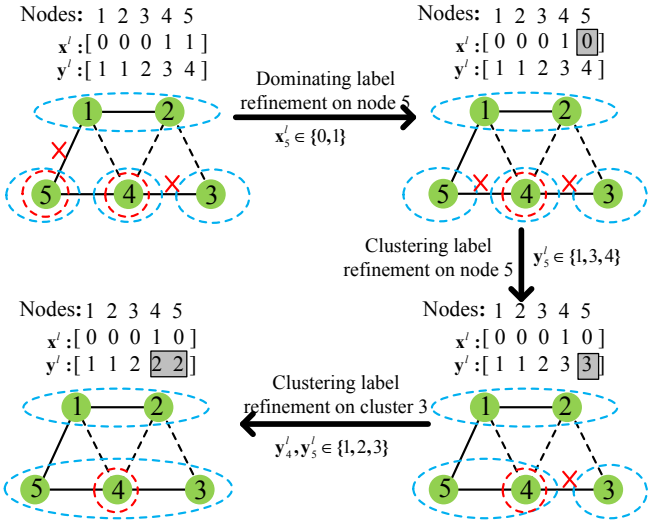


Fig. 4. Schematic illustration of the local search on the solution $(\mathbf{x}^l, \mathbf{y}^l)$.

Algorithm 4 Dominating_Refinement()/Node_Refinement()/ Cluster_Refinement()

Input: $(\mathbf{x}^l(t), \mathbf{y}^l(t))$.
$g = 0$;
while [$(\mathbf{x}^l(t), \mathbf{y}^l(t))$ is not converged and $g < g_t$] **do**
  for (each node i) **do**
    Update $\mathbf{x}_i^l(t)$/$\mathbf{y}_i^l(t)$/$\mathbf{y}_i^l(t)$ based on (7)/(8)/(9).
  end for
  $g = g + 1$;
end while

The dominating label refinement and the cluster label refinement optimize the labels of nodes from the first one to the last one. Given a solution $(\mathbf{x}^l(t), \mathbf{y}^l(t))$ and a node v_i, the refinements work as follows.

Dominating label refinement. The refinement of $\mathbf{x}_i^l(t)$

Algorithm 5 Local_Search()

1: **Input**: Best solution in offspring solutions: $(\mathbf{x}^l(t), \mathbf{y}^l(t))$.
2: $(\mathbf{x}^l(t), \mathbf{y}^l(t)) \leftarrow$ **Dominating_Refinement**$(\mathbf{x}^l(t), \mathbf{y}^l(t))$.
3: $(\mathbf{x}^l(t), \mathbf{y}^l(t)) \leftarrow$ **Node_Refinement**$(\mathbf{x}^l(t), \mathbf{y}^l(t))$.
4: $(\mathbf{x}^l(t), \mathbf{y}^l(t)) \leftarrow$ **Cluster_Refinement**$(\mathbf{x}^l(t), \mathbf{y}^l(t))$.

would result in the changes of the numbers of dominating nodes and unbalanced links (shown in Fig. 4). $\mathbf{x}_i^l(t)$ is updated by the label from $\{0, 1\}$ such that F is minimized. Specifically, $\mathbf{x}_i^l(t)$ is refined as follows.

$$\mathbf{x}_i^l(t) \leftarrow \underset{\mathbf{x}_i^l(t)\in\{0,\ 1\}}{\arg\min} \Big[\mathbf{x}_i^l(t) + c \cdot \Big(\sum_{j\in\mathcal{T}_i} h_{ij}(\mathbf{y}_i^l(t), \mathbf{y}_j^l(t)) \cdot (\mathbf{x}_i^l(t) \odot \mathbf{x}_j^l(t))\Big)\Big]. \quad (7)$$

Clustering label refinement for each node. The refinement of $\mathbf{y}_i^l(t)$ would change the number of unbalanced links (shown in Fig. 4). $\mathbf{y}_i^l(t)$ is refined as the clustering label of v_i's neighbors such that F is minimized. Specifically, $\mathbf{y}_i^l(t)$ is updated as follows.

$$\mathbf{y}_i^l(t) \leftarrow \underset{\mathbf{y}_u^l(t):u\in\mathcal{T}_i}{\arg\min} \sum_{j\in\mathcal{T}_i} h_{ij}(\mathbf{y}_u^l(t), \mathbf{y}_j^l(t)) \cdot (\mathbf{x}_i^l(t) \odot \mathbf{x}_j^l(t)). \quad (8)$$

Clustering label refinement for each cluster. As known from [49], the greedy search only with a node refinement can get easily trapped in a local optimum in the clustering process. This is the reason for decreasing F by merging small clusters into large ones. To solve this problem, the cluster label refinement in [49] is further extended to optimize $\mathbf{y}$. First, a novel graph with a number k of super-nodes is constructed, in which each super-node i is the set of nodes in cluster c_i. Then, for each super-node c_i, its clustering label $\mathbf{y}_{c_i}^l(t)$ is refined as follows (shown in Fig. 4).

$$\mathbf{y}_{c_i}^l(t) \leftarrow \underset{\mathbf{y}_{c_j}^l(t):c_j\in\mathcal{T}_{c_i}}{\arg\min} \sum_{u\in c_i} \sum_{v\in\mathcal{T}_{c_i}} h_{uv}(\mathbf{y}_{c_j}^l(t), \mathbf{y}_v^l(t)) \cdot (\mathbf{x}_u^l(t) \odot \mathbf{x}_v^l(t)). \quad (9)$$

All label refinements are executed until none of the labels is updated or the number of iterations reaches a maximum number g_t (set to 10 in this paper). **Algorithm 4** shows the operations of all label refinements, in which **Dominating_Refinement()** executes the dominating label refinement while **Node_Refinement()** and **Cluster_Refinement()** perform the clustering label refinements for nodes and clusters, respectively. **Algorithm 5** provides the whole framework of the local search.

E. Computational Complexity of MCRC

As shown in **Algorithm 1**, MCRC mainly consists of the following operations.

Initialization(). As known from **Algorithm 2**, Initialization() executes the loop $n_{\mathcal{P}}$ times in lines 4-13. Within this loop, an inner loop in lines 6-12 is repeated during the predefined g_t times at most. In this inner loop, as known from (3) and (4), the updates of $\mathbf{x}_i$ and $\mathbf{y}_i$ take the averaged computational complexity as $O(\bar{k}^2)$, where $\bar{k}$ is the average node degree of the network. Therefore, Initialization() has a computational complexity $O(n_{\mathcal{P}} \cdot g_t \cdot n \cdot \bar{k}^2)$.

Genetic_Operator(). As shown in **Algorithm 3**, Genetic_Operator() executes the loops in lines 3-8 and lines 9-14 approximately $n_{\mathcal{O}}/2$ and $n_{\mathcal{O}} \cdot 2 \cdot n$ times, respectively. The first **for** loop takes n operations to execute the crossover on two parent solutions (line 6), while the other one executes $\bar{k}^2$ operations at most (lines 11 and 12). Moreover, it takes $n_{\mathcal{O}} \cdot (n + m)$ operations to evaluate offspring solutions. Generally, $\bar{k} \ll m$ and $\bar{k}^2 \leq m$. Therefore, the computational complexity of Genetic_Operator() is $O(n_{\mathcal{O}} \cdot (n + m))$.

Local_Search(). As known from Section IV-D, by incorporating some problem-specific information, the computational complexity of the updates for each node's label in (7-9) is reduced to $O(\bar{k}^2)$. Those updates for all nodes are iterated g_t times at most. Therefore, Local_Search() has a computational complexity $O(g_t \cdot n \cdot \bar{k}^2)$.

In **Algorithm 1**, MCRC executes Genetic_Operator() and Local_Search() with g_m times. Therefore, MCRC has a computational complexity $O\Big(g_m \cdot n_{\mathcal{O}} \cdot (n+m) + (g_m + n_{\mathcal{P}}) \cdot g_t \cdot n \cdot \bar{k}^2\Big)$, which makes it possible to tackle large-scale networks.

V. Experimental Results

In this section, we test MCRC on nine real-world networks, and compare MCRC with respect to six classical algorithms in three criteria. In the following, the experimental settings are first listed, and then a comparison of MCRC with the six algorithms on nine networks is given. Finally, the effects of some parameters are discussed.

A. Experimental Settings

1) Experimental Networks: Two small-scale social networks, four medium-scale biological networks and three large-scale social networks are chosen. The RC in social systems can effectively supervise their undesirable behaviors (e.g., conflicts, cascading failures and malicious attacks [2]) while that in biological systems can comprehensively understand the function of biomolecules (e.g., genes, complexes, proteins, etc.). The basic properties of those networks are shown in Table I, and their detailed descriptions are given as follows.

Gahuku-Gama Subtribes Network (GGS): This network represents the cultures (political unions/oppositions) of of highland 16 Gahuku-Gama subtribes in New Guinea [50].

Illustrative Signed Network (ISN): This network constructed by Yang et al. [51] consists of 28 nodes and 42 links, denoting the cooperations and competitions among individuals.

EGFR Network: This network consists of 330 vertices and 779 links. The nodes and links represent the state of the Epidermal growth factor in receptor pathway signaling systems and the transition of the state of the systems, respectively [52].

Yeast Network: This network represents the gene regulatory network of S.cerevisiae with 690 transcription factors and 1,082 binding site inter-actions [53].

Macrophage Network: This network models the interaction between Macrophage's molecules [54].

TABLE I
BASIC PROPERTIES OF THE TEST NETWORKS. m_+ AND m_- ARE THE NUMBERS OF POSITIVE AND NEGATIVE LINKS, RESPECTIVELY.

Networks	n	m	m_+	m_-	Fields
GGS	16	58	29	29	Social
ISN	28	42	30	12	Social
EGFR	329	779	515	264	Biological
Yeast	690	1,080	860	220	Biological
Macrophage	678	1,425	947	478	Biological
E.coli	1,461	3,215	1,879	1,336	Social
Wiki	7,114	10,0321	78,792	21,529	Social
Wiki-rfa	11,276	17,0973	132,988	37,985	Social
Slashdot	22,936	288,638	217,563	71,075	Social

E.coli Network: This network is the gene regulatory network of Escherichia coli [55] with 1,476 operon and 3,215 transcriptional interactions. It can be downloaded from RegulonDB database (http://regulondb.ccg.unam.mx).

Wiki-ele Network: This network is composed of 7,114 electors and candidates. In this network, electors cast votes to candidates with positive and negative elections. A detailed description of this network is given in [40], [56].

Wiki-rfa Network: This network represents the votes of Wikipedia members to editors who want to become an administrator. The votes can be supporting, neutral or opposing. Here, a symmetrical network (shown in http://snap.stanford.edu/data/wiki-RfA.html and [21]) is used as the test network by discarding the neutral votes and the same voter/votee pairs.

Slashdot network: This network collects the tags (including friends or foes) among users in a news website entitled as "Slashdot". Its original version in website http://snap.stanford.edu/data/#signnets contains 77,357 users and 516,575 tags. Here, its largest connected component with dense links is used as the test network, containing of 22,936 users, 217,563 tags labelled as friends and 71,075 tags written as foes.

2) Comparison Algorithms: Due to the complexity of the RC in signed networks, none of the algorithms available can solve it well. Here, to verify the proposed MCRC, three classical algorithms for the RC in unsigned networks and three variants of MCRC are chosen as the comparison algorithms.

Hybrid memetic algorithm (HMA) [10]: It first formulates the MDS problem as a constrained 0-1 programming problem, and then designs an MA for solving this problem. The HMA adopts a GA and a tabu search as the global search and the local search, respectively.

Greedy algorithm (Greedy) [14]: It solves the MDS problem by a greedy strategy in which the nodes are gradually selected as dominating nodes based on their degrees.

Branch-and-bound algorithm (BBA) [4]: It models the controllability of proteins in Yeast networks as a MDS problem and uses the BBA to solve this problem. By solving a set of linear programming relaxation problems, BBA can effectively discover the best solution of MDS in its search tree based on the relaxation of the binary constraint to variables.

GARC, MRC: They are the variants of MCRC without the local search and the cluster label refinement, respectively.

GRC: It is the variant of MCRC without the global search.

The reason for choosing HMA, Greedy and BBA as the comparison algorithms is to show the superiority of MCRC over the state-of-the-art algorithms (MAs, greedy algorithms and linear programming) on the RC in signed networks. A comparison of MCRC with GARC and MRC tries to show the effectiveness of the devised local search engine in terms of convergence and effectiveness, while a comparison of MCRC with GRC aims to show the advantages of a memetic optimization framework over a greedy optimization framework.

TABLE II
PARAMETERS SETTINGS OF SOME ALGORITHMS USED IN OUR COMPARATIVE STUDY.

Algorithm	g_m	$n_{\mathcal{P}}$	$n_{\mathcal{O}}$	p_c	p_m
MCRC	300	50	5	0.9	0.15
GARC	300	50	5	0.9	0.15
MRC	300	50	5	0.9	0.15
GRC	300	50	5	0.9	0.15
HMA	300	50	5	1	—

3) Criteria: To verify the performance of all algorithms, the cost-aware objective F, the number of dominating nodes D and a common index in statistical hypothesis testing (P-value of tt-test [57]) are adopted. The F evaluates the overall performance of all algorithms on controlling a signed network with a minimum cost, while the D measures the performance of some algorithms (like BBA, Greedy and HMA) without balanced clustering techniques on solving the MDS problem. The P-value is used to determine the difference of solutions obtained by all algorithms. Specifically, if the P-value is larger than a significance level 5%, there is no significant difference between the solutions of the two algorithms. Here, each P-value is computed based on a comparison of our MCRC and other algorithms on the control cost F.

4) Simulation Settings: MCRC and its variants GARC, MRC and GRC are simulated by Matlab on a PC with Intel(R), Core (TM), i7-6700 CPU and 3.41 GHZ, while HMA, Greedy and BBA are coded with C++. For each network, all algorithms, excepted for BBA and Greedy which are stable, are run for 30 trials with the parameters shown in Table II.

B. Experiments on Real-world Networks

1) Experiments on Small-scale Social Networks: All algorithms are tested on two small-scale social networks, i.e., GGS and ISN, and the statistic values of criteria are summarized in Table III, including the average F, average D and P-values over 30 independent trials. The best statistic values are highlighted in **boldface** for each network. The results illustrate that MCRC has lower F values than the other algorithms, and has similar D values as HMA, GARC and MRC. Moreover, except for MRC, the P-values are smaller than 0.05, indicating that MCRC significantly outperforms BBA, HMA, GARC and GRC on the cost of controlling those two small-scale networks.

To verify the effectiveness of the presented MA optimization framework and local refinements, comparisons of MCRC with its variants are also made in Table III. The results show that GARC and GRC have higher F and D values than MRC and MCRC. This is reasonable as GARC (GRC) with a GA (local refinement) optimization framework favors exploration (exploitation) but lacks of exploitation (exploration), whereas both MRC and MCRC with an MA optimization framework

TABLE III
STATISTICAL RESULTS OF ALL ALGORITHMS USED IN OUR COMPARATIVE STUDY OVER 30 INDEPENDENT TRIALS ON TWO SMALL-SCALE SOCIAL NETWORKS. THE RESULTS ARE AVERAGED OVER 30 INDEPENDENT TRIALS. '—' DENOTES THAT THE ALGORITHM WAS INCOMPARABLE OR USELESS.

Networks	Indexes	BBA	Greedy	HMA	GARC	MRC	GRC	MCRC
GGS	F	12.00	11.50	5.500	3.517	2.100	4.017	**2.067**
	D	5.000	4.000	**2.000**	2.800	2.067	3.567	2.067
	P-value	1.2E-32	1.9E-27	1.3E-34	1.1E-13	**0.6453**	2.1E-12	—
ISN	F	18.50	15.50	10.77	9.925	8.447	10.19	**8.350**
	D	15.00	11.00	8.000	7.878	**7.575**	10.19	8.144
	P-value	6.195E-40	1.6E-35	6.0E-30	9.2E-19	**0.5876**	6.1E-13	—

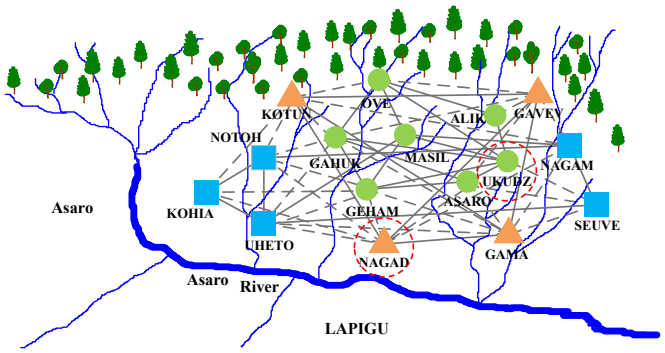


Fig. 5. Solution for the robust control of the GGS network. The dominating nodes are surrounded with the dotted oval. Nodes in different clusters are drawn using different shapes.

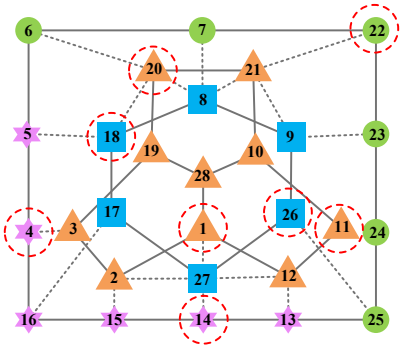


Fig. 6. Solution for the robust control of the ISN network. The dominating nodes are surrounded with a dotted oval.

are able to profit from the advantages of both exploration and exploitation. Specifically, by incorporating the dominating label refinement into GARC, MRC has a better performance than GARC, especially in terms of D. This validates the effectiveness of the presented dominating label refinement on the minimization of the dominating nodes. Similarly, by incorporating GRC into GARC, MCRC has a better performance than GRC, GARC and MRC in terms of F and D, which shows the superiority of the proposed MA optimization framework as well as the effectiveness of the clustering label refinement procedure.

As shown in Table III, the baseline MAs (HMA and MRC) have similar D values as our MCRC, but they have higher F values than MCRC. This is because they neglect the balance properties of signed networks and have no optimal strategy for discovering a balanced clustering with a minimum number of unbalanced links.

To visualize the robust control of signed networks, we plot the domination and balance divisions of the small-scale GGS and ISN networks detected by MCRC with the minimum F in Figs. 5 and 6, respectively. As shown in Fig. 5, the GGS network can be dominated when the nodes marked as "NAGAD" and "UKUDZ" are chosen as dominating ones, and the links between the dominating nodes and the other ones are balanced when the network is classified into three clusters. In other words, the GGS network is robustly controlled with the minimum F as 2 by constructing the "NAGAD" and "UKUDZ" as dominating nodes. From Fig. 6, we can see that for the ISN network, the optimal control strategy of MCRC suggests that its nodes "1, 4, 11, 14, 18, 20, 22, 26" should be chosen as dominating nodes, and meanwhile it should be divided into 4 clusters.

2) Experiments on Medium-scale Biological Networks: To further validate the superior performance of MCRC, we test all the algorithms adopted in our comparative study on four medium-scale biological networks and record the corresponding statistical criteria values in Table IV. We can obtain similar observations from the comparison results on the medium-scale biological networks and the small-scale social networks. Specifically, the best results are obtained by MCRC for all biological networks (100%) in terms of F. Moreover, MRC shows a better performance than GARC. In addition, with the framework of memetic optimization and the dominating label refinement, HMA, MRC and MCRC have a good performance in terms of D. In addition, the P-values between MCRC and the other algorithms are smaller than 0.05, indicating that BBA, HMA, GARC, GRC and MRC are not so effective to solve the RC in the medium-scale biological networks.

As shown in Table IV, for most networks, GARC and MRC have lower D values than GRC, whereas they have higher F values than GRC. This is because the population based methods (GARC and MRC) favor exploration whereas they have difficulties to converge to local optimal solutions in a short time. Moreover, discovering dominating nodes has smaller search spaces than detecting the balanced clustering in a signed network. In addition, GRC has a local refinement to the cluster label of signed networks, which makes it possible to discover better solutions than the population based methods (GARC, MRC and HMA) in terms of F.

It can be seen from Table IV that except for MCRC, all algorithms have difficulties to find out good solutions due to the complexity of the CRC problem in signed networks (i.e., the large search space with $2^n \cdot n^n$ possible solutions for a

TABLE IV
STATISTICAL RESULTS OF ALL ALGORITHMS USED IN OUR COMPARATIVE STUDY OVER 30 INDEPENDENT TRIALS ON FOUR MEDIUM-SCALE BIOLOGICAL NETWORKS.

Networks	Indexes	BBA	Greedy	HMA	GARC	MRC	GRC	MCRC
EGFR	F	169.0	166.5	147.2	196.3	146.8	133.9	**94.07**
	D	87.00	81.00	**62.95**	112.6	68.17	119.5	69.43
	P-value	3.0E-19	1.0E-26	5.1E-38	3.9E-44	4.7E-33	1.1E-23	—
Yeast	F	187.5	249.5	165.8	239.3	157.0	243.3	**119.6**
	D	118.0	157.0	**95.00**	147.8	98.73	167.6	96.27
	P-value	1.1E-45	7.4E-54	4.6E-37	1.6E-34	2.7E-34	1.4E-21	—
Macrophage	F	325.0	326.0	261.0	404.0	302.8	267.1	**206.2**
	D	197.0	186.0	**148.5**	244.4	187.3	160.2	159.5
	P-value	3.5E-46	2.7E-46	2.5E-47	8.0E-40	1.2E-35	1.4E-23	—
E.coli	F	641.5	729.5	618.5	742.7	540.4	355.0	**287.7**
	D	140.0	282.0	**109.0**	264.1	141.7	275.2	135.6
	P-value	4.1E-34	1.8E-37	1.1E-46	1.2E-40	1.0E-36	2.8E-17	—

TABLE V
STATISTICAL RESULTS OF ALL ALGORITHMS USED IN OUR COMPARATIVE STUDY OVER 30 INDEPENDENT TRIALS ON THREE LARGE-SCALE SOCIAL NETWORKS.

Networks	Indexes	BBA	Greedy	HMA	GARC	MRC	GRC	MCRC
Wiki	F	7,121	8,079	7,100	10,575	10,616	5,667	**3,280**
	D	1,148	4,490	**1,117**	4,218	4,212	4,246	2,419
	P-value	5.6E-33	9.1E-36	1.6E-33	1.9E-41	2.5E-40	2.8E-27	—
Wiki-rfa	F	12,286	13,417	12,243	18,089	18,111	9,912	**5,016**
	D	1,625	7,289	**1,623**	7,070	7,074	7,230	3,775
	P-value	2.3E-64	3.5E-66	2.0E-41	6.1E-55	1.9E-52	2.5E-46	—
Slashdot	F	—	19,756	—	44,994	25,685	23,316	**16,159**
	D	—	9,936	—	15,059	**6,425**	14,605	6,693
	P-value	—	2.3E-40	—	1.9E-53	1.7E-7	7.6E-44	—

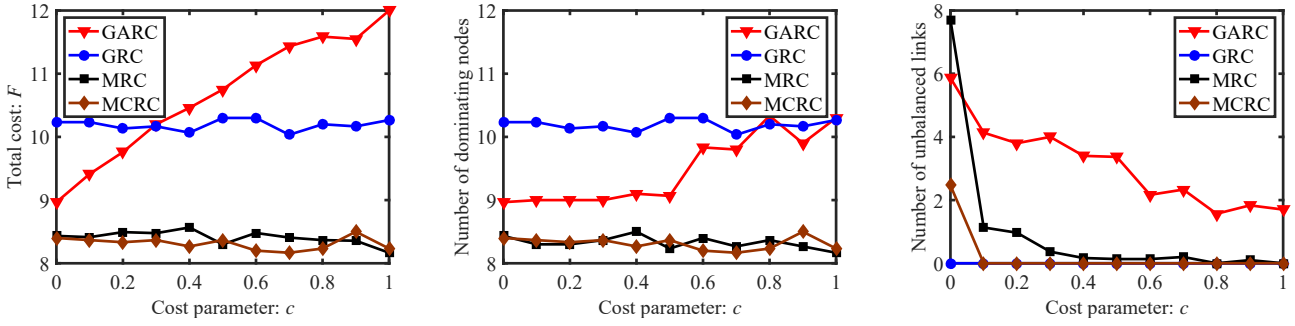


Fig. 7. Different criteria VS. Cost parameter c for controlling the ISN network. (a) the total control cost F, (b) the number of dominating nodes D and (c) the number of unbalanced links.

signed network with n nodes, the intractable optimization on the dominating label and clustering label, and the dominating constraints in the search of a feasible solution). MCRC can achieve a high-quality solution as it designs a problem-specific GA, a dominating refinement and a cluster refinement for exploration and exploitation. Moreover, it devises an effective strategy to solve the dominating constraints which avoids searching for infeasible solutions in the initialization, and when applying the genetic operators and the local search.

3) Experiments on Large-scale Social Networks: To discuss their applicability to real systems with thousands of nodes and links, all algorithms are tested on three large-scale social networks (Wiki, Wiki-rfa, and Slashdot), and the corresponding statistical results are collected in Table V. From Table V, we can get similar conclusions as those on the small-scale social networks. For instance, MCRC has a superior performance than the other algorithms in terms of the control cost F; HMA has the best performance on discovering dominating nodes without considering the control cost; and the P-values between MCRC and the other algorithms are smaller than 0.5.

The comparisons on large-scale social networks produce results which are different from those obtained on the small-scale social networks. For instance, MRC has obviously a better performance than GARC in the small-scale social networks, while it has a performance similar to that of GARC in the large-scale networks. In other words, for the large-scale networks, the dominating refinement has difficulties to improve the performance of GARC. This was somehow expected since both the solution space and the number of local optimal solutions increase exponentially with the node sizes of the networks, making MRC to get easily trapped into local optimal solutions after a predefined number of generations.

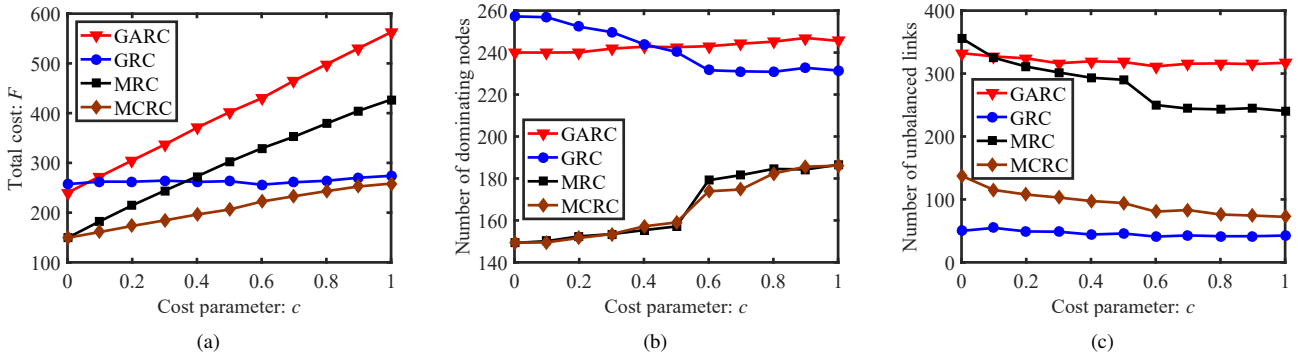


Fig. 8. Different criteria VS. Cost parameter c for controlling the EGFR network. (a) F, (b) D and (c) the number of unbalanced links.

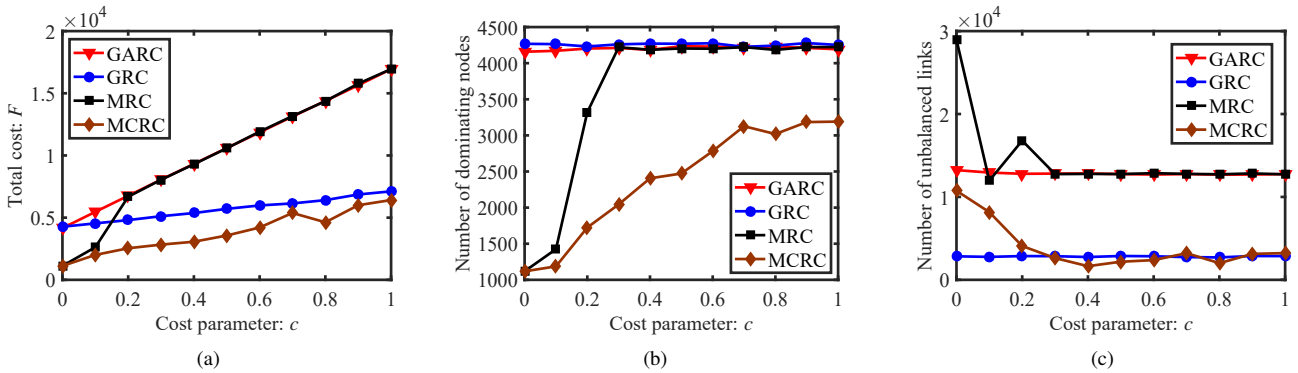


Fig. 9. Different criteria VS. Cost parameter c for controlling the Wiki network. (a) F, (b) D and (c) the number of unbalanced links.

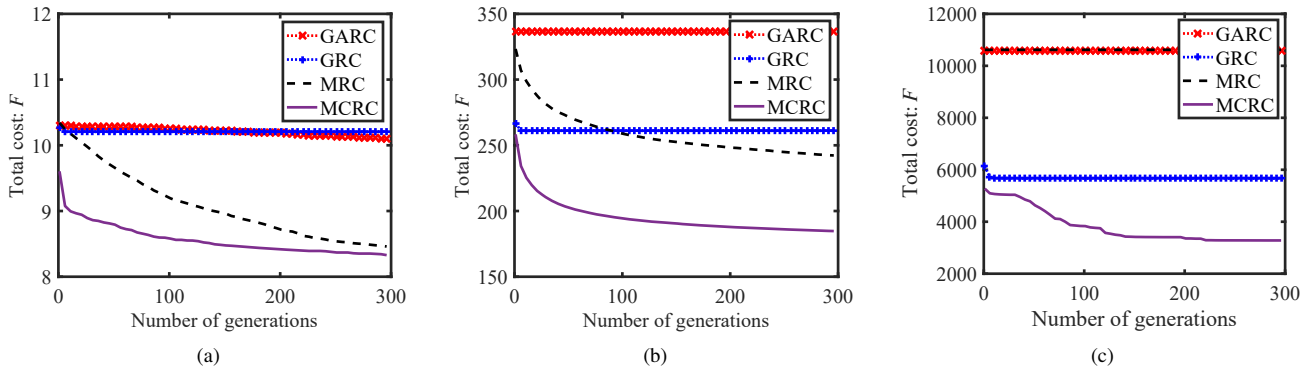


Fig. 10. Convergence of the algorithms (GARC, GRC, MRC and MCRC) on the (a) ISN network, (b) EGFR network and (c) Wiki network.

C. Effects of Parameter Settings

In this part, MCRC and its variants are tested on the small-scale ISN network, the medium-scale EGFR network and the large-scale Wiki network with different parameters settings to investigate the effects of these parameters.

In the devised optimization model shown in (2), there is a key parameter (i.e., c), denoting the ratio of the cost of transforming an unbalanced link to the cost of constructing a dominating node. Generally, $0 \leq c \leq 1$ as it takes less cost to control an unbalanced link. Here, the c value is varied from 0 to 1 with the interval 0.1, and the results with different c are recorded in Figs. 7-9. The results demonstrate that i) MCRC has lower F values than the other algorithms for all c values; ii) with the increase of c, the F values of all algorithms increase, while the number of dominating nodes (unbalanced links) increases (decreases); and iii) the difference of F values between MCRC and the other algorithms becomes larger. This is because with the increase of c, it has a higher cost to control unbalanced links, and the other algorithms have no effective strategies for discovering the balanced clustering with a minimum number of unbalanced links.

Moreover, to investigate the convergence of MCRC, we show the variation of F averaged over 30 independent trials with the number of iterations in Fig. 10. The results illustrate that for the small-scale and medium-scale networks, all algorithms can quickly converge to good solutions. Regarding

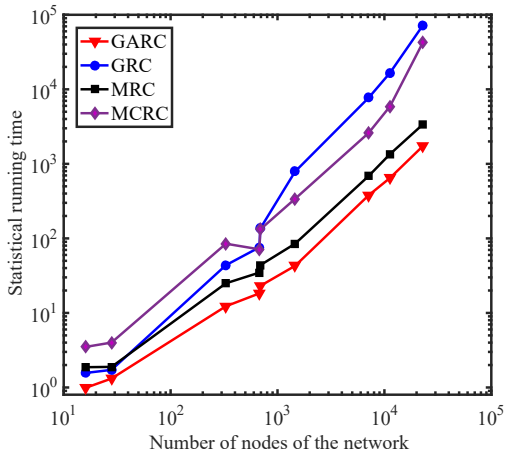


Fig. 11. Statistical running time of the algorithms used in our comparative study VS. Number of nodes of the networks.

the large-scale Wiki network, MCRC can quickly converge, whereas the other algorithms have difficulties to converge within 300 generations to good solutions. These phenomena demonstrate the superiority of MCRC in terms of convergence.

Finally, the average CPU running times taken by GARC, MRC, GRC and MCRC simulated using Matlab are displayed in Fig. 11 with different network sizes. Fig. 11 shows that the running times of the algorithms compared have an increase which is approximately linear with the number of nodes in the networks, especially for the networks with less than 10^3 nodes. The comparison between GRC and MCRC illustrates that it takes much time to execute the local search for large-scale networks, and MCRC can effectively reduce the CPU running time of GRC due to the exploration of the GA. To conclude, MCRC takes a reasonably longer time than GARC and MRC when solving the CRC problem in signed networks.

VI. Conclusions

In this paper, we presented a cost-aware robust control problem in signed systems, which considers the control cost and the functional control of an entire system by constructing dominating nodes and transforming unbalanced links into balanced ones. To solve this problem, we first modeled it as a constrained optimization problem, which combines the MDS problem with the balanced clustering problem in signed networks, and then presented the MCRC algorithm with some problem-specific knowledge (like the neighbors of nodes, the constraints of CRC, and the fast computation of the cost under each optimization). With these knowledge, MCRC can effectively reduce the computational complexity and accelerate the convergence towards good solutions for the RC problem in large-scale signed networks. Systematical experiments on nine real networks demonstrated the advantages of MCRC over the state-of-the-art algorithms for the control of signed networks, and the effectiveness of the designed memetic optimization framework and local search mechanism.

This work was done under the assumption that all nodes and edges of the network can be controlled, which may

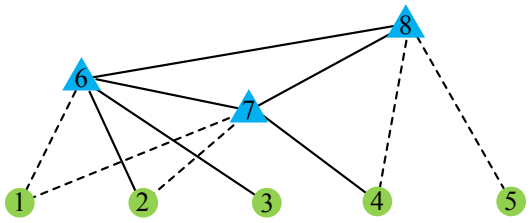


Fig. 12. Schematic illustration of the graph construction processes for a toy cover set problem with $\mathcal{U} = \{1,2,3,4,5\}$ and $\mathcal{S} = \{\{1,2,3\},\{1,3,4\},\{4,5\}\}$. The nodes plotted with triangle and circle denote the nodes in $\mathcal{D}$ and $\mathcal{O}$, respectively. The solid and dotted lines represent the positive links and the negative links, respectively.

be inconsistent in some applications. As part of our future work, we will study the RC in signed networks with more constraints, e.g., some nodes and links cannot be controlled, adjacent nodes cannot be controlled simultaneously, etc. Moreover, we will study the robust control of multiplex networks. Controlling a multiplex network is difficult as the control of one platform cannot guarantee that of the other platforms. Finally, inspired by the work in [22], we will study a novel compressed representation of memes (solutions) on the RC of signed networks, which enables the presented memetic algorithm (MCRC) to quickly tackle large-scale networks.

Appendix
Proof of the NP-hard of the CRC problem

To prove the NP-hard of the CRC problem, a well-known NP-hard problem, the set cover problem [11], [58], is reduced to the special instance of CRC with $c = 0$.

1) **Set cover**: Let $\mathcal{U} = \{1, 2, \ldots, m\}$ be a universe, and let $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q\}$ be a family, in which $\mathcal{S}_i \subset \mathcal{U}$, $i = 1, 2, \ldots, q$ and the union of $\mathcal{S}$ equals $\mathcal{U}$. Given an integer k, does the family have a set $\mathcal{C} \subset \mathcal{S}$ with $|\mathcal{C}| \leq k$ such that the union of $\mathcal{C}$ equals $\mathcal{U}$?
2) **Special instance of CRC**: Given a signed network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-\}$ and an integer k, does the network $\mathcal{G}$ have a set $\mathcal{D} = \{i \in \mathcal{V} : \mathbf{x}_i = 1\}$ with $|\mathcal{D}| \leq k$ such that each node is linked with one of nodes in $\mathcal{D}$?

Given the universe $\mathcal{U}$, family $\mathcal{S}$ and an integer k, the reduction processes are shown as follows.

First, we construct a signed graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-\}$ as follows: i) each element in $\mathcal{U}$ is modeled as a node; ii) a number $|\mathcal{S}|$ of nodes $\mathcal{O} = \{o_i\}$, $i = 1, 2, \ldots, |\mathcal{S}|$, are constructed, and iii) each node $o_i \in \mathcal{O}$ is (positively or negatively) linked with all the other nodes in $\mathcal{O}$, and is linked with all nodes in $\mathcal{S}_i$. Fig. 12 gives a schematic illustration of the graph construction processes for a toy cover set problem with $\mathcal{U} = \{1,2,3,4,5\}$, $\mathcal{S} = \{\{1,2,3\},\{1,3,4\},\{4,5\}\}$, and $\mathcal{O} = \{6,7,8\}$.

Second, we prove that $\mathcal{G}$ contains a dominating set $\mathcal{D}$ with $|\mathcal{D}| = k$ if and only if there is a set cover $\mathcal{C} \subset \mathcal{S}$ with $|\mathcal{C}| = k$ such that the union of $\mathcal{C}$ equals $\mathcal{U}$. This proof consists of the following two parts:

Step 1: Let's assume that $\mathcal{C} = \{\mathcal{S}_{i_1}, \mathcal{S}_{i_2}, \ldots, \mathcal{S}_{i_k}\}$ with $|\mathcal{C}| = k$ is a feasible solution for the set cover problem. As known from the constructed graph, for each $\mathcal{S}_{i_a}$, $a =$

$1, 2, \ldots, k$, its nodes are linked with the node $o_{i_a} \in \mathcal{O}$. Moreover, as known from the definition of covering set, the $\mathcal{C}$ contains all nodes in $\mathcal{U}$, and a node in $\mathcal{O}$ is connected with all the other nodes in $\mathcal{O}$. Therefore, $\mathcal{D} = \{o_{i_1}, o_{i_2}, \ldots, o_{i_k}\}$ is a dominating set with size k in $\mathcal{G}$.

Step 2: Let's assume that $\mathcal{D} = \{i_1, i_2, \ldots, i_k\}$, $i_a \in \mathcal{V}$ and $a = 1, 2, \ldots, k$, is a dominating set in $\mathcal{G}$. As known from the definition of dominating set, all nodes in both $\mathcal{U}$ and $\mathcal{O}$ of $\mathcal{G}$ are linked with at least one node in $\mathcal{D}$. When all nodes of $\mathcal{D}$ are in $\mathcal{O}$, $\mathcal{C} = \{\mathcal{S}_{i_1}, \mathcal{S}_{i_2}, \ldots, \mathcal{S}_{i_k}\}$ is a covering set with size k in the universe $\mathcal{U}$. When some nodes of $\mathcal{D}$ are not in $\mathcal{O}$, e.g., i_a, there exists a corresponding node o_{j_a} in $\mathcal{O}$ such that i_a is linked with o_{j_a}. In this case, $\mathcal{C} = \{\mathcal{S}_{i_1}, \mathcal{S}_{i_2}, \ldots, \mathcal{S}_{j_a}, \ldots, \mathcal{S}_{i_k}\}$ is a covering set in the universe $\mathcal{U}$ with size k.

It takes polynomial time to construct the $\mathcal{G} = \{\mathcal{V}, \mathcal{E}^+, \mathcal{E}^-\}$, which indicates that the special instance of CRC can be reduced from the cover set problem in polynomial time. Therefore, the CRC problem is also NP-hard.

REFERENCES

[1] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[2] J. Ruths and D. Ruths, "Control profiles of complex networks," *Science*, vol. 343, no. 6177, pp. 1373–1376, 2014.

[3] Y.-Y. Liu and A.-L. Barabási, "Control principles of complex systems," *Reviews of Modern Physics*, vol. 88, no. 3, p. 035006, 2016.

[4] S. Wuchty, "Controllability in protein interaction networks," *Proceedings of the National Academy of Sciences*, vol. 111, no. 19, pp. 7156–7160, 2014.

[5] J. Z. Kim, J. M. Soffer, A. E. Kahn, J. M. Vettel, F. Pasqualetti, and D. S. Bassett, "Role of graph architecture in controlling dynamical networks with applications to neural systems," *Nature Physics*, vol. 14, no. 1, p. 91, 2018.

[6] C. Shi, Y. Li, J. Zhang, Y. Sun, and S. Y. Philip, "A survey of heterogeneous information network analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 1, pp. 17–37, 2017.

[7] J. Wu, X. Shen, and K. Jiao, "Game-based memetic algorithm to the vertex cover of networks," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 974–988, 2019.

[8] J. C. Nacher and T. Akutsu, "Dominating scale-free networks with variable scaling exponent: heterogeneous networks are not difficult to control," *New Journal of Physics*, vol. 14, no. 7, p. 073005, 2012.

[9] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, 1998.

[10] G. Lin, W. Zhu, and M. M. Ali, "An effective hybrid memetic algorithm for the minimum weight dominating set problem," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 892–907, 2016.

[11] M. R. Garey and D. S. Johnson, "Computers and intractability: A guide to the theory of np-completeness," *WH Freemann, New York*, 1979.

[12] A. Potluri and A. Singh, "Hybrid metaheuristic algorithms for minimum weight dominating set," *Applied Soft Computing*, vol. 13, no. 1, pp. 76–88, 2013.

[13] Y. Wang, W. Wang, and X.-Y. Li, "Efficient distributed low-cost backbone formation for wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 17, no. 7, pp. 681–693, 2006.

[14] A. Campan, T. M. Truta, and M. Beckerich, "Fast dominating set algorithms for social networks." in *Modern Artificial Intelligence and Cognitive Science*, North Carolina, USA, 2015.

[15] D. Chalupa, "An order-based algorithm for minimum dominating set with application in graph mining," *arXiv:1705.00318*, 2017.

[16] S. S. Mousavi, M. Haeri, and M. Mesbahi, "On the structural and strong structural controllability of undirected networks," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 2234–2241, 2018.

[17] R. Jovanovic and M. Tuba, "Ant colony optimization algorithm with pheromone correction strategy for the minimum connected dominating set problem," *Computer Science and Information Systems*, vol. 10, no. 1, pp. 133–149, 2013.

[18] Q. H. Nguyen, Y.-S. Ong, and M. H. Lim, "A probabilistic memetic framework," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 604–623, 2009.

[19] Y.-S. Ong, M. H. Lim, and X. Chen, "Memetic computationłpast, present & future [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 5, no. 2, pp. 24–31, 2010.

[20] X. Chen, Y.-S. Ong, M.-H. Lim, and K. C. Tan, "A multi-facet survey on memetic computation," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 5, pp. 591–607, 2011.

[21] L. Ma, M. Gong, H. Du, B. Shen, and L. Jiao, "A memetic algorithm for computing and transforming structural balance in signed networks," *Knowledge-Based Systems*, vol. 85, pp. 196–209, 2015.

[22] L. Feng, A. Gupta, and Y.-S. Ong, "Compressed representation for higher-level meme space evolution: a case study on big knapsack problems," *Memetic Computing*, vol. 11, no. 1, pp. 3–17, 2019.

[23] C. Pizzuti, "Evolutionary computation for community detection in networks: a review," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 464–483, 2018.

[24] M. Gong, C. Song, C. Duan, L. Ma, and B. Shen, "An efficient memetic algorithm for influence maximization in social networks," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 22–33, 2016.

[25] C.-P. Chen, S. C. Mukhopadhyay, C.-L. Chuang, T.-S. Lin, M.-S. Liao, Y.-C. Wang, and J.-A. Jiang, "A hybrid memetic framework for coverage optimization in wireless sensor networks," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2309–2322, 2015.

[26] A. Gupta and Y.-S. Ong, *Memetic Computation: The Mainspring of Knowledge Transfer in a Data-Driven Optimization Era*. Cham, Switzerland: Springer, 2018, vol. 21.

[27] J. Sun, Z. Miao, D. Gong, X.-J. Zeng, J. Li, and G. Wang, "Interval multiobjective optimization with memetic algorithms," *IEEE Transactions on Cybernetics*, p. doi: 10.1109/TCYB.2019.2908485, 2019.

[28] S. Wang, J. Liu, and Y. Jin, "Finding influential nodes in multiplex networks using a memetic algorithm," *IEEE Transactions on Cybernetics*, p. doi: 10.1109/TCYB.2019.2917059, 2019.

[29] Y. Zhou, J.-K. Hao, and F. Glover, "Memetic search for identifying critical nodes in sparse graphs," *IEEE Transactions on Cybernetics*, vol. 49, no. 10, pp. 3699–3712, 2019.

[30] A. Braunstein, L. DallAsta, G. Semerjian, and L. Zdeborová, "Network dismantling," *Proceedings of the National Academy of Sciences*, vol. 113, no. 4, pp. 12 368–12 373, 2016.

[31] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y.-S. Ong, K.-C. Tan, and A. Qin, "Evolutionary multitasking via explicit autoencoding," *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3457–3470, 2018.

[32] L. Feng, Y.-S. Ong, M.-H. Lim, and I. W. Tsang, "Memetic search with interdomain learning: A realization between cvrp and carp," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 644–658, 2014.

[33] W. Liu, M. Gong, S. Wang, and L. Ma, "A two-level learning strategy based memetic algorithm for enhancing community robustness of networks," *Information Sciences*, vol. 422, pp. 290–304, 2018.

[34] B. Acharya, "Domination and absorbance in signed graphs and digraphs, i: Foundations," *Journal of Combinatorial Mathematics and Combinatorial Computing*, vol. 84, pp. 5–20, 2013.

[35] P. Ashraf and K. Germina, "Double domination in signed graphs," *Cogent Mathematics*, vol. 3, no. 1, p. 1186135, 2016.

[36] M. Szell, R. Lambiotte, and S. Thurner, "Multirelational organization of large-scale social networks in an online world," *Proceedings of the National Academy of Sciences*, vol. 107, no. 31, pp. 13 636–13 641, 2010.

[37] X. Zheng, D. Zeng, and F.-Y. Wang, "Social balance in signed networks," *Information Systems Frontiers*, vol. 17, no. 5, pp. 1077–1095, 2015.

[38] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[39] D. Easley and J. Kleinberg, *Networks, crowds, and markets: Reasoning about a highly connected world*. Cambridge University Press, 2010.

[40] G. Facchetti, G. Iacono, and C. Altafini, "Computing global structural balance in large-scale signed social networks," *Proceedings of the National Academy of Sciences*, vol. 108, no. 52, pp. 20 953–20 958, 2011.

[41] F. Heider, "Attitudes and cognitive organization," *The Journal of Psychology*, vol. 21, no. 1, pp. 107–112, 1946.

[42] Z. Yuan, C. Zhao, Z. Di, W.-X. Wang, and Y.-C. Lai, "Exact controllability of complex networks," *Nature Communications*, vol. 4, p. 2447, 2013.

[43] J. Gao, Y.-Y. Liu, R. M. D'souza, and A.-L. Barabási, "Target control of complex networks," *Nature Communications*, vol. 5, p. 5415, 2014.

[44] L. Lü, T. Zhou, Q.-M. Zhang, and H. E. Stanley, "The h-index of a network node and its relation to degree and coreness," *Nature Communications*, vol. 7, p. 10168, 2016.

[45] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, 2015.

[46] M. Hifi, "A genetic algorithm-based heuristic for solving the weighted maximum independent set and some equivalent problems," *Journal of the Operational Research Society*, vol. 48, no. 6, pp. 612–622, 1997.

[47] L. Ma, M. Gong, J. Liu, Q. Cai, and L. Jiao, "Multi-level learning based memetic algorithm for community detection," *Applied Soft Computing*, vol. 19, pp. 121–133, 2014.

[48] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

[49] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008, 2008.

[50] K. E. Read, "Cultures of the central highlands, new guinea," *Southwestern Journal of Anthropology*, vol. 10, no. 1, pp. 1–43, 1954.

[51] B. Yang, W. Cheung, and J. Liu, "Community mining from signed social networks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 10, pp. 1333–1348, 2007.

[52] K. Oda, Y. Matsuoka, A. Funahashi, and H. Kitano, "A comprehensive pathway map of epidermal growth factor receptor signaling," *Molecular Systems Biology*, vol. 1, no. 1, p. 2005.0010, 2005.

[53] R. Milo *et al.*, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, p. 824C827, 2002.

[54] K. Oda, T. Kimura, Y. Matsuoka, A. Funahashi, M. Muramatsu, and H. Kitano, "Molecular interaction map of a macrophage," *AfCS Research Reports*, vol. 2, no. 14, p. 1C12, 2004.

[55] H. Salgado *et al.*, "Regulondb (version 5.0): Escherichia coli k-12 transcriptional regulatory network, operon organization, and growth conditions," *Nucleic Acids Research*, vol. 34, no. Database issue, pp. D394–D397, 2006.

[56] J. Leskovec, D. Huttenlocher, and J. Kleinberg, "Signed networks in social media," in *CHI' 10 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New York, NY, USA, Apri. 2010, pp. 1361–1370.

[57] S. S. Young, S. S. Young, and S. S. Young, *Resampling-Based Multiple Testing: Examples and Methods for P-Value Adjustment*. John Wiley & Sons, 1993.

[58] V. Chvatal, "A greedy heuristic for the set-covering problem," *Mathematics of operations research*, vol. 4, no. 3, pp. 233–235, 1979.

Lijia Ma received the B.S. degree in communication engineering from Hunan Normal University, Changsha, China, and the Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2010 and 2015, respectively.

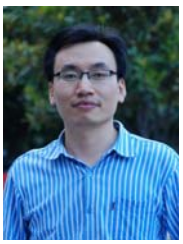
From 2015 to 2016, he was a Postdoctoral Fellow with Hong Kong Baptist University, Hong Kong, and with Nanyang Technological University, Singapore, from 2016 to 2017. He is an assistant professor at the College of Computer and Software Engineering of Shenzhen University. His research interests mainly include evolutionary computation, machine learning and complex networks.

Jianqiang Li received his B.S. and Ph.D. Degree in automation major from from South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively.

He is a professor at the College of Computer and Software Engineering of Shenzhen University. He led a project of the National Natural Science Foundation, and a project of the Natural Science Foundation of Guangdong Province, China. His major research interests include embedded systems and Internet of Things.

Qiuzhen Lin (M'19) received the B.S. degree from Zhaoqing University and the M.S. degree from Shenzhen University, China, in 2007 and 2010, respectively. He received the Ph.D. degree from Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, in 2014.

He is currently an associate professor in College of Computer Science and Software Engineering, Shenzhen University. His current research interests include artificial immune system, multi-objective optimization and dynamic system.



Maoguo Gong (M'07-SM'14) received the B.S. degree and Ph.D. degree in electronic science and technology from Xidian University, Xi'an, China, in 2003 and 2009, respectively.

Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted as an Associate Professor and as a Full Professor, respectively, both with exceptive admission. His research interests are in the area of computational intelligence with applications to optimization, learning, data mining and image understanding.

Dr. Gong received the prestigious National Program for the support of Top-Notch Young Professionals from the Central Organization Department of China, the Excellent Young Scientist Foundation from the National Natural Science Foundation of China, and the New Century Excellent Talent in University from the Ministry of Education of China. He is the Vice Chair of the IEEE Computational Intelligence Society Task Force on Memetic Computing, an Executive Committee Member of the Chinese Association for Artificial Intelligence, and a Senior Member of the Chinese Computer Federation. He is also the associate editor of *IEEE Trans. Evolutionary Computation*.

Carlos A. Coello Coello (M'98-SM'04-F'11) received PhD degree in computer science from Tulane University, USA, in 1996. He is currently Professor (CINVESTAV-3F Researcher) at the Computer Science Department of CINVESTAV-IPN, in Mexico City, Mxico. Dr. Coello has authored and co-authored over 450 technical papers and book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Second Edition, Springer, 2007). His publications report over 45,000 citations in Google Scholar (his h-index is 83). Currently, he is associate editor of the *IEEE Transactions on Evolutionary Computation* and serves in the editorial board of 12 other international journals. His major research interests are: evolutionary multi-objective optimization and constraint-handling techniques for evolutionary algorithms. He received the *2007 National Research Award* from the Mexican Academy of Sciences in the area of Exact Sciences, the *2013 IEEE Kiyo Tomiyasu Award* and the *2012 National Medal of Science and Arts* in the area of *Physical, Mathematical and Natural Sciences*. He is a Fellow of the IEEE, and a member of the ACM, Sigma Xi, and the Mexican Academy of Science.

Zhong Ming received the B.S. degree in computer science from Nanchang University, Nanchang, China, and the M.S. and Ph.D. degrees in computer software and theory from Sun Yat-sen University, Guangzhou, China, in 2000 and 2004, respectively.

Zhong Ming is a professor at College of Computer and Software Engineering of Shenzhen University. He is a member of a council and senior member of China Computer Federation. His major research interests are AI and cloud computing. He led two projects of National Natural Science Foundation.