Community Detection Based on Symbiotic Organisms Search and Neighborhood Information

Jing Xiao, Chao Wang, and Xiao-Ke Xu, *Member, IEEE*

***Abstract*— Modularity optimization methods based on nature-inspired metaheuristics are popular and competent for community detection in complex networks. However, on some real-world networks with complex and vague structures, most contemporary algorithms are difficult to obtain the global optimal partition. There are two key factors that are seriously affecting the global optimization capability: one is the convergence performance of the incorporated optimization strategy and the other is the sufficient and rational utilization of network topological information. In this article, a novel community detection method is proposed, named symbiotic organisms search community detection (SOSCD). The bio-inspired metaheuristic [symbiotic organisms search (SOS)] is discretized and utilized as the optimization strategy to improve global convergence performance of modularity optimization. Meanwhile, by utilizing neighborhood information of each node to guide community optimization, two different local search (LS) schemes are designed to intensify exploitation and, thus, assisting the global search, including the neighbor-based community modification (NCM) and the neighbor-based LS (NLS). Experimental results on both synthetic and real-world networks have validated the effectiveness and superiority of the proposed operations in SOSCD. Moreover, SOSCD can significantly improve the precision and stability of the identified optimal partition, comparing with many state-of-the-art modularity optimization algorithms.**

***Index Terms*— Community detection, local search (LS), modularity optimization, nature-inspired computation, neighborhood information, symbiotic organism search.**

I. Introduction

COMMUNITY structure is one of the most important characteristics of complex networks that are commonly found in various kinds of modern complex systems [1], [2]. Discovering community structures is a basic and important technology in the area of complex network analysis, which not only enables us to gain insights into network topological features but also helps us to unearth in-depth information about networks, such as functional properties and dynamic characteristics [3], [4].

Manuscript received April 21, 2019; revised July 30, 2019; accepted September 11, 2019. Date of publication October 1, 2019; date of current version December 9, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61603073 and Grant 61773091, in part by the Key Research and Development Plan of Liaoning Province under Grant 2018104016, in part by the Liaoning Revitalization Talents Program under Grant XLYC1807106, and in part by the Program for the Outstanding Innovative Teams of Higher Learning Institutions of Liaoning under Grant LR2016070. *(Corresponding author: Xiao-Ke Xu.)*

J. Xiao and X.-K. Xu are with the College of Information and Communication Engineering, Dalian Minzu University, Dalian 116600, China (e-mail: hrbeuxiaojing@aliyun.com; xuxiaoke@foxmail.com).

C. Wang is with the School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: wangchao8@ahu.edu.cn).

Digital Object Identifier 10.1109/TCSS.2019.2941988

In recent years, we have experienced an increasing demand for the technology of community detection from a wide range of science domains [5]. At present, community detection has been intensively studied, and a lot of approaches based on different knowledge backgrounds have been proposed [4], [6], [7]. However, up to now, it is still a difficult task for available methods to get accurate and stable optimal partitions from many real-world networks, especially those with complicated and fuzzy community structures [1], [5], [8].

Mathematically, community detection can be essentially viewed as a global optimization problem [1]–[4], which usually takes specific criterion function well-reflecting the concept of community as the objective function, such as modularity [9]. Such kind of optimization problems has been proved to be a typical nondeterministic polynomial (NP)-hard problem [1]–[5], [8]. In the last decade, many heuristic optimization methods have been proposed to find the global optimal division of a network, where the population-based metaheuristics are the most competitive and widely utilized methods. Typical representatives mainly include evolutionary algorithms (EAs), in particular the genetic algorithm (GA) [10]–[15], the differential evolution (DE) [3], [16]–[19], and the estimation distribution algorithm (EDA) [20]. In addition, a relevant number of nature-inspired metaheuristics, such as the particle swarm optimization (PSO) [21]–[23], the ant colony optimization (ACO) [24]–[27], the fruit fly optimization algorithm (FOA) [28], the bat algorithm (BA) [29], [30], the fire spread algorithm (FSA) [5], and the state transition algorithm (STA) [8], have also been employed.

Because of the inherent features of parallel computation and self-organization, population-based metaheuristics present a number of advantages, such as higher accuracy and automatically determined community numbers [1]–[4], [6], [7]. However, due to complex and vague structures in some real-world networks, existing algorithms cannot obtain stable results and often trap in local optimum [1], [4], [5], [8]. Actually, the global optimization performance in dealing with community detection can be improved from two main aspects. One is to strengthen the global convergence ability of the incorporated optimization strategy. The other is to utilize network topological information (e.g., neighborhood information of nodes) efficiently and rationally to assist the global search by intensifying exploitation.

Up to now, most of the existing approaches only focus on incorporating EAs or nature-inspired metaheuristics into the framework of modularity optimization for community detection [1]–[4]. For example, individual encoding schemes and variation operators are redesigned in a discrete form to adapt to the discrete scenario of community optimization [5], [8], [10], [16], [20], [31], whereas domain-specific knowledge such as known topological information of real-life networks, in particular neighborhood information of each node, has not been fully utilized.

Based on the above-mentioned analysis, a novel community detection method based on the framework of modularity optimization is proposed in this article, named symbiotic organisms search community detection (SOSCD). The major improvements and innovations of SOSCD include the following two points.

First, a novel bio-inspired metaheuristic method, called symbiotic organisms search (SOS) [32], is employed to serve as the optimization strategy. Since the standard SOS algorithm has been proved to be more accurate and stable than many well-known metaheuristics (including the standard GA, DE, PSO, and BA), in solving various complex numerical optimization problems [32]–[34], its good convergence performance in the modularity optimization could be expected. However, the original SOS algorithm is designed for solving continuous optimization problems and cannot be applied in the discrete community detection problem directly. Therefore, we proposed a discrete SOS for modularity optimization, adopting the widely used label-based representation [1]–[3], [11] as the encoding scheme which generates integer vectors as individuals to represent feasible community partitions of a network.

Second, network topological information (i.e., node neighborhood information) is also sufficiently used in SOSCD, including the initialization and two newly designed local search (LS) strategies, named neighbor-based community modification (NCM) and neighbor-based LS (NLS), respectively. In NCM, neighborhood information is mainly used to correct the wrong community membership of nodes and improve the quality of community divisions. NCM can efficiently enhance the exploitation ability of SOSCD and avoid falling into local optimum at the same time. Moreover, NLS helps high quality but suboptimal individuals escape from local optimum and increase the probability of finding the global optimal partition. It utilizes node neighborhood information to avoid random invalid search normally found in traditional LS approaches and reduces computational complexity.

In the proposed SOSCD, both a discrete bio-inspired method and network topological information are utilized as the crucial factors to enhance its global optimization performance. To verify the effectiveness and superiority of the SOS-based optimization strategy and our two newly designed LS methods, both synthetic and real-world networks are employed in experiments. Superiority of SOSCD in community detection is further validated through comparison with many other well-known modularity optimization algorithms. Moreover, SOSCD works without prior knowledge of the total number of communities and only requires a few specific parameters for optimization.

This article is organized as follows. Section II introduces definitions and concepts related to the optimization problem in community detection. Section III describes the designing ideas, procedures, and detailed operations of the proposed community detection algorithm SOSCD. Section IV evaluates the core operations in SOSCD. Section V compares the performance of SOSCD against many typical state-of-the-art modularity optimization algorithms. In Section VI, we conclude this article.

II. Mathematical Model of Optimization Problem in Community Detection

In community detection, a network is often modeled as a graph $G = (V, E)$, where V represents the set of vertices and E is the set of edges, respectively. Generally, topological information in G can be represented by an adjacency matrix A, in which each element A_{ij} determines the kind of G and whether there is an edge between vertices i and j [1]. In this article, we merely focus on the kind of undirected networks with $A_{ij} \in \{0, 1\}$ and $A_{ij} = A_{ji}$. The community structure is often defined as a partition $C = \{c_1, \cdots, c_k\}$ with k divided subgraphs, such that $V = \bigcup_{i=1}^{k} c_i$, where the number of k is often unknown in real-world networks. For nonoverlapping community structures, a node can only be allowed to participate in one community, such that $c_i \cap c_j = \varnothing, \forall i, j$.

Community detection can be modeled mathematically as a single-objective global optimization problem [1], [3], [4]. All the feasible community partitions of graph G construct the search space $\Omega = \{C_1, \cdots, C_r\}$. The goal of the optimization problem is to find the global optimal partition C^* in Ω, for which

$$F(C^*) = \max(F(C)) \quad \text{s.t. } C \in \Omega. \tag{1}$$

In (1), $F : \Omega \to R$ is the fitness function used to evaluate the quality of the candidate partition C, which also maps the solution space into the objective function space. As shown in (1), the optimal partition C^* possesses the global optimal value of F.

The selection of fitness function F is crucial for the performance of community detection algorithms. In this article, we use the most popular fitness function (i.e., modularity) [1]–[4], [9], defined as

$$Q = \frac{1}{2M} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2M} \right) \delta(i, j) \tag{2}$$

where M is the number of edges and k_i and k_j are the degrees of nodes i and j, respectively. Here, δ indicates the community relationship between nodes i and j. Through Q suffers from the so-called resolution limit problem [1]–[4], [9] and does not take into account the important feature of community size [35], it remains the most competitive and widely used criterion function to optimize. In the optimization, we seek for the global maximum of Q, the higher the value of Q, the better the quality of the candidate partition C.

III. Proposed Community Detection Algorithm

In this section, we will introduce our proposed community detection algorithm SOSCD in detail. As it was mentioned earlier, the global optimization performance of community detection algorithms can be improved from two main aspects: enhancing the global convergence ability of the incorporated

Algorithm 1 SOSCD

Input: The Original Network.
Output: Optimal Community Partition.
1: Encoding and Initialization
2: REPEAT
3: – Mutualism operation
4: – Commensalism operation
5: – Parasitism operation
6: – Neighbor-based Community Modification (NCM)
7: – Neighbor-based Local Search (NLS)
8: UNTIL(termination criterion is met)
9: Output the Current Best Result

optimization strategy and utilizing network topological information to assist the global search by strengthening exploitation. Especially, the two aspects are both fully considered and are utilized as equally important factors to improve the global optimization performance of SOSCD, thus improving the quality of detection results.

The algorithm framework of SOSCD is presented in Algorithm 1. For one thing, SOSCD utilizes high-performance standard SOS to serve as the optimization strategy for modularity optimization, promoting the global convergence in the huge search space. As far as we have known, the standard SOS algorithm is employed for the first time in solving the problem of community detection. For another, topological information of networks (e.g., node neighborhood) is also fully utilized in crucial components of SOSCD, especially in the two newly designed LS operations.

As we can see from Algorithm 1, NCM and NLS are executed after the evolution of SOS in the main loop, assisting the global search of SOSCD by strengthening exploitation. NCM is used to improve the community membership of nodes in all the individuals, and NLS is designed for helping a few high-quality but suboptimal individuals escape from a local optimum. In both operations, neighborhood information is used directly to improve the individual quality by modifying the community membership of nodes. Thus, the population can converge faster to the promising region in the search space, and the exploitation ability of the algorithm can be enhanced. In the following, we provide more details of core operations in SOSCD, mainly including the SOS-based modularity optimization and LS.

A. Symbiotic Organisms Search

SOS is a novel bio-inspired metaheuristic method that simulates the symbiotic interaction and survival behaviors of organisms in modern ecosystems [32]–[34]. In the algorithm, a randomly generated initial population is used to simulate the real ecosystem, and each individual is considered to be a unique organism. The fitness value of each individual represents the degree of adaptability of each organism to the ecosystem. SOS searches the global optimum by iteratively updating the population according to three special variation operations, including mutualism, commensalism, and parasitism. The operations resemble three kinds of the most common symbiotic relationships between any two organisms in all the modern ecosystems [32]–[34].

Specially, SOS does not require any specific parameters [32]. Besides, compared with several well-known metaheuristics (i.e., the standard GA, PSO and DE algorithms), SOS can obtain better optimization results in a wide variety of benchmark problems. Recently, it has been applied to solve many real-world complex optimization problems, such as electromagnetic optimization [36], capacitated vehicle routing [37], and the distribution feeder reconfiguration in a vehicle-to-grid system [38]. Due to the advantages of accurate, robust, and easy to be implemented, SOS provides an ideal optimization strategy for modularity optimization in community detection.

Considering the fact that community detection is a discrete integer optimization problem, in order to make the original SOS algorithm usable in this discrete context, we adopt the widely used label-based representation [1]–[3], [11] as the encoding scheme in initialization and generate n-dimensional integer vectors as population individuals to represent feasible community partitions of a network. Here, n is the number of nodes. The three crucial variation operations in the standard SOS are all performed on discrete individuals. Besides, after each variation operation, each dimension of individuals is rounded to the nearest integer and restricted within the predefined boundary (i.e. $[1,n]$), guaranteeing the legality of individuals.

1) Mutualism Operation: Mutualism is a typical symbiotic interaction relationship, in which both the two participate organisms can be benefited through enhancing survival advantages in an ecosystem. In SOS, mutualism is the first variation operation, in which both the interacting individuals are mutated and updated concurrently by utilizing their average information. Concrete steps in mutualism are as follows. Each individual in the current population is selected, in turn, as the target individual for variation. Suppose X_i is the current target individual corresponding to the ith organism in the ecosystem. Another individual X_j is randomly selected from population for interacting with X_i. Then, two new offspring individuals are generated according to the mutualistic relation modeled in

$$X_{i\text{new}} = X_i + \text{rand}(0,1) \cdot (X_{\text{best}} - \text{Mutual\_Vector} \cdot \text{BF}_1) \quad (3)$$

$$X_{j\text{new}} = X_j + \text{rand}(0,1) \cdot (X_{\text{best}} - \text{Mutual\_Vector} \cdot \text{BF}_2) \quad (4)$$

$$\text{Mutual\_Vector} = 0.5 \cdot (X_i + X_j) \quad (5)$$

where $\text{rand}(0,1)$ denotes a vector of random numbers with each element valued between 0 and 1. X_{best} corresponds to the current best individual in the population. BF_1 and BF_2 are the benefit factors determined randomly as either 1 or 2, representing partially or fully benefit from the interaction, respectively. Mutual_Vector is the mean value of X_i and X_j, representing the mutualistic relationship characteristic between the two organisms. $X_{\text{best}} - \text{Mutual\_Vector} \cdot \text{BF}_1$ and $X_{\text{best}} - \text{Mutual\_Vector} \cdot \text{BF}_2$ reflect the mutualistic effort to increase the survival probability in the ecosystem, provided by the two organisms, respectively. After calculation, two new offspring individuals $X_{i\text{new}}$ and $X_{j\text{new}}$ are generated, which will replace X_i and X_j in the population, respectively, if they have better fitness values.

Intuitively, mutualism seems like the target − to − best/1 mutation strategy of the DE algorithm [19], both of which can maintain a good balance between exploration and exploitation. In (3) and (4), random interactive information in Mutual_Vector is helpful to keep population diversity, whereas the current best individual X_{best} is used to guide the variation direction for global search. Moreover, mutualism updates two parent individuals simultaneously in a single operation and performs one-to-one elite selection right after the operation, which makes it more efficient and greedy than the mutation in the standard DE.

2) Commensalism Operation: Commensalism is another typical symbiotic interaction relationship in an ecosystem. Different from mutualism relationship, there is only one organism may benefit from the interaction, and at the same time, the other participant is unaffected or neutral. In SOS, commensalism is performed after the mutualism, in which the target individual can be updated by utilizing the information brought from the interacting participant. Concrete steps in commensalism are as follows. Each individual in the current population is selected, in turn, as the target individual for variation. Suppose X_i is the current target individual; we also need to randomly select an individual X_j from the population for interaction. Only the target individual X_i may be benefited, and its offspring individual $X_{i\text{new}}$ is generated utilizing the information introduced by X_j. Commensalism relation is modeled in

$$X_{i\text{new}} = X_i + \text{rand}(-1, 1) \cdot (X_{\text{best}} - X_j) \tag{6}$$

where $\text{rand}(-1, 1)$ denotes a vector of random numbers with each element valued between −1 and 1. The current best individual X_{best} represents the highest level of survival adaptation in the current organism. Thus, $X_{\text{best}} - X_j$ reflects the beneficial advantage introduced by X_j, which aims to help the target individual increase its survival advantage in an ecosystem. After calculation, new offspring individual $X_{i\text{new}}$ is generated, which can only be accepted and reserved if its fitness value is better than its father individual X_i.

It can be observed from (6) that commensalism is also similar to the mutation strategy of DE/target-to-best/1 [19]. However, the differential vector $X_{\text{best}} - X_j$ puts more emphasis on the influence brought by the other individual X_j in the population, rather than the mutual interaction between X_i and X_j. In addition, the influence of X_i on the evolution of X_j is not considered in commensalism. Similarly, the one-to-one elite selection is conducted directly after the commensalism to update the population in real time.

3) Parasitism Operation: Distinct from mutualism and commensalism relationship, parasitism depicts the parasitic and game relationship between any pair of parasite and host. In parasitism, parasite and host compete for survival in the ecosystem. Only one of them with a higher level of adaptation can be able to live, while the other participant will be killed and eliminated from the ecosystem, rather than unaffected.

In SOS, parasitism is implemented after the mutualism and commensalism, which is the last but the most special one among the three variation operations. Concrete operations in parasitism are as follows. Each individual in the current population is selected, in turn, as the target individual for variation. Suppose X_i is the current target individual corresponding to the ith organism in the ecosystem. Based on the target individual, a special Parasite_Vector that serves as the parasite is created by randomly selecting and modifying a set of elements in X_i. Meanwhile, another individual X_j that is given a role as the host is randomly selected from the current population. Then, Parasite_Vector competes with X_j that represents the game between the parasite and the host. Only the superior individual with the higher fitness value can survive and participate in the next generation of evolution, while the other individual will be eliminated from the population.

It can be seen from the above-mentioned description, parasitism works as an LS operation for each individual in the last step of the standard SOS algorithm, which is particularly important for global convergence in the later evolutionary stage. Parasitism is similar to the multipoint mutation operation in GA. In both parasitism and mutation, each individual can promote itself evolve by performing a local genetic variation, rather than relying on interacting with other individuals. However, unlike genetic variation, the dimension of randomly selected elements for variation in each individual ranges from 1 to n (n is the length of the individual) in parasitism. Therefore, the newly generated Parasite_Vector may be located around X_i in the search space or in a completely different area [32]. In addition, the fitness value of Parasite_Vector may not be better than that of X_i, but Parasite_Vector can also be preserved as long as its fitness is higher than that of another randomly selected individual X_j. To sum up, the highly random nature of parasitism helps SOS maintain population diversity and prevent premature convergence, especially in the last stage of evolution.

B. Local Search Based on Neighborhood Information

Though the high performance of the bio-inspired SOS helps to improve the global convergence ability in its application of community detection, the variation operations in the standard SOS often generate unreasonable divisions with disconnected nodes be assigned to a same community. In order to improve the quality of the community divisions, two new LS strategies (NCM and NLS) are designed by utilizing node neighborhood information of networks. Both of them are incorporated in SOSCD to further enhance the exploitation ability and the probability of finding the global optimal division, with the consideration that neighbor vertices are more likely to locate in a same community.

1) Neighbor-Based Community Modification: The first LS strategy is called NCM, which is mainly used to correct the unreasonable divisions and enhance their quality.

NCM makes a rough LS on all the population individuals. On each individual, a certain number of nodes are randomly selected and each node will be checked whether it is significantly different from its neighbors in terms of community membership. Once the difference is great, the node is deemed to be wrongly partitioned with high probability. Therefore, it will be modified into a new community that is selected

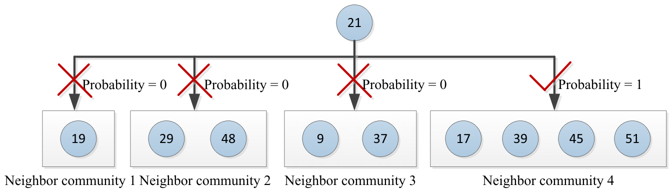

Fig. 1. Clean-up operation on the 21st node of the Dolphin network.

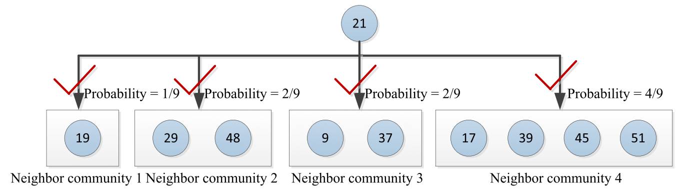

Fig. 2. NCM operation on the 21st node of the Dolphin network.

from the neighbor communities (communities that its neighbor nodes belong to). Specially, all the neighbor communities can be chosen with probability proportional to the community scale, rather than only the largest one.

The newly designed NCM is similar to the clean-up operation mentioned in [16] but with more relaxed restriction in the process of community modification (CM). In cleanup, nodes can only be assigned to the largest-scale neighbor community [2]. Unfortunately, sometimes, only when a node is moved to other smaller size neighbor communities, we can get the global optimal partition with the largest value of modularity [19]. It does mean that excessive restriction has been imposed during the modification, so the search space containing the global optimum may probably be omitted. In other words, the clean-up operation may make an individual be trapped into a local optimum and it is extremely difficult to achieve the global optimum. NCM operation successfully avoids this problem by adopting a more reasonable manner in utilizing neighborhood information of each node. Moreover, NCM strategy can also be transplanted to other bio-inspired or EA-based metaheuristics, which is suitable to be conducted at the early stage of evolution.

To gain a better understanding of the operation, we take the Dolphin network for example. In a typical suboptimal community partition of the Dolphin network, the 21st node has nine neighbors located in four different-sized communities. When we modify the community membership of this node according to the clean-up operation, it can only be moved to the neighbor community with the largest scale (i.e., neighbor community 4), as shown in Fig. 1. Unfortunately, the above-mentioned modification cannot enable the individual to achieve the global optimum. Only when the 21st node is transferred to another smaller sized neighbor community $\{29, 48\}$ (neighbor community 2), we can get the global optimal partition with the largest value of modularity. This makes us realize that excessive restriction has been imposed during the modification, and the search space containing the global best partition is missed. In other words, the individual is trapped into a local optimum and it is difficult to jump out.

To avoid this problem, we relax the restriction imposed on nodes during the modification. For any node whose community membership needs to be modified, all of its neighbor communities can be selected with the probability proportional to the community scale, rather than only the largest one. In the example of the Dolphin network, the 21st node can be modified in the manner shown in Fig. 2. In this way, we can utilize the neighborhood information of nodes more reasonably, and the search space containing the global optimal partition will not be missed.

2) Neighbor-Based Local Search: Apart from the NCM strategy, another LS strategy, called NLS, is also designed. Different from NCM, NLS conducts a more-refined and accurate search around a few high-quality but suboptimal individuals. It helps them jump out of local optimum through minor changes, especially at the final stage of evolution, increasing the probability of finding out the global optimum.

At the last stage of SOSCD, a large amount of high-quality similar but suboptimal partitions can usually be witnessed in the population. Actually, the difference between them and the global optimum often only lies in the community membership of very few nodes (e.g., one or two for a small-scale network). However, as the community size increases and the number of communities decreasing at the same time, the population diversity becomes relatively low. It is very difficult for these high-quality partitions to be further adjusted to achieve the global optimum, even adopting NCM.

NLS operation can contribute to overcome this problem, and the concrete operations are illustrated in Algorithm 2. For a target individual X, the community membership of each node represented by community identifier (commID) is locally optimized successively by utilizing the information of neighbor nodes. First, for each node i in X, all of its neighbor nodes as well as their commIDs are identified. Second, the unique

Algorithm 2 NLS

Input: Target individual: X; Modularity of X : Q_X.
Output: Local optimal partition of X: $X_{localbest}$; Modularity of $X_{localbest}$: $Q\_X_{localbest}$.

for $i = 1 : n$ **do**
  $commID(i) = x(i)$
  $neighbors(i) = \{j | (i, j) \in E\}$
  $commID\_neighbors(i) = \{commID(j) | j \in neighbors(i)\}$
  $commID\_candidate(i) = unique(setdiff(commID\_neighbors(i), commID\_i))$
  for $j = 1 : length(commID\_candidate(i))$ **do**
    $X_{new} = X$
    $X_{new}(i) = commID\_candidate(i)(j)$
    if $Q\_X_{new} > Q\_X$ **then**
      $X = X_{new}$
      $Q\_X = Q\_X_{new}$
    end if
  end for
end for
$X_{localbest} = X$
$Q\_X_{localbest} = Q\_X$

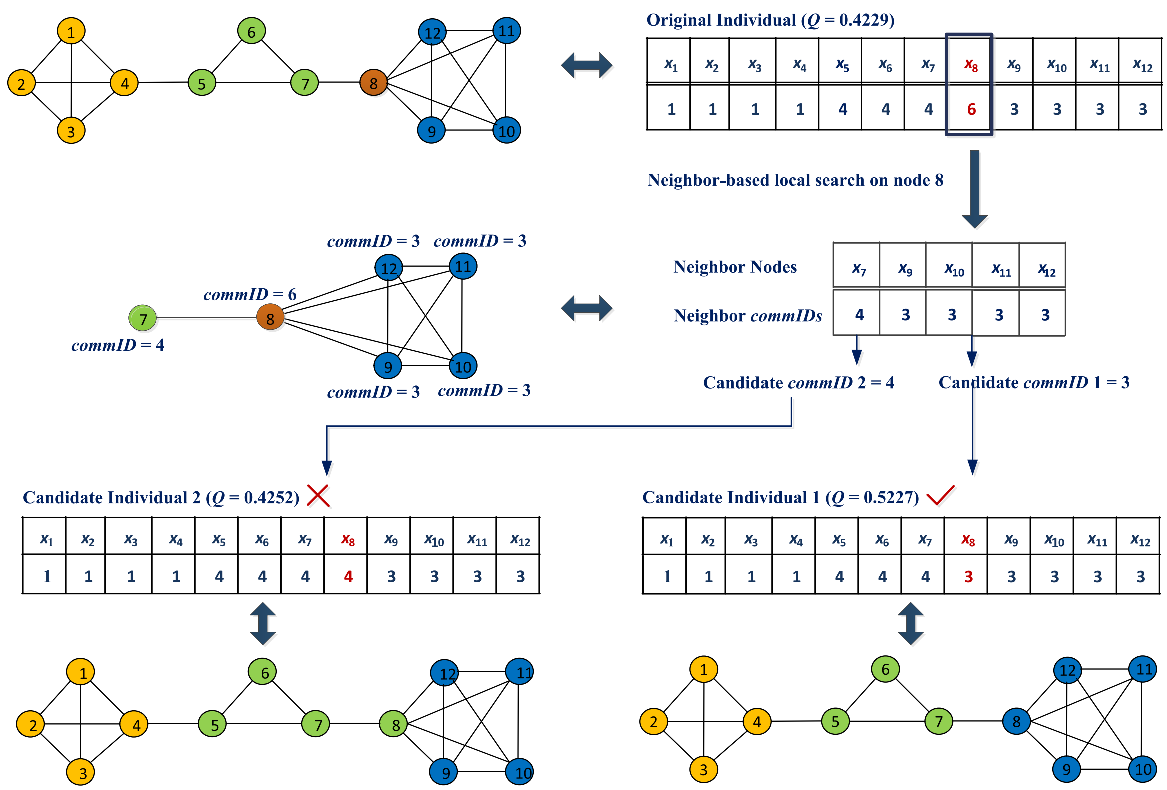

Fig. 3. Detailed illustration of the NLS operation on a toy network.

neighbor commIDs, which are different from the commID of node i, are selected as candidates for node i to choose. Third, replace the original commID of node i with each element in the candidates to generate a set of new neighbor partitions of X. Here, the best neighbor partition will compare with X and replace X if it has a better value of modularity (Q). After the above-mentioned three steps conducted on each node in X, the LS for X is accomplished. The final best neighbor partition saved in X is considered to be the local optimal partition $X_{\text{localbest}}$. In each generation of SOSCD, the above-mentioned operation is imposed on half of the top individuals, and thus, the promising areas around historical good individuals can be carefully detected and the population can be updated toward a better direction greedily.

The proposed NLS operation is similar to the hill-climbing method in [2], [28], and [39] but with stronger directional guidance brought from the useful information of neighbor nodes. It generates neighbor partitions by replacing the community membership of a node with only the communities containing its neighbor nodes, rather than all the other communities in the original partition. For a partition $C = \{c_1, \cdots, c_k\}$, suppose there are k ($k < n$) different communities, and then, the time complexity of NLS can be reduced to $O(n * k_d)$ from $O(n * k)$ of the hill-climbing. Here, k_d denotes the average number of different communities containing d neighbors, which is usually much smaller than k.

Fig. 3 shows a detailed illustration of NLS on a toy network that has 12 nodes and three communities. In a typical suboptimal individual acquired during the evolution, node x_8 is wrongly arranged in an isolated community with commID equals to 6, and thus, the individual needs to be further optimized. When performing the LS operation on node x_8, all of its neighbor nodes are checked. We find that there are five neighbors $\{7, 9, 10, 11, 12\}$ of node x_8, where $\{9, 10, 11, 12\}$ are in the same community with commID equal to 3 and node $\{7\}$ locates in another community with commID equal to 4. Because the two neighbor communities are different with the one that node x_8 belongs to, both of them can be selected as candidate communities. By replacing the current community of node x_8 with the candidates, respectively, two new candidate individuals can be generated. As shown in Fig. 3, two new candidate individuals are evaluated and compared with the original individual, and only the best one with the largest value of Q ($Q = 0.5227$) can be reserved in the population. After performing the NLS operation, this suboptimal individual can achieve the global optimal division of the toy network.

C. Algorithm Framework and Complexity Analysis

The detailed flowchart of the algorithm is described as follows. First, in order to adapt to the discrete optimization scenario in community detection, SOSCD adopts the most popular encoding method, named label-based representation [16], [17], [19], [28], to construct population individuals. In the representation, an individual X is represented by a label-based integer string $X = \{x_1, x_2, \cdots, x_i, \cdots, x_n\}$ and $i \in [1, n]$, which serves as a candidate community partition, where n is the number of nodes. Each element x_i indicates the commID of node i. A community structure C can be identified automatically from X without decoding operation. Based on the label-based representation, the population in SOSCD is initialized by constructing *NP* individuals. To avoid generating lots of unreasonable partitions, a biased process utilized in [16] is introduced to improve the quality of individuals after initialization. The operation can help to increase the probability of neighbor nodes to locate in a same community. SOSCD only needs three initial parameters, including the population size *NP*, the maximal number of iterations $t_{\max}$, and the threshold value δ for CM. Second, SOSCD enters the main loop to update the population iteratively for convergence in the search space. The main loop consists of three variation operations in the standard SOS and two newly designed LS strategies. The variation operations in SOS include mutualism, commensalism, and parasitism, and each operation is performed on every individual in the population. Detailed description can be found in Section III-A. After variation, LS operations (NCM and NLS) described in Section III-B are implemented on all and half the number of individuals with higher fitness, respectively, further enhancing the exploitation ability of SOSCD for global convergence. The main loop stops when some predefined terminal conditions are satisfied. At last, the current best individual in the population is the output as the final optimal community partition.

In SOSCD, the computational cost mainly relies on the operations in the main loop. Since the time complexity of the operations (3–7) in Algorithm 1 are $O(3n)$, $O(n)$, $O(n)$, $O(nd)$, and $O(n * k_d)$, respectively, the total time complexity of SOSCD can be simplified as $O(t_{\max} * NP * n * d)$, where n is the number of nodes and d defines the average number of neighbors for each node. The time complexity of SOSCD is a little lower than its most competitive contrast algorithm CDEMO that is proposed in our previous study [19].

IV. Evaluation of Core Operations in SOSCD

This section provides a detailed demonstration of the effectiveness of core operations to improve the global convergence performance in SOSCD. The proposed algorithm SOSCD is implemented in MATLAB 7 (2016b), and the experiments are executed on a computer with Intel Core 2.2-GHz CPU and 8-GB RAM.

A. Benchmark Networks

In our experiments, both the synthetic and real-world benchmarks networks are adopted, all of which are well-studied in the literature [16]–[20], [28].

1) GN Networks: GN network is the most frequently used synthetic benchmark network in community detection. GN network, introduced by Girvan and Newman [40], is the most frequently used synthetic benchmark network in community detection [8], [43]. In each GN benchmark network, there are 128 nodes divided into four communities of 32 nodes each. Each node has z_{in} edges connecting nodes within the same community and z_{out} edges connecting nodes in other communities. Besides, the total expected degree of each node is $z_{\text{in}}+z_{\text{out}} = 16$. In our experiments, ten different kinds of GN networks are designed for test, with the community structures becoming much vaguer with parameter z_{out} ranging from 0 to 9 with a span of 1.0. Community detection becomes difficult on the condition of $z_{\text{out}} > z_{\text{in}}$, so an algorithm that can find out real communities is considered to be stable enough.

2) LFR Networks: LFR network is another well-known synthetic benchmark network [43], whose community structures are much closer to real-world networks. For LFR benchmark networks, the distribution of degrees and community sizes are both power laws with tunable exponents. The mixing parameter μ determines the fuzzy degree of communities, and a larger value of μ indicates vaguer community structures. In the experiments, eight different kinds of LFR networks containing 1000 nodes are also designed for the test, where parameter μ is ranged from 0 to 0.7 with a span of 0.1. Besides, the community size and average degree are set in the range of [10, 50] and [20, 50], respectively.

3) Real-World Networks: Apart from synthetic benchmark networks, many well-known real-world networks are also frequently used in studies. Table I summarizes a list of typical real-world networks along with their unique topological information [3], [23], [40]–[42], [44].

TABLE I
Typical Real-World Networks. N Represents the Number of Nodes, and M Represents the Number of Edges. Abbreviations in Brackets Are Quoted in Table II to Refer to the Corresponding Networks

Networks	N	M
Zachary Karate Club (ZKC)	34	78
Bottlenose Dolphins (BD)	62	159
Kreb Political Books (KPB)	105	441
American College Football (ACF)	115	613
Santa Fe Institute (SFI)	118	200
Jazz Musicians (JM)	198	2742
Celegans Neural (CN)	453	2040
E-mail Communication (EMC)	1133	5451
Yeast Protein-Protein Interaction (PPI)	1430	6531
NetScience (NS)	1589	2742
Facebook	2888	2981
Power Grid (PG)	4941	6594
Scientific Collaboration (SC)	5242	14496
Erdos (Erdos)	6927	11850
PGP	10680	24340

B. Performance Metrics

Up to now, there is no standard performance metric that can be used to evaluate the quality of the community structures detected in complex networks [28]. From the perspective of experiments, modularity (Q) and normalized mutual information (NMI) are the two most widely used metrics at present [16]–[20], [28].

1) Modularity: Modularity (Q) defined in (2) is the most widely used performance metric for community detection in existing studies. It mainly measures the significant level of the community structure detected from real-world networks whose real communities are usually unknown.

2) Normalized Mutual Information: NMI is defined as

$$\text{NMI}(A, B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log\left(\frac{C_{ij}N}{C_iC_j}\right)}{\sum_{i=1}^{C_A} C_i\log\left(\frac{C_i}{N}\right) + \sum_{j=1}^{C_B} C_j\log\left(\frac{C_j}{N}\right)}. \quad (7)$$

TABLE II

TYPICAL OPTIMIZATION ALGORITHMS FOR COMMUNITY DETECTION BASED ON EAs AND NATURE-INSPIRED METHODS. ABBREVIATIONS IN THE COLUMN OF "TEST NETWORKS" ARE EXPLAINED IN DETAIL IN TABLE I

Strategies	Algorithms	Objective functions	Test networks	References
GA	GATHB	Modularity	ZKC, ACF, EMC	Tasgin et al. 2007 [11]
	GA-Net	Community Score	GN, BD, KPB, ACF	Pizzuti. 2008 [10]
	LGA	Modularity	GN, BD, KPB, ACF	Jin et al. 2011 [12]
	Meme-Net	Modularity Density	GN, ZKC, BD, KPB, ACF	Gong et al. 2011 [39]
	MOGA-Net	Community Score and Community fitness	GN, ZKC, BD, KPB, ACF	Pizzuti et al. 2012 [15]
	MOEA/D-Net	Negative Ratio Association and Ratio Cut	GN, ZKC, BD, KPB, ACF	Gong et al. 2012 [51]
	ECGA	Modularity	GN, LFR, ZKC, BD, KPB, ACF, CN, JM	Li et al. 2013 [13]
	MAGA-Net	Modularity	LFR, ZKC, BD, KPB, ACF	Li and Liu. 2016 [14]
	GGA+	Modularity	KPB, ACF	Guerrero et al. 2017 [52]
	CC-GA	Modularity	GR, LFR, BA, FF, WS, ZKC, BD, KPB, ACF, JM, NS, Facebook, SC, PGP	Said et al. 2018 [53]
PSO	MODPSO	Kernel K-Means and Ratio Cut	GN, LFR, ZKC, BD, ACF, SFI, NS, PG	Gong et al. 2014 [23]
	GDPSO	Modularity	GN, ZKC, BD, ACF, SFI, EMC, NS, PG, PGP	Cai et al. 2015 [21]
	Discrete PSO	Modularity Density	GN, LFR, ZKC, BD, KPB, ACF, SFI, NS	Zhou et al. 2016 [22]
	MPSOA	Modularity Density	GN, ZKC, BD, KPB, ACF	Zhang et al. 2016 [54]
	MOPSO-Net	Kernel K-Means and Ratio Cut	GN, ZKC, BD, KPB, ACF	Rahimi et al. 2018 [55]
AC	ACOMRW	Fraction of Vertices Classified Correctly	RN, ZKC, BD, ACF	Jin et al. 2011 [26]
	ACODCS	Modularity	GN, ZKC, BD, KPB, ACF	Chen et al.2012 [1]
	ACO	Modularity	GN, ZKC, BD, KPB, ACF	Chang et al. 2013 [27]
EDA	UMDA	Modularity or Community Score	GN, LFR, BD, KPB, ACF	Parsa et al. 2015 [20]
DE	DECD	Modularity	GN, ZKC, ACF	Jia et al. 2012 [16]
	CCDECD	Modularity	ZKC, BD, ACF, PPI, Erdos	Huang et al. 2012 [17]
	IDDE	Modularity	GN, LFR, ZKC, BD, KPB, ACF, SFI, JM	Zhang et al. 2015 [18]
	CoCoMi	Modularity	ZKC, BD, KPB, ACF, CN, EMC, Erdos, PGP	He et al. 2016 [3]
	CDEMO	Modularity	GN, LFR, ZKC, BD, KPB, ACF	Xiao et al. 2018 [19]
FOA	CDMFOA	Modularity or Modularity Density	GN, LFR, ZKC, BD, KPB, ACF	Liu et al. 2016 [28]
Bat	BA	Modularity	ZKC, BD, ACF	Hassan et al. 2015 [29]
	DBA	Modularity	GN, ZKC, KPB, ACF	Song et al. 2016 [30]
FSA	FSA	Modularity and Conductance	LFR, KPB, ACF, EMC, NS, PGP, Twitter	Pattanayak et al. 2019 [5]
STA	MDSTA	Modularity	GN, ZKC, BD, KPB, ACF, JM, EMC, NS, PG	Zhou et al. 2019 [8]

NMI is mainly used to measure the similarity between the detected community structures and the real ones in synthetic networks. The value of NMI ranges from 0 to 1, and a larger value of NMI indicates a higher detection performance. Detailed description of the parameters in (7) can also be found in the literature [3], [19].

C. Algorithms for Performance Comparison

Up to now, many algorithms have been proposed for community detection. Classical algorithms are most widely used, which usually use traditional techniques, such as greedy technique, mathematical programming, and spectral optimization, for modularity optimization [45]. Typical examples include GN [46], Fast Nm [47], CNM [48], BGLL [49], MSFCM [50], FMM/H1 [44], and so on. In recent years, an impressive growth of EAs and bio-inspired methods was appeared, approaching the community detection problem with nature-inspired computational models. We summarize many state-of-the-art algorithms in Table II, from which a set of representatives are used to make a comparison with SOSCD.

D. Parameter Setting

SOSCD only includes three parameters: NP, $t_{\max}$, and δ. NP represents the population size. $t_{\max}$ means the maximum number of iterations, which is used as the termination condition. If the number of iterations of the main loop in an algorithm exceeds $t_{\max}$, the algorithm will be terminated. δ is used as a threshold value in the NCM operation, determining whether the community membership of the target node should be modified. We calculate the probability of a node that it is not in the same community as its neighbors. If the probability is larger than δ, it will be placed into a new community.

SOSCD is not sensitive to parameters, and basic parameter setting can meet most network detection requirements. In our experiment, SOSCD uses the same parameter setting on all test networks, $NP = 100$, $t_{\max} = 200$, and $\delta \in [0.2, 0.6]$. For large and complex real networks, a smaller value of δ can improve the global search performance of SOSCD for detecting the global optimal partition.

E. Validation of Optimization Strategy

In SOSCD, a novel bio-inspired algorithm (i.e., SOS) has been served as the optimizing strategy for modularity optimization. In order to prove the superiority of SOS, a comparison between SOS and other three popular optimization strategies in community detection is made, including GA, PSO, and DE.

For a fair comparison, three independent community detection algorithms based on the standard GA, PSO, and DE are constructed, named GACD, PSOCD, and DECD separately. All the three competitors follow the same optimization framework as SOSCD with only the optimization strategy being

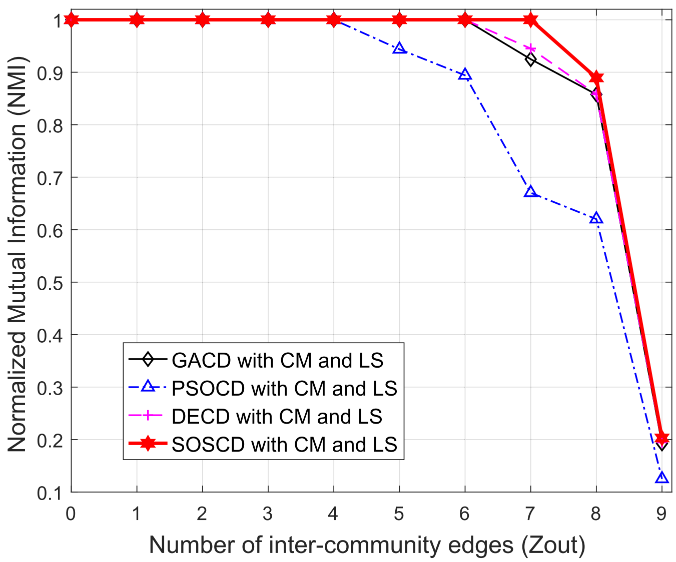

Fig. 4. Comparison of optimization strategies on GN networks.

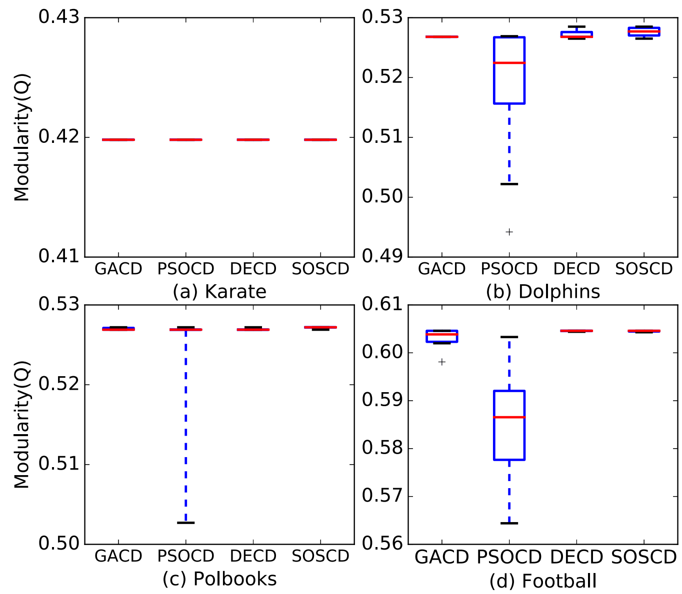

Fig. 5. Comparison of optimization strategies on real-world networks. (a) Karate network. (b) Dolphins network. (c) Polbooks network. (d) Football network.

replaced with the standard GA, PSO, and DE. In all the experiments, the population size NP and maximum number of iterations $t_{\max}$ are set to be 100 and 200, respectively. The threshold value δ utilized in NCM operation is assigned in the range of $[0.2, 0.6]$. Special parameters in different optimization strategies are set as follows: the crossover rate $Pc = 0.8$ and mutation rate $Pm = 0.2$ in GA, the learning factors $C1, C2 = 2$ in PSO, the mutation factor $F = 0.9$, and crossover factor $CR = 0.3$ in DE.

First, the synthetic benchmark network (GN) is adopted to test the accuracy and stability of all the algorithms. Since the correct community structure of GN networks can be obtained, we mainly focus on whether a detection algorithm can identify the real community partition. Test results are presented in Fig. 4, where each data point represents an average optimal value of NMI acquired by each algorithm on each network in 30 independent runs.

As shown in Fig. 4, all the algorithms can achieve the best value of NMI when z_{out} is smaller than 4, which means that real community structures can be efficiently identified. However, when z_{out} is larger than 4, differences begin to appear. Specifically, when z_{out} increases from 5 to 6, all the algorithms except the PSOCD, can still get the optimum. However, when z_{out} increases from 7 to 8, all the algorithms deteriorate sharply, where PSOCD shows the fastest decline in accuracy, and GACD and DECD perform relatively better than PSOCD with the value of NMI always greater than 0.8. SOSCD performs the best among the four competitors because its NMI always achieves 1.0 when z_{out} is not larger than 7 and with the highest accuracy in other cases.

Second, four real-world networks shown in Table I are also utilized to further examine the detection performance of SOSCD. Experimental results in Fig. 5 shows the box plots of statistical values of modularity obtained by all of the four competitors over ten runs. In each box plot, the red line represents the average best values of modularity, and the edges of the box indicate the upper quartile and lower quartile, respectively. Whiskers extend to the most extreme data is no more than 1.5 times the length of the box away from the box, and the outliers are represented by crosses.

As we can see from Fig. 5(a), all the average best values of modularity are equal to the maximum value ($Q = 0.4198$), and there are no outliers, which illustrates that all the algorithms can identify real partitions of the Karate Club network with high efficiency. Fig. 5(b) shows that the performance difference between algorithms on the Bottlenose dolphin network is much greater than that on the Zachary Karate Club network. As we can see, SOSCD performs rather well and outperforms the other three competitors in terms of accuracy and stability according to the statistical values of modularity. Among the three competitors, the performance of GACD is remarkably consistent, but it never finds the global optimum. PSOCD performs rather unstable, and its accuracy is the worst. Compared with GACD and PSOCD, the performance of DECD is very competitive, which gets a detection result closing to the global optimum obtained by SOSCD ($Q = 0.5285$). However, it is not as accurate and stable as SOSCD. Fig. 5(c) shows the modularity optimization results on the political book network. Although all the algorithms perform rather stable and almost have the same median values of modularity, the optimal partition obtained by SOSCD is still the best. Statistic values obtained by GACD, PSOCD, and DECD are all close to 0.5269, but the average best value acquired by SOSCD can reach 0.5272. Fig. 5(d) shows the detection results on the American college football network. Due to the complexity of the network, large differences are witnessed from the detection results of the four competitors. PSOCD still performs the worst and the average best value is not larger than 0.5900. DECD and SOSCD perform rather well on this network, and both of them can achieve the global optimal partition ($Q = 0.6046$) with excellent stability.

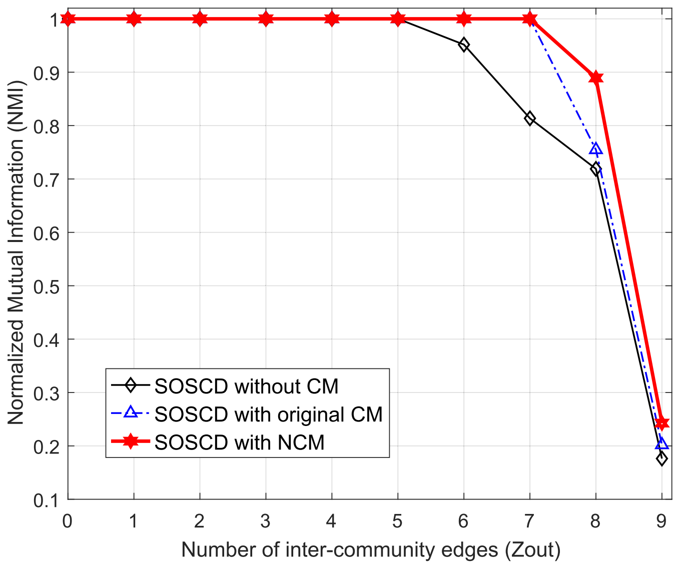

Fig. 6. Performance validation of NCM on GN networks.

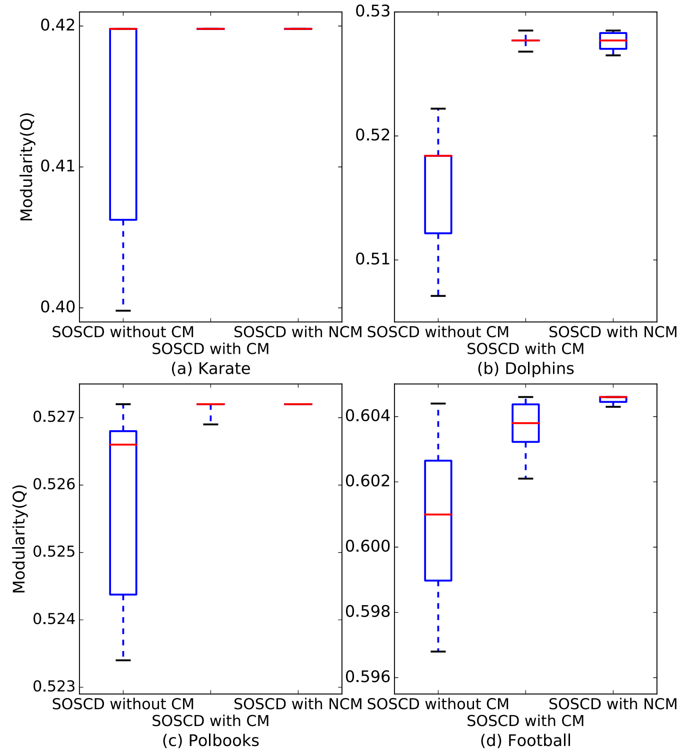

Fig. 7. Performance validation of NCM on real-world networks. (a) Karate network. (b) Dolphins network. (c) Polbooks network. (d) Football network.

To sum up, SOSCD can efficiently discover community structures from both the synthetic benchmark and real-world networks. Experimental results make us believe that optimization strategy is truly important in community detection. It can greatly influence the global convergence ability of the detection algorithm and the quality of community partitions. The introduced bio-inspired algorithm (SOS) is truly effective to serve as the optimization strategy, and it can improve the global convergence performance of SOSCD.

F. Validation of Neighbor-Based Community Modification

SOSCD employs a NCM strategy to improve the quality of the candidate community partitions in the population. NCM is designed based on the utilization of node neighborhood information and acts as a LS operation to assist the global search by promoting exploitation.

In order to verify the effectiveness of NCM strategy on the performance of SOSCD, another two competitors based on the framework of SOSCD are designed: one is with the well-known clean-up operation that we name it as CM for comparison purposes and the other does not use any CM strategy for LS. All of the three algorithms are tested on both the synthetic benchmark and real-world networks, and their performance is compared by assessing the quality of the obtained optimal partitions.

We first test the algorithms on GN networks to illustrate the efficiency of the NCM strategy. Fig. 6 shows the comparison results on ten different GN networks with z_{out} increasing from 0 to 9. Accuracy and stability are evaluated according to the average best values of NMI acquired in 30 independent runs. From the test results shown in Fig. 6, when z_{out} is smaller than 5, all the three algorithms can efficiently find out true partitions with NMI $= 1$. However, when z_{out} increases from 6 to 7, except the algorithm without CM strategy, another two competitors can still get the global optimum. Moreover, when z_{out} is larger than 7, the performance of all the algorithms declines very quickly, and SOSCD with NCM acquires relatively better results but still fails to identify the actual structures.

Apart from GN networks, real-world networks are also adopted to test the performance of the NCM strategy, and the statistical results are shown in Fig. 7, which presents the box plots of statistical values of Q recorded over ten runs. It can be observed in Fig. 7 that SOSCD without CM strategy performs the worst, especially on the Dolphins and Football networks. It can only achieve the best median value of Q on the Zachary Karate Club network and gets the lowest values on the other three networks. In addition, its statistical consequences are very volatile. On the contrary, SOSCD with NCM strategy always obtains the largest value of Q, and its outliers are much smaller than those of the others. SOSCD with the original CM strategy performs relatively competitive. It can achieve robust results on most of the test networks, and its accuracy and stability can be significantly improved comparing with the one without CM strategy.

From the above-mentioned experimental results, we can see that the CM strategy is truly an effective LS method for the performance improvement of community detection. Especially, the newly designed NCM strategy is of great importance in the framework of SOSCD, which can effectively enhance the performance of SOSCD in terms of precision and stability.

G. Validation of Neighbor-Based Local Search

In SOSCD, the NLS strategy is designed based on node neighborhood information and acted as a more-refined LS operation performed after NCM to help high-quality but

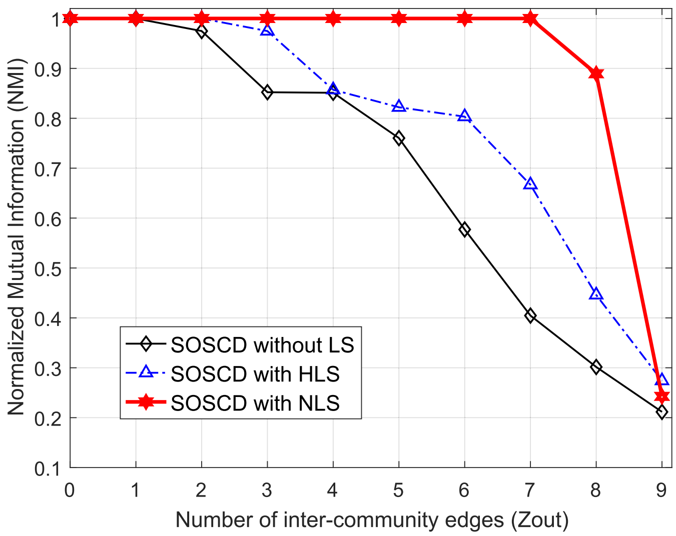

Fig. 8. Performance validation of NLS on GN networks.

suboptimal individuals jump out of the local optimum and increase the probability of finding the global optimal partition.

In order to test the effectiveness of NLS strategy on the performance of SOSCD, another two competitors based on the framework of SOSCD are also designed: one is with the hill-climbing LS (HLS) strategy and the other does not use any LS strategy. In the experiment, all the three algorithms are tested on both the synthetic and real-world networks, and their performance is compared by evaluating the quality of the obtained optimal partition.

We first test the algorithms on GN networks to illustrate the efficiency of the NLS strategy. Fig. 8 shows the comparison result on ten different GN networks with z_{out} increasing from 0 to 9. Accuracy and stability are measured according to the average best values of NMI acquired in 30 independent runs. As shown in Fig. 8, when z_{out} is smaller than 1, all the three algorithms can find the true partitions. When z_{out} increases from 2 to 7, both the algorithms without the LS and with HLS deteriorate rapidly, only SOSCD with NLS can still find out the real community structures. However, when z_{out} is greater than 7, none of the three algorithms can achieve the global optimum, but SOSCD still performs the best. When z_{out} rises to 8, the value of NMI obtained by SOSCD remains close to 0.9.

To further validate the superiority of NLS in promoting the global convergence ability of SOSCD, performances of the three algorithms are further studied on real-world networks. As shown in Fig. 9, statistical values of Q obtained by the three algorithms are presented in the box plots. It can be seen that the convergence performance of the three competitive algorithms is quite different. For the algorithm without LS, the accuracy is always the worst since the statistic values of Q are always the smallest. For the algorithm with HLS, the average best value is greatly improved comparing with the one without LS, especially on the Karate and Football networks. By contrast, the algorithm with NLS is always better than the other two competitors, as its median values represented by the red lines are always the largest and the

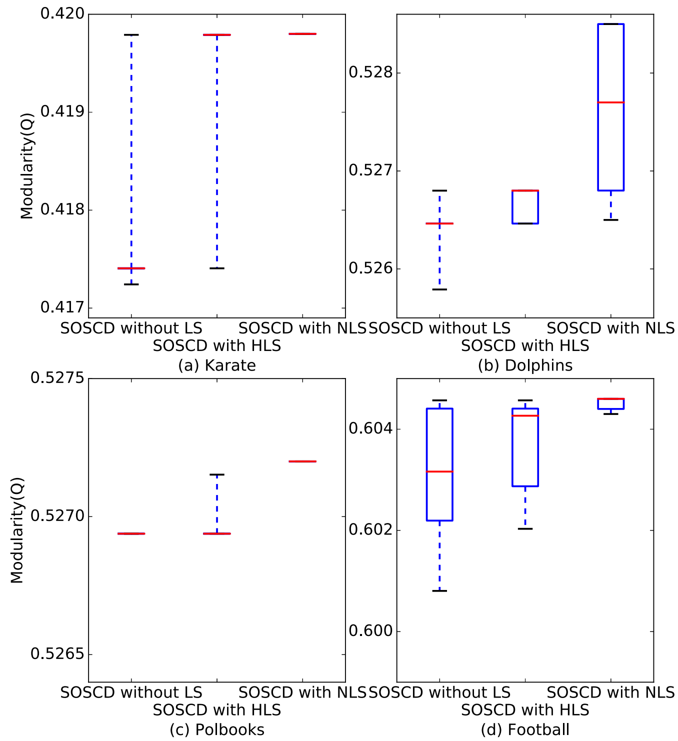

Fig. 9. Performance validation of NLS on real-world networks. (a) Karate network. (b) Dolphins network. (c) Polbooks network. (d) Football network.

outliers are relatively small except for the result on the Bottlenose dolphin network.

The above-mentioned experimental results make us believe that NLS in SOSCD is truly important for making a more-refined and accurate LS. It utilizes neighborhood information of vertexes to avoid a random LS and thus further enhances the exploitation ability. NLS can greatly affect the quality of the high-performance candidate partitions and increase the probability of find the global optimum.

V. Performance Comparison Between SOSCD and State-of-the-Art Algorithms

This section provides a detailed comparison between SOSCD and many typical state-of-the-art modularity optimization algorithms. Community detection performance of all the algorithms is verified on both the synthetic benchmarks and well-studied real-world networks.

A. Performance Comparison on GN Benchmarks

In the following, we first verify the effectiveness of SOSCD on a set of gradually changed GN networks with z_{out} increasing from 0 to 8 with an interval of 1.0. We choose 11 typical and high-performance modularity optimization algorithms for comparison, including GN [46], CNM [48], GATHB [11], ECGA [13], MPSOA [54], UMDA [20], BGLL [49], DECD [16], CDEMO [19], CDMFOA [28], and MDSTA [8]. Among all the competitors, GN, CNM, and BGLL are the three traditional and classical algorithms designed based on deterministic optimization strategies, while the others are metaheuristics based on typical EAs (GA, EDA,

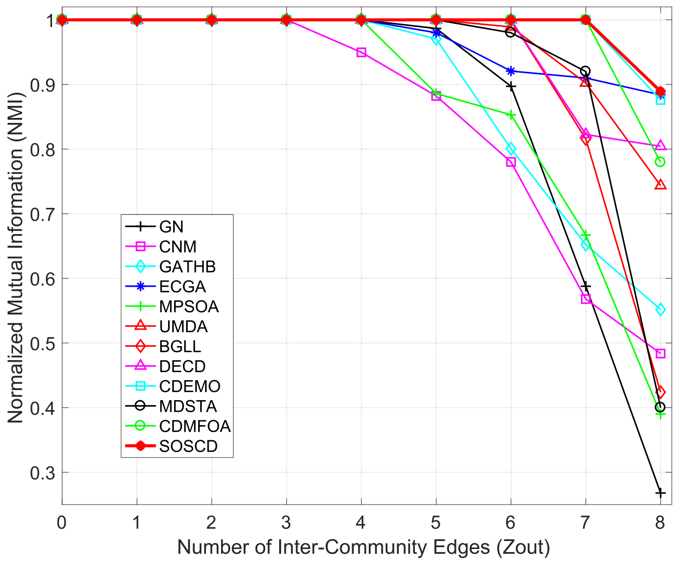

Fig. 10. Best NMI of SOSCD and 11 competitors on a set of GN networks with z_{out} increasing from 0 to 8 with an interval of 1.0. Each data point is obtained based on averaging 30 times running of each algorithm.

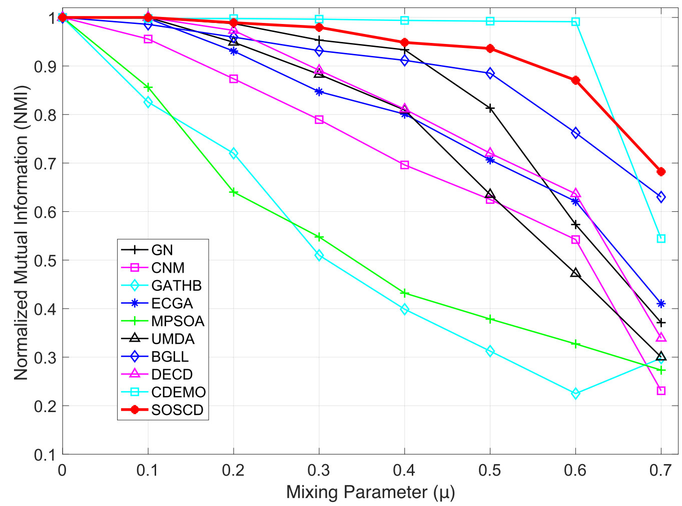

Fig. 11. Best NMI of SOSCD and nine competitors on a set of LFR networks with u increasing from 0 to 0.7 with an interval of 0.1. Each data point is obtained based on average 30 times running of each algorithm.

and DE) or nature-inspired approaches (PSO, FOA, and STA). Fig. 10 shows the variation of the detection results obtained by SOSCD and all the competitors, in which each data point represents an average best value of NMI. For some of the competitors shown in Fig. 10, such as the MDSTA, CDMFOA, UMDA, MPSOA, and ECGA, we collect part of the results from the literature [8], [13], [20], [28], [54], and we run the codes of other algorithms to obtain the results after averaging 30 times running of each algorithm.

As shown in Fig. 10, all the algorithms perform very well and the real partitions of GN networks can be easily found out when $z_{\text{out}} \leqslant 3$. As the value of z_{out} gradually increases, the community structure begins to become more vague. When z_{out} increases from 4 to 6, the performance of GN, CNM, GATHB, MPSOA, and ECGA begins to decline dramatically, while other algorithms can basically get the optimal partition. However, when z_{out} is greater than 6, all the algorithms deteriorate very quickly and almost all of them get stuck into local optimum except SOSCD, CDEMO, and CDMFOA. If the value of z_{out} rises to 8, none of the algorithms can detect real communities, but SOSCD, CDEMO, and ECGA can still obtain competitive detection results. However, the performance of ECGA is not as good as that of the SOSCD and CDEMO at the conditions of $z_{\text{out}} = 5$, 6, and 7.

To sum up, SOSCD can always obtain the best detection results on all the GN networks, according to the statistical results of NMI shown in Fig. 10. Compared with many state-of-the-art modularity optimization algorithms, SOSCD can get very competitive detection results with high-quality optimal partitions in terms of precision and stability.

B. Performance Comparison on LFR Benchmarks

Second, we try to verify the effectiveness of SOSCD on a set of gradually changed LFR networks with μ increasing from 0 to 0.7 with an interval of 0.1 to further investigate the robust of SOSCD. Similarly, we choose nine typical and high-performance modularity optimization algorithms for comparison, including GN [46], CNM [48], GATHB [11], ECGA [13], MPSOA [54], UMDA [20], BGLL [49], DECD [16], and CDEMO [19]. Fig. 11 shows the variation of the detection results obtained by SOSCD and all the competitors, in which each data point represents an average best value of NMI. For some of the competitors shown in Fig. 11, such as the UMDA, MPSOA, and ECGA, we collected the part of the results from the literature [13], [20], [54] and run the codes of the other algorithms to obtain the results after averaging 30 times running of each algorithm.

From the variation curve presented in Fig. 11, it can be observed that compared with GN benchmark networks, LFR networks are much more difficult to detect because their community structures are more complicated, considering the number, scale, and diversity of communities. As a result, the performance of all the algorithms is not as satisfactory as that on GN networks. Among all the tested algorithms, CDEMO and SOSCD perform relatively superior to others, as illustrated by the value of NMI on each network, especially when $u \leqslant 0.6$. On the networks with small values of u ($u \leqslant 0.2$), CDEMO, SOSCD, GN, DECD, BGLL, UMDA, and ECGA obtain similar and competitive detection results. However, most of the algorithms deteriorate very quickly with the growth of u, and at the same time, the difference among algorithms are gradually increasing. When u is between 0.2 and 0.4, CDEMO, SOSCD, GN, and BGLL outperform the others. However, when u is larger than 0.4, the performance of GN decreases rapidly, while CDEMO, SOSCD, and BGLL show good stability and their superiority in terms of accuracy becomes much more significant. As u increases to 0.7, most algorithms can no longer achieve accurate detection results, while SOSCD, CDEMO, and BGLL can still detect significant community structures. Compared with the most competitive algorithm CDEMO, although the accuracy of SOSCD is lower than that of CDEMO when u is less than 0.6, SOSCD shows better stability when u increases from 0.6 to 0.7.

To sum up, SOSCD can always obtain accurate community partitions on all the tested LFR networks, illustrated by the averaging best value of NMI. In another word, SOSCD can efficiently identify community structures from LFR benchmark networks and with excellent scalability and accuracy.

C. Performance Comparison on Small-Scale Real Networks

Third, to further investigate the detection ability of SOSCD, we attempt to verify its effectiveness and superiority on four most widely used small-scale real-world networks. A detailed comparison between SOSCD and many state-of-the-art modularity optimization algorithms is implemented. According to the incorporated optimization strategies, all the competitors are roughly classified into three categories. The first category contains six traditional and classical algorithms, most of which are designed based on deterministic optimization strategies, including GN [46], Fast Nm [47], CNM [48], BGLL [49], MSFCM [50], and FMM/H1 [44]. The second category includes eight well-known metaheuristics based on typical EAs, such as GA and DE, including MOEA/D-Net [51], GATHB [11], ECGA [13], CC-GA [53], DECD [16], CCDECD [17], IDDE [18], and CDEMO [19]. The last category mainly focuses on methods based on nature-inspired approaches, such as DBA [30], ACODCS [1], MPSOA [54], MODPSO [23], CDMFOA [28], MDSTA [8], and SOSCD. We test all the algorithms on the four small-scale real-world networks and utilize modularity as the metric for measurement. Table III records the highest value of Q obtained by each algorithm, part of the results for competitors (such as ECGA, CC-GA, DBA, ACODCS, MPSOA, CDMFOA, and MDSTA), are collected from the literature [1], [3], [8], [13], [28], [30], [53], [54], while others are acquired by running the codes of each algorithm for an average of 30 times.

From the statistical data presented in Table III, it can be observed that traditional and classical community detection algorithms are able to find out certain community structures from real-world networks, but their accuracy is barely satisfactory and the difference between their performance is not significant. For example, on the Karate network, none of the algorithms can obtain the best value of Q ($Q = 0.4198$) that is the highest modularity value we have known and all of the algorithms fall into local optimum.

Compared with classical algorithms, the metaheuristics based on typical EAs show better performance, where the first four algorithms are designed based on GA and the others take DE as the optimization strategy. As we can see, ECGA, CC-GA, and the four DE-based algorithms can get the optimal partition containing four communities with the value of Q equal to 0.4198 on the Karate network. Besides, on the Dolphins network, the novel GA-based algorithm CC-GA and the DE-based algorithm CDEMO obtain the best partition with Q equal to 0.5285. All of the four DE-based algorithms achieve the best partition on the Football network. After carefully examination, we find that ECGA and CC-GA perform better than other GA-based algorithms. In addition, DE-based algorithms show very competitive performance when compared with the GA-based ones.

TABLE III

HIGHEST VALUE OF Q OBTAINED BY CLASSICAL ALGORITHMS, METAHEURISTICS BASED ON TYPICAL EAs AND NATURE-INSPIRED METHODS. NOTE THAT THE BEST RESULT FOR EACH NETWORK IS HIGHLIGHTED IN BOLDFACE

Algorithm	Karate	Dolphins	Polbooks	Football
GN	0.4013	0.5194	0.5099	0.5994
Fast Nm	0.3810	0.4960	0.5020	0.5490
CNM	0.3807	0.4950	0.5019	0.5770
BGLL	0.4150	0.4950	0.5150	0.6010
MSFCM	0.4132	0.3991	0.4601	0.5268
FMM/H1	0.3941	0.4882	0.5175	0.5960
MOEA/D-Net	0.3715	0.3735	0.5180	0.6005
GATHB	0.4024	0.5219	0.5176	0.5508
ECGA	**0.4198**	0.5242	0.5269	0.6010
CC-GA	**0.4198**	**0.5285**	0.5270	0.5940
DECD	**0.4198**	0.5249	0.5262	**0.6046**
CCDECD	**0.4198**	0.5216	0.5268	**0.6046**
IDDE	**0.4198**	0.5282	0.5271	**0.6046**
CDEMO	**0.4198**	**0.5285**	0.5271	**0.6046**
DBA	0.4028	0.5277	0.5265	0.5990
ACODCS	0.4165	0.5268	0.5262	0.6031
MPSOA	**0.4198**	0.5191	0.5255	0.6030
MODPSO	**0.4198**	0.5268	0.5260	**0.6046**
CDMFOA	**0.4198**	**0.5285**	**0.5272**	0.6033
MDSTA	**0.4198**	**0.5285**	**0.5272**	**0.6046**
SOSCD	**0.4198**	**0.5285**	**0.5272**	**0.6046**

The last section in Table III provides relevant detection results of a set of metaheuristics based on typical nature-inspired methods, with BA, ACO, PSO, FOA, STA, and SOS as optimization strategies, respectively. The performance of these emerging algorithms is competitive when compared with the first two categories of methods in terms of accuracy. In particular, SOSCD, MDSTA, and CDMFOA show excellent detection performance among all the competitors since they can basically obtain the best results on almost all the tested small-scale real-world networks.

D. Performance Comparison on Large-Scale Real Networks

Finally, three well-known large-scale real-world networks are used to further test the scalability of SOSCD, including the NetScience network, the PowerGrid network, and the PGP network. The NetScience network represents the collaborate relationship among scientists, which contains 1589 nodes and 2742 edges [5], [8]. The PowerGrid network with 4941 nodes and 6594 edges is constructed based on the high-voltage power grid in the U.S. [8]. The PGP network is a web-trust network of signature for private communication, containing 10 680 nodes and 24 316 edges [5].

The SOSCD is compared with six different well-known state-of-the-art community detection algorithms, including LPA [56], GN [46], BGLL [49], CC-GA [53], MDSTA [8], and FSA [5]. CC-GA is a clustering coefficient-based GA with novel population initialization and mutation methods. MDSTA is a novel modularity-based discrete STA, which is constructed based on a novel intelligent optimization algorithm STA. FSA

TABLE IV

MAXIMUM AND AVERAGE VALUES OF Q OBTAINED BY LPA, GN, BGLL, CC-GA, MDSTA, FSA, AND SOSCD ALGORITHMS FOR THE NETSCIENCE, POWER GRID, AND PGP NETWORKS. NOTE THAT THE HIGHEST PERFORMANCE OF Q_{BEST} AND Q_{MEAN} FOR EACH NETWORK IS HIGHLIGHTED IN BOLDFACE. THE SYMBOL "–" INDICATES THAT THE RESULTS OF THE CORRESPONDING ALGORITHMS CANNOT BE FOUND FROM THE CORRESPONDING LITERATURE

Network	NetScience (NS)		Power Grid (PG)		PGP	
Modularity	Q_{best}	Q_{mean}	Q_{best}	Q_{mean}	Q_{best}	Q_{mean}
LPA	0.9255	0.9197	0.7532	0.7471	0.8192	0.8055
GN	0.9579	0.9579	0.9330	0.9330	0.8516	0.8516
BGLL	0.9597	**0.9597**	0.9356	0.9356	0.8794	**0.8794**
CC-GA	0.9580	0.9550	–	–	0.8520	0.8430
MDSTA	**0.9599**	**0.9597**	**0.9376**	0.9345	–	–
FSA	0.9585	0.9585	–	–	0.8717	0.8717
SOSCD	**0.9599**	**0.9597**	0.9369	**0.9365**	**0.8796**	**0.8794**

is a fire spread community detection algorithm that is inspired by the fire propagation model.

Since the real partitions of the three large-scale real-world networks are unknown, we also utilize modularity as the metric for comparison. Each of the experiment is conducted for ten runs, and the maximum and average values of Q (i.e., Q_{best} and Q_{mean}) are recorded in Table IV. Note that in Table IV, the experimental results for CC-GA, MDSTA, and FSA are collected from the literature [5], [8], [53]. The highest values of Q_{best} and Q_{mean} obtained by all of the algorithms are shown in bold face, and the symbol "–" indicates that the results cannot be found from the corresponding literature.

As we can see from Table IV, among all the seven algorithms, SOSCD obtains the highest Q_{best} values of 0.9599 and 0.8796 on NetScience and PGP networks, respectively. Compared with the results of LPA, GN, and BGLL on the PowerGrid network, SOSCD can obtain a partition with higher Q_{best}, and it is only less than that of MDSTA. On all of the three networks, SOSCD can always get the partitions with the highest Q_{mean} among these seven algorithms. From the above-mentioned experiment results in large-scale networks, it can be concluded that SOSCD has very excellent scalability on community detection.

From the above-mentioned experimental results, we can conclude that SOSCD is an effective bio-inspired metaheuristic algorithm for community detection. It can identify modules from both of synthetic and real-world networks with high precision and stability, outperforming many state-of-the-art modularity optimization algorithms.

VI. CONCLUSION

In this article, a discrete bio-inspired metaheuristic algorithm called SOSCD is designed to improve the performance of modularity optimization for community detection of networks. By simultaneously enhancing the convergence ability of the incorporated optimization strategy and making sufficient and rational utilization of neighborhood information to assist global search, the global convergence performance of SOSCD in modularity optimization is greatly improved. Experimental results on two sets of well-known synthetic benchmarks and typical real-world networks from small to large scale validate the effectiveness and superiority of SOSCD in community detection. Compared with many state-of-the-art modularity optimization algorithms, SOSCD can significantly improve the quality of community partition in terms of precision and stability.

The innovation of SOSCD mainly includes three aspects. First, a novel bio-inspired method, SOS, is discretized and incorporated in SOSCD as the optimization strategy. It effectively improves the global convergence performance of SOSCD in modularity optimization when compared with other optimization strategies, such as the standard GA, PSO, and DE. Second, an NCM strategy is newly designed based on the well-known clean-up operation but with more relaxed restriction when modifying the communities of nodes for better partitions. By taking a more rational utilization of node neighborhood information, it can efficiently enhance the exploitation ability of SOSCD and avoid falling into local optimum at the same time. The effectiveness of NCM in correcting wrong partitions and improving the quality of individuals has been verified. At last, an NLS strategy is also proposed, which helps high-quality but suboptimal individuals escape from local optimum and increase the probability of finding the global optimum. Different from other LS methods, NLS can avoid random invalid search and reduce computational complexity by using node neighborhood information.

The above-mentioned improvement measures make SOSCD accurate and robust in recovering communities from complex networks, providing a high-efficient method for further investigation of networks, such as link prediction [57], [58], user recommendation [59], and information propagation [60], [61]. In addition, due to the advantages of requiring no special parameters and feature information of networks, SOSCD can be extended for application in more kinds of complex networks, such as signed networks [31], [62] and the networks with fuzzy overlapping communities [44]. In this article, SOSCD is mainly designed to deal with networks with less than 10 000 nodes. In our future work, we will concentrate on handling large-scale networks with millions of nodes that are very common in the big data era.

REFERENCES

[1] C. Pizzuti, "Evolutionary computation for community detection in networks: A review," *IEEE Trans. Evol. Comput.*, vol. 22, no. 3, pp. 464–483, Jun. 2018.

[2] Q. Cai, L. Ma, M. Gong, and D. Tian, "A survey on network community detection based on evolutionary computation," *Int. J. Bio-Inspired Comput.*, vol. 8, no. 2, pp. 84–98, 2016.

[3] S. He *et al.*, "Cooperative co-evolutionary module identification with application to cancer disease module discovery," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 874–891, Dec. 2016.

[4] W. Zhang, R. Zhang, R. Shang, J. Li, and L. Jiao, "Application of natural computation inspired method in community detection," *Phys. A, Stat. Mech. Appl.*, vol. 515, pp. 130–150, Feb. 2019.

[5] H. S. Pattanayak, A. L. Sangal, and H. K. Verma, "Community detection in social networks based on fire propagation," *Swarm Evol. Comput.*, vol. 44, pp. 31–48, Feb. 2019.

[6] M. Coscia, F. Giannotti, and D. Pedreschi, "A classification for community discovery methods in complex networks," *Stat. Anal. Data Mining, ASA Data Sci. J.*, vol. 4, no. 5, pp. 512–546, 2011.

[7] B. S. Khan and M. A. Niazi, "Network community detection: A review and visual survey," 2017, *arXiv:1708.00977*. [Online]. Available: https://arxiv.org/abs/1708.00977

[8] X. Zhou, K. Yang, Y. Xie, C. Yang, and T. Huang, "A novel modularity-based discrete state transition algorithm for community detection in networks," *Neurocomputing*, vol. 334, pp. 89–99, Mar. 2019.

[9] M. Chen, K. Kuzmin, and B. K. Szymanski, "Community detection via maximization of modularity and its variants," *IEEE Trans. Comput. Social Syst.*, vol. 1, no. 1, pp. 46–65, Mar. 2014.

[10] C. Pizzuti, "GA-Net: A genetic algorithm for community detection in social networks," in *Proc. Int. Conf. Parallel Problem Solving Nature (PPSN X)*, 2008, pp. 1081–1090.

[11] M. Tasgin, A. Herdagdelen, and H. Bingol, "Community detection in complex networks using genetic algorithms," 2008, *arXiv:0711.0491*. [Online]. Available: https://arxiv.org/abs/0711.0491

[12] D. Jin, B. Yang, J. Liu, D. He, and L. Dayou, "Genetic algorithm with local search for community detection in large-scale complex networks," *Acta Automatica Sinica*, vol. 37, no. 7, pp. 873–882, 2011.

[13] J. Li and Y. Song, "Community detection in complex networks using extended compact genetic algorithm," *Soft Comput.*, vol. 17, no. 6, pp. 925–937, 2013.

[14] Z. Li and J. Liu, "A multi-agent genetic algorithm for community detection in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 449, pp. 336–347, May 2016.

[15] C. Pizzuti, "A multiobjective genetic algorithm to find communities in complex networks," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 418–430, Jun. 2012.

[16] G. Jia *et al.*, *Community Detection in Social and Biological Networks Using Differential Evolution*. Berlin, Germany: Springer, 2012.

[17] Q. Huang *et al.*, *Community Detection Using Cooperative Co-Evolutionary Differential Evolution*. Berlin, Germany: Springer, 2012.

[18] Y.-J. Zhang, Z.-H. Gong, and Q.-K. Chen, "Community detection in complex networks using immune discrete differential evolution algorithm," *Acta Automatica Sinica*, vol. 41, no. 4, pp. 749–757, 2015.

[19] J. Xiao, Y.-J. Zhang, and X.-K. Xu, "Convergence improvement of differential evolution for community detection in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 503, pp. 762–779, Aug. 2018.

[20] M. G. Parsa, N. Mozayani, and A. Esmaeili, "An EDA-based community detection in complex networks," in *Proc. Int. Symp. Telecommun.*, 2014, pp. 476–480.

[21] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, and L. Jiao, "Greedy discrete particle swarm optimization for large-scale social network clustering," *Inf. Sci.*, vol. 316, pp. 503–516, Sep. 2015.

[22] D. Zhou and X. Wang, "A neighborhood-impact based community detection algorithm via discrete PSO," *Math. Problems Eng.*, vol. 2016, no. 4, 2016, Art. no. 3790590.

[23] M. Gong, Q. Cai, X. Chen, and L. Ma, "Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 82–97, Feb. 2014.

[24] Y. Liu, J. Luo, H. Yang, and L. Liu, "Finding closely communicating community based on ant colony clustering model," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, 2010, pp. 127–131.

[25] S. Sadi, Ş. Ögüdücü, and A. Ş. Uyar, "An efficient community detection method using parallel clique-finding ants," in *Proc. Evol. Comput.*, Jul. 2010, vol. 5, no. 1, pp. 1–7.

[26] D. Jin, D. Liu, B. Yang, C. Baquero, and D. He, *Ant Colony Optimization With Markov Random Walk for Community Detection in Graphs*. Berlin, Germany: Springer, 2011.

[27] C. Honghao, F. Zuren, and R. Zhigang, "Community detection using ant colony optimization," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 3072–3078.

[28] Q. Liu, B. Zhou, S. Li, A.-P. Li, P. Zou, and Y. Jia, "Community detection utilizing a novel multi-swarm fruit fly optimization algorithm with hill-climbing strategy," *Arabian J. Sci. Eng.*, vol. 41, no. 3, pp. 807–828, 2016.

[29] E. A. Hassan, A. I. Hafez, A. E. Hassanien, and A. A. Fahmy, "A discrete bat algorithm for the community detection problem," in *Hybrid Artificial Intelligent Systems* (Lecture Notes in Computer Science), vol. 9121. Cham, Switzerland: Springer, 2015.

[30] A. Song, M. Li, X. Ding, W. Cao, and K. Pu, "Community detection using discrete bat algorithm," *IAENG Int. J. Comput. Sci.*, vol. 43, no. 1, pp. 37–43, 2016.

[31] Q. Cai, M. Gong, B. Shen, L. Ma, and L. Jiao, "Discrete particle swarm optimization for identifying community structures in signed social networks," *Neural Netw.*, vol. 58, no. 10, pp. 4–13, 2014.

[32] M.-Y. Cheng and D. Prayogo, "Symbiotic Organisms Search: A new metaheuristic optimization algorithm," *Comput. Struct.*, vol. 139, pp. 98–112, Jul. 2014.

[33] S. Al-Sharhan and M. G. H. Omran, "An enhanced symbiosis organisms search algorithm: An empirical study," *Neural Comput. Appl.*, vol. 29, no. 11, pp. 1025–1043, 2018.

[34] A. E. Ezugwu and D. Prayogo, "Symbiotic organisms search algorithm: Theory, recent advances and applications," *Expert Syst. Appl.*, vol. 119, pp. 184–209, Apr. 2019.

[35] P. Wagenseller, F. Wang, and W. Wu, "Size matters: A comparative analysis of community detection algorithms," *IEEE Trans. Comput. Social Syst.*, vol. 5, no. 4, pp. 951–960, Dec. 2018.

[36] H. V. H. Ayala, C. E. Klein, V. C. Mariani, and L. D. S. Coelho, "Multiobjective symbiotic search algorithm approaches for electromagnetic optimization," *IEEE Trans. Magn.*, vol. 53, no. 6, Jun. 2017, Art. no. 7205504.

[37] V. F. Yu, A. A. N. P. Redi, C.-L. Yang, E. Ruskartina, and B. Santosa, "Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem," *Appl. Soft Comput.*, vol. 52, pp. 657–672, Mar. 2017.

[38] A. Kavousi-Fard, M. A. Rostami, and T. Niknam, "Reliability-oriented reconfiguration of vehicle-to-grid networks," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 682–691, Jun. 2015.

[39] M. Gong, B. Fu, L. Jiao, and H. Du, "Memetic algorithm for community detection in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 84, no. 5, 2011, Art. no. 056101.

[40] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proc. Nat. Acad. Sci. USA*, vol. 99, no. 12, pp. 7821–7826, Apr. 2002.

[41] M. E. J. Newman, "Modularity and community structure in networks," *Proc. Nat. Acad. Sci. USA*, vol. 103, no. 23, pp. 8577–8582, 2006.

[42] D. Lusseau and M. E. J. Newman, "Identifying the role that animals play in their social networks," *Proc. Roy. Soc. B, Biol. Sci.*, vol. 271, no. Suppl_6, pp. S477–S481, 2004.

[43] A. Lancichinetti, S. Fortunato, and F. Radicchi, "Benchmark graphs for testing community detection algorithms," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 78, no. 4, 2008, Art. no. 046110.

[44] J. Su and T. C. Havens, "Quadratic program-based modularity maximization for fuzzy community detection in social networks," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 5, pp. 1356–1371, Oct. 2015.

[45] M. A. Javed, M. S. Younis, S. Latif, J. Qadir, and A. Baig, "Community detection in networks: A multidisciplinary review," *J. Netw. Comput. Appl.*, vol. 108, pp. 87–111, Apr. 2018.

[46] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 2, Feb. 2004, Art. no. 026113.

[47] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, no. 6, 2004, Art. no. 066133.

[48] A. Clauset, M. E. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 70, no. 6, 2004, Art. no. 066111.

[49] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech., Theory Exp.*, vol. 2008, no. 10, pp. 155–168, 2008.

[50] T. C. Havens, J. C. Bezdek, C. Leckie, K. Ramamohanarao, and M. Palaniswami, "A soft modularity function for detecting fuzzy communities in social networks," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1170–1175, Dec. 2013.

[51] M. Gong, L. Ma, Q. Zhang, and L. Jiao, "Community detection in networks by using multiobjective evolutionary algorithm with decomposition," *Phys. A, Stat. Mech. Appl.*, vol. 391, no. 15, pp. 4050–4060, 2012.

[52] M. Guerrero, F. G. Montoya, R. Baños, A. Alcayde, and C. Gil, "Adaptive community detection in complex networks using genetic algorithms," *Neurocomputing*, vol. 266, pp. 101–113, Nov. 2017.

[53] A. Said, R. A. Abbasi, O. Maqbool, A. Daud, and N. R. Aljohani, "CC-GA: A clustering coefficient based genetic algorithm for detecting communities in social networks," *Appl. Soft Comput. J.*, vol. 63, pp. 59–70, Feb. 2018.

[54] C. Zhang, X. Hei, D. Yang, and L. Wang, "A memetic particle swarm optimization algorithm for community detection in complex networks," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 30, no. 2, 2016, Art. no. 1659003.

[55] S. Rahimi, A. Abdollahpouri, and P. Moradi, "A multi-objective particle swarm optimization algorithm for community detection in complex networks," *Swarm Evol. Comput.*, vol. 39, pp. 297–309, Apr. 2018.

[56] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 76, no. 3, 2007, Art. no. 036106.

[57] A. Biswas and B. Biswas, "Community-based link prediction," *Multimedia Tools Appl.*, vol. 76, no. 18, pp. 18619–18639, 2017.

[58] C. De Bacco, E. A. Power, D. B. Larremore, and C. Moore, "Community detection, link prediction, and layer interdependence in multilayer networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 95, nos. 1–4, 2017, Art. no. 042317.

[59] J. D. West, I. Wesley-Smith, and C. T. Bergstrom, "A recommendation system based on hierarchical clustering of an article-level citation network," *IEEE Trans. Big Data*, vol. 2, no. 2, pp. 113–123, Jun. 2016.

[60] C. Stegehuis, R. van der Hofstad, and J. S. H. van Leeuwaarden, "Epidemic spreading on complex networks with community structures," *Sci. Rep.*, vol. 6, Jul. 2016, Art. no. 29748.

[61] R. Paluch, X. Lu, K. Suchecki, B. K. Szymański, and J. A. Hołyst, "Fast and accurate detection of spread source in large complex networks," *Sci. Rep.*, vol. 8, no. 1, 2018, Art. no. 2508.

[62] C. Liu, J. Liu, and Z. Jiang, "A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2274–2287, Dec. 2014.

Jing Xiao received the Ph.D. degree in signal and information processing from Harbin Engineering University, Harbin, China, in 2012. She completed her Post-Doctoral Training in automation at Harbin Engineering University in 2018.

She is currently a Lecturer of information and communication engineering with Dalian Minzu University, Dalian, China. Her current research interests include network community detection, swarm intelligence computing, and many-objective optimization.

Chao Wang received the M.S. and Ph.D. degrees from Harbin Engineering University, Harbin, China, in 2013 and 2018, respectively.

He is currently a Lecturer with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research interests include multi-objective optimization methods and their applications.

Xiao-Ke Xu (M'13) received the Ph.D. degree from the College of Information and Communication Engineering, Dalian Maritime University, Dalian, China, in 2008.

He was a Post-Doctoral Fellow with The Hong Kong Polytechnic University, Hong Kong, and a Visiting Scholar with the City University of Hong Kong, Hong Kong. He is currently a Professor with the College of Information and Communication Engineering, Dalian Minzu University, Dalian. His current research interests include information spreading on complex networks, network community detection, and data mining in social networks.