



# 基于模体的模块度优化高阶社区检测算法

肖 婧, 邹雨承, 吴 双, 许小可\*

(大连民族大学信息与通信工程学院 辽宁 大连 116600)

**【摘要】**为提升现有高阶社区检测算法的性能,提出一种基于模体的模块度优化高阶社区检测算法。通过量化网络模体数量为节点间权重,将高阶基于模体的社区检测转化为低阶基于边的加权网络社区检测,并构造为加权模块度优化问题求解。为保证加权模块度优化性能以获得逼近全局最优的高阶社区划分,一方面优化策略采用典型的生物启发式算法;另一方面综合利用网络低阶和高阶拓扑结构信息,通过节点邻域社区修正操作和局部搜索操作,进一步提升社区划分质量并防止算法陷入局部最优。在人工合成和真实世界网络上的实验表明,利用模体高阶结构信息有助于提升复杂网络社区的检测性能,尤其是在社区结构较为模糊的情况下。该算法能够有效实现基于模体的高阶社区检测,与现有基于模体的典型方法相比在精确性和质量性上表现出一定优势,有助于加深对网络高阶结构及功能特性的理解。

**关键词** 复杂网络; 高阶社区检测; 元启发式算法; 模块度优化; 模体

**中图分类号** TP391 **文献标志码** A **doi**:10.12178/1001-0548.2022111

## A Higher-Order Community Detection Algorithm Based on Motif-Based Modularity Optimization

XIAO Jing, ZOU Yucheng, WU Shuang, and XU Xiaoke\*

(School of Information and Communication Engineering, Dalian Minzu University Dalian Liaoning 116600)

**Abstract** In order to improve the performance of existing higher-order community detection algorithms, a higher-order community detection algorithm based on motif-based modularity optimization is proposed. By quantifying the number of motifs as the weight between nodes, the higher-order community detection based on motifs is transformed into lower-order weighted network community detection based on edges, and a weighted modularity optimization problem is constructed. Based on the meta-heuristic algorithm as the optimization strategy, the lower-order topology structure and higher-order weight information are comprehensively utilized to design the neighborhood community modification operation and local search operation of nodes, so as to improve the quality of community partitions and prevent the algorithm from falling into local optimum. Experimental results on synthetic and real-world networks show that the utilization of motifs is helpful to improve the detection performance under the condition of fuzzy community structure. The proposed algorithm can effectively realize motif-based community detection and has certain advantages in accuracy and quality compared with existing typical motif-based algorithms, which helps to deepen the understanding of the higher-order structure and functional characteristics of complex networks.

**Key words** complex networks; higher-order community detection; metaheuristics; modularity optimization; motifs

社区结构是复杂网络中最重要的结构特性之一,广泛存在于不同类型的真实世界网络中,如生物网络、金融网络、社交网络等<sup>[1-2]</sup>。社区检测不仅能够揭示网络的中尺度拓扑结构特征,而且有助于深入分析网络的功能及动力学特性<sup>[3]</sup>,因此在节点

重要性识别<sup>[4]</sup>、神经元功能分区<sup>[5]</sup>和疾病传播预测<sup>[6]</sup>等领域具有重要的实际应用价值。

近年来涌现出大量基于不同知识背景和社区检测方法<sup>[7-8]</sup>,能够有效识别出网络中隐含的复杂社区结构。然而,现有方法通常仅关注网络中的节点和

收稿日期: 2022-04-10; 修回日期: 2022-08-30

基金项目: 国家自然科学基金(61773091, 62173065); 辽宁省教育厅高校基本科研项目(LJKMZ20220399); 辽宁省自然科学基金(2020-MZLH-22)

作者简介: 肖婧(1985-),女,主要从事复杂网络社区检测、进化计算和高维多目标优化方面的研究。

\*通信作者: 许小可, E-mail: xuxiaoke@foxmail.com

连边等低阶结构信息,而忽略了网络中广泛存在的高阶组织结构。真实世界网络中包含丰富的高阶组织结构,即网络中规模较小但统计意义上显著的子图,如网络模体(motifs)<sup>[9-10]</sup>等。模体广泛存在于社交网络、生物网络等真实网络中,是最普遍的高阶连接模式和交互模式,被认为是复杂网络中的基本拓扑和功能单元<sup>[9-11]</sup>。如社交网络和生物网络中通常包含大量的三角形模体<sup>[9,11]</sup>,对网络结构特性和功能特性均有重要影响。基于网络模体可揭示出复杂网络中新的高阶社区结构,具有社区内模体连接紧密、社区间模体连接稀疏的特性<sup>[9,11]</sup>。基于模体的高阶社区检测,为复杂网络中尺度结构分析提供了新视角,可展现出节点间更丰富且有意义的拓扑连接关系,从而有助于加深对网络功能特性的理解。如在线虫神经网络中,基于“bi-fan”模体可识别出额叶部分中包含 20 个神经元的运动感觉控制功能模块<sup>[9]</sup>。上述具有特殊功能特性的社区结构是传统社区检测方法难以获得的。

近年来,基于模体的复杂网络高阶社区检测逐渐受到关注<sup>[9-16]</sup>。文献[9]通过有效整合网络模体分析与社区检测,构建出基于模体的广义高阶聚类框架,并通过扩展的谱图聚类方法获得近似最优的高阶社区划分。此后,多种基于模体的社区检测算法被相继提出,包括基于模体的图嵌入方法 LinLog-Motif<sup>[13]</sup>、基于模体的边增强方法 EdMot<sup>[14]</sup>以及基于模体的标签传播方法 MWLP<sup>[15]</sup>等。上述典型方法均能挖掘、表示并利用网络模体信息,有效识别出复杂网络中的高阶社区结构。然而,基于特定高阶社区性能评价指标,并通过最优化方法获得高阶社区划分,本质上属于典型的 NP 难问题<sup>[9]</sup>且求解较为困难。现有方法中采用的启发式或谱优化方法容易陷入局部最优,从而仅获得局部最优高阶社区划分,使复杂网络中高阶社区结构分析的精确性受到制约。此外,基于模体的高阶社区检测不仅包含基于模体的高阶结构信息,还涉及到节点和连边等低阶结构信息,进一步增加了全局最优高阶社区划分检测的难度。最后,部分检测方法还需要预知社区数目<sup>[9,12]</sup>,在真实世界网络检测中难以实现。

针对上述问题,本文提出了一种基于模体的模块度优化方法以实现高阶社区检测(motif-based modularity optimization for higher-order community detection, MMHCD),致力于提高基于模体的社区划分质量。

## 1 相关工作

### 1.1 基于模体的高阶社区检测

基于模体的高阶社区检测有助于揭示复杂网络的高阶组织结构及功能特性<sup>[9]</sup>,近年来逐渐获得广泛关注,涌现出多种典型的代表性方法<sup>[9-16]</sup>。如文献[9]构建的模体传导(motif conductance)函数用于评估高阶社区结构质量,并通过扩展的谱图聚类方法获得近似最优的高阶社区划分。该研究提出的广义高阶聚类框架,广泛应用于基于模体的高阶社区检测。在此基础上,文献[12]提出了三角形模体传导函数及基于三角形模体的谱聚类算法 TECTONIC。该算法根据网络连边参与三角形模体的数目进行加权,并通过最小割三角形模体数量以获得高阶社区划分。除上述典型谱聚类算法外,文献[13]提出一种基于模体的图嵌入方法 LinLog-Motif,在基于模体的加权网络上,通过结合模体的力引导嵌入方法,将加权网络映射到低维空间,并采用  $K$  均值聚类获得高阶社区划分。文献[14]提出了一种基于边增强的模体社区检测方法 EdMot,在模体超图的前  $K$  个最大连通片上,通过添加边集强化原始网络中模体连通部分,再利用经典的 Louvain 方法<sup>[12]</sup>获取基于模体的社区结构。EdMot 能有效克服模体社区检测中存在的网络不连通及孤立节点问题。文献[15]提出了一种基于模体的加权标签传播算法 MWLP,综合考虑模体高阶结构特征与边低阶结构特征为网络连边加权,并通过标签传播获得基于模体的高阶社区结构。

上述方法虽然对模体信息的处理方式不同,但均已证明能够从复杂网络中有效检测出基于模体的高阶社区结构。然而,上述方法通过优化基于模体的社区评价函数,如模体传导函数<sup>[9,12]</sup>,以获得高阶社区结构的过程,本质上属于典型的 NP 难优化问题,求解较为困难。现有算法中采用的启发式或谱优化方法,存在容易早熟收敛的问题,难以获取全局最优的高阶社区划分。此外,现有算法在模体高阶结构信息利用过程中,容易忽略节点和连边等低阶结构信息,造成孤立节点和连通碎片。最后,部分算法需要预先设定社区数目,如 Motif-SC<sup>[9]</sup>和 TECTONIC<sup>[12]</sup>等,在真实网络环境下通常难以实现。

### 1.2 基于模块度优化的社区检测

模块度优化是最典型的复杂网络社区检测方法之一,尽管受到分辨率限制问题的影响,但由于具有精确度高、可移植性强等优势,广泛应用于真实

世界网络的社区结构发现<sup>[16-20]</sup>。模块度优化的NP难特性使其求解较为困难, 在过去几十年的研究中, 大量元启发式优化算法被采用并作为优化策略<sup>[19-20]</sup>, 以提高模块度全局最优化性能。如新兴的蝙蝠算法<sup>[21]</sup>和果蝇优化算法<sup>[22]</sup>等生物启发式算法, 以及状态转移算法<sup>[23]</sup>和火势蔓延算法<sup>[24]</sup>等自然启发式算法。元启发式优化算法由于具有基于种群的并行计算、自组织、有效结合网络结构信息及自动确定社区数目等优点, 为模块度优化提供了良好的优化策略支持。

基于元启发式算法的模块度优化方法, 已成功地用于求解传统低阶社区的检测问题, 但其极少被用于模体高阶社区的检测。分析该类方法的应用主要包括两项关键因素: 1) 高阶社区检测中的结构单元由节点(或连边)变为模体, 因此首先要解决网络中的模体结构信息的挖掘和表示, 以构建高阶模块度优化模型; 2) 利用元启发式算法进行模块度优化的过程中, 需有效结合网络中基于模体的高阶结构信息, 以及基于节点、连边的低阶结构信息, 以保证社区划分的完整性并提高检测性能。

## 2 基于模体的高阶社区检测数学描述

模体是复杂网络中规模较小但统计意义上显著的子图, 作为网络节点之间最普遍的高阶连接和交互模式, 通常对应于复杂网络中特定的功能特性<sup>[9,11]</sup>。不失一般性, 以无向无权图 $G=(V,E)$ 为例,  $|V|=n$ 代表节点集合,  $|E|=m$ 代表边集合。图 $G$ 中的单个模体可由元组 $(M,V(M))$ 表示, 其中 $M$ 为单个特定模体,  $V(M)$ 表示组成模体 $M$ 的节点集合,  $|V(M)|$ 为模体 $M$ 中包含的节点数目, 即模体 $M$ 的阶数, 如 $|V(M)|=3$ 代表 $M$ 为包含3个节点的3阶模体。图 $G$ 中模体 $M$ 的实例集合可表示为 $S(M)=\{H|H \text{ is a subgraph of } G, H \cong M\}$ , 其中 $H \cong M$ 表示图 $G$ 中与 $M$ 同构的子图<sup>[9]</sup>。

研究主要考虑无向无权网络中最普遍的8种三阶和四阶模体, 如图1所示。图1中的三阶模体包括 $M_1$ 和 $M_2$ , 四阶模体包括 $M_3 \sim M_8$ , 其中三角形模体 $M_1$ 被认为是社交网络和生物网络中的重要模体结构<sup>[9,11]</sup>, 也是现有基于模体的高阶社区检测方法中最普遍使用的模体结构<sup>[9-16]</sup>。

复杂网络中基于模体的高阶社区检测, 可看作是对传统社区检测的自然扩展<sup>[9]</sup>。不失一般性, 检测图 $G$ 中基于模体 $M$ 的高阶社区结构, 实质是将 $G$ 中的节点划分为若干社区, 并使同一社区内的模

体连接尽可能紧密, 而不同社区之间的模体连接尽可能稀疏<sup>[9-16]</sup>。基于上述定义, 图 $G$ 中基于模体 $M$ 的一个社区划分可表示为 $C=\{C_1, C_2, \dots, C_k\}$ 且 $V=\bigcup_{i=1}^k C_i$ , 其中 $k$ 代表社区数目,  $C_i$ 代表第 $i$ 个社区内的一组节点。在基于模体 $M$ 的非重叠社区划分中, 单个模体实例 $H$ 仅能隶属于一个社区, 意味着 $H$ 所涉及的所有节点隶属于同一社区, 即:

$$C_i \cap C_j = \emptyset \quad \forall i \neq j, i, j \in \{1, 2, \dots, k\}$$

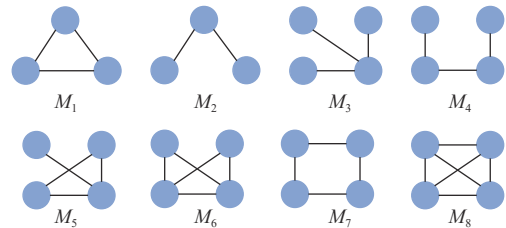


图1 无向网络中的三阶和四阶模体

## 3 加权模块度优化高阶社区检测算法

为提升基于模体的高阶社区检测性能, 提出一种基于模体的模块度优化方法MMHCD。首先, 量化并表示出网络模体的数量及分布, 将基于模体的高阶信息嵌入网络中作为节点间的权重。在此基础上, 将基于模体的高阶社区检测问题, 转化为加权网络上的模块度优化问题进行求解。其次, 为提升NP难加权模块度的全局最优化性能, 尽可能获得逼近全局最优的高阶社区划分, 一方面采用典型元启发式算法为优化策略, 逼近全局最优解; 另一方面综合利用网络中节点邻域内的低阶拓扑结构和高阶权重信息, 通过节点邻域社区修正操作和局部搜索操作, 进一步提升优化所得的种群个体质量, 辅助优化策略提升全局优化性能。

### 3.1 基于模体的加权模块度优化模型构建

为实现基于模体的高阶社区检测, 先将网络中隐含的模体数量及分布信息表示为网络节点之间的权重, 并在加权后的网络上构建模块度优化模型。

#### 3.1.1 基于模体的网络高阶信息加权

考虑到网络中任意两节点共同参与的模体实例的个数, 反映了其之间高阶连接的紧密性, 因此将其作为节点间的高阶权重。两节点共同参与的模体实例数目越多, 表示其之间基于模体的高阶连接越紧密, 因此对应权重越大, 且越有可能划分到同一社区。以无向无权图 $G=(V,E)$ 为例, 基于特定模体 $M$ 为网络加权, 并构建模块度优化模型。首先基于图 $G$ 及模体 $M$ 的拓扑结构, 可获得图 $G$ 基于



$M$ 的模体邻接矩阵  $\mathbf{W}_M=[w_{ij}]^{n \times n}$ , 其中元素  $w_{ij}$  代表节点  $i$  和  $j$  共同参与基于模体  $M$  的实例  $H$  数目:

$$w_{ij} = \sum_H \{H | i \in H, j \in H, H \cong M\} \quad (1)$$

当元素  $w_{ij}$  为 0 时, 代表节点  $i$  和  $j$  没有共同参与任何模体实例, 即使两节点之间存在连边。基于图  $G$  的模体邻接矩阵  $\mathbf{W}_M=[w_{ij}]^{n \times n}$  为图  $G$  中连边加权, 即可构建出加权网络  $G_W=(V,E,\mathbf{W}_M)$ 。如图 2 所示, 由于节点 1 和 3 共同参与两个模体实例, 包括  $\{1,2,3\}$  和  $\{1,3,4\}$ , 因此连边  $e_{13}$  权重赋值为  $w_{13}=2$ ; 节点 4 和 5 由于共同参与模体实例数目为 0, 因此连边  $e_{45}$  权重赋值为  $w_{45}=0$ 。

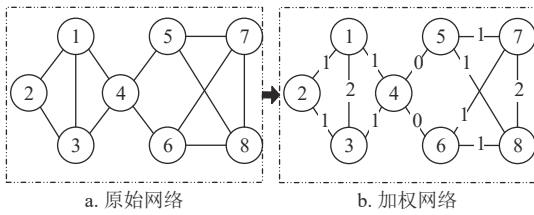


图 2 基于三角形模体  $M_1$  的网络加权示意图

上述基于节点共同参与模体数目的加权方式, 可在不丢失或虚增网络拓扑连接的前提下, 有效表示出网络中隐含的模体分布和数量信息。

### 3.1.2 加权网络上的模块度优化模型构建

在基于模体高阶信息构建的加权网络中, 为获取网络中的高阶社区结构, 需促使权重大 (高阶连接紧密) 的节点对划分到同一社区, 权重小 (高阶连接稀疏) 的节点对划分到不同社区。换言之, 同一社区内的权重分布尽可能集中, 不同社区间的权重分布尽可能稀疏, 而该种社区划分可通过加权模块度优化方法实现。此外, 由于模体高阶权重能够衡量节点间的连接紧密性, 有助于克服模块度优化中的分辨率限制问题, 从而提升检测性能。基于上述分析, 在基于模体  $M$  加权后的加权网络  $G_W=(V,E,\mathbf{W}_M)$  上构建加权模块度优化模型, 即以加权模块度函数  $Q_W$ <sup>[25]</sup> 为目标函数的全局最大化问题:

$$Q_W(C^*) = \max(Q_W(C)) \quad (2)$$

式中,  $C=\{C_1, C_2, \dots, C_k\}$ ,  $C \in \Omega$  代表搜索空间  $\Omega$  中的任一可行社区划分,  $k$  为社区数目。目标函数加权模块度  $Q_W$  定义为:

$$Q_W = \frac{1}{2|W|} \sum_{ij} \left[ w_{ij} - \frac{s_i s_j}{2|W|} \right] \delta(C_i, C_j) \quad (3)$$

式中,  $|W| = \frac{1}{2} \sum_{ij} w_{ij}$  为图  $G$  中所有连边权重之

和;  $s_i = \sum_{j \in N_i} w_{ij}$  代表节点  $i$  的强度, 即与节点  $i$  相连所有边上的权重总和;  $C_i$  和  $C_j$  分别表示节点  $i$  和  $j$  所属的社区,  $ij \in \{1, 2, \dots, k\}$ ;  $\delta(C_i, C_j)$  为罚函数, 当节点  $i$  和  $j$  属于同一社区, 其值为 1, 否则为 0。  $Q_W$  最大化时对应的社区划分  $C^*$  作为加权网络  $G_W$  的最优社区划分输出, 即原始网络  $G$  上基于模体  $M$  检测所得的高阶社区划分。

### 3.2 基于模体的加权模块度优化算法

基于上述模型, 提出基于模体的加权模块度优化算法 MMHCD, 具体流程如算法 1 所示。为尽可能逼近加权模块度优化问题的全局最优解, 一方面采用求解 NP 难优化问题高效的元启发式优化算法作为优化策略; 另一方面充分利用网络中节点邻域内的低阶结构和高阶权重信息。

#### 算法 1 MMHCD 算法框架

输入: 原始网络  $G(V,E)$ , 邻接矩阵  $\mathbf{A}$ , 模体  $M$ , 种群规模 NP, 最大迭代次数  $t_{\max}$

输出: 最优社区划分  $C^*$

基于模体  $M$  生成模体邻接矩阵  $\mathbf{W}_M$  及加权网络  $G_W$

构建初始种群  $\mathbf{P}=[X_1, X_2, \dots, X_{NP}]$

for  $t=0$  to  $t_{\max}$  do

元启发式优化算法更新种群  $\mathbf{P}$  生成  $\mathbf{P}_{\text{OPT}}$

邻域社区修正 LH-NCM 操作更新  $\mathbf{P}_{\text{OPT}}$  生成

$\mathbf{P}_{\text{WNCM}}$

邻域社区局部搜索 NCLS 操作更新  $\mathbf{P}_{\text{WNCM}}$

生成  $\mathbf{P}_{\text{NCLS}}$

基于种群  $\mathbf{P}_{\text{NCLS}}$  构建下一代初始种群  $\mathbf{P}$

end for

输出最大  $Q_W$  值对应种群个体  $X_{\text{best}}$  作为最优社区划分  $C^*$

在 MMHCD 算法中, 首先根据特定模体  $M$  对原始网络  $G$  加权, 获得模体邻接矩阵  $\mathbf{W}_M$  及加权网络  $G_W$ ; 其次构建规模为 NP 的初始种群  $\mathbf{P}$ , 种群个体代表网络社区划分, 采用广泛使用的社区标签编码策略<sup>[19]</sup> 以自动确定社区数目; 算法主循环部分包括元启发式种群优化、节点邻域社区修正和邻域社区局部搜索 3 项操作, 用于种群  $\mathbf{P}$  进化迭代更新。算法迭代循环至最大迭代次数  $t_{\max}$ , 输出种群中加权模块度  $Q_W$  最大个体  $X_{\text{best}}$  作为最优社区划分  $C^*$ , 即原始网络  $G$  上检测所得的高阶社区划分。以下对 MMHCD 算法主循环部分的 3 项操作及算法时间复杂度进行说明。

### 3.2.1 基于元启发式算法的模块度优化策略

共生生物搜索 (symbiotic organisms search, SOS)<sup>[25]</sup> 是一种典型的生物启发式优化算法, 通过模拟生态系统中生物体之间的 3 种典型共生交互行为, 即互利共生、共栖和寄生, 实现种群个体的更新和优化。该算法具有精度高、控制参数少、稳定性好等优点, 在多种 Benchmark 优化函数测试集上的性能表现显著优于遗传算法、粒子群优化和差分进化算法等经典算法<sup>[25-28]</sup>。因此, 本文采用该算法进行加权模块度优化, 提升获取全局最优解概率。SOS 算法操作流程如算法 2 所示。

算法 2 SOS 算法框架

输入: 初始种群  $P=[X_1, X_2, \dots, X_{NP}]$ , 最大迭代次数  $t_{max}$

输出: 优化后种群  $P_{OPT}$  和最优解  $X^*$

构建初始种群  $P=[X_1, X_2, \dots, X_{NP}]$

for  $t=0$  to  $t_{max}$  do

    互利共生操作更新种群  $P$  生成  $P_{MUTU}$

    共栖操作更新种群  $P_{MUTU}$  生成  $P_{COMM}$

    寄生操作更新种群  $P_{COMM}$  生成  $P_{PARA}$

    基于种群  $P_{PARA}$  构建下一代初始种群  $P$

end for

输出优化后种群  $P_{OPT}$  及最优解  $X^*$

对包含 NP 个个体的初始种群  $P$ , 每一代迭代优化过程中, 通过互利共生、共栖和寄生 3 项操作更新种群, 达到最大迭代次数时输出优化后种群  $P_{OPT}$  及最优解  $X^*$ 。3 项操作具体描述如下。

1) 互利共生操作: 该操作模拟了自然界中生物体间的共生交互关系。种群个体通过两两共生交互共享进化信息, 在当前最优个体引导下, 增强在生态系统中的生存优势。对于种群中的目标个体  $X_i$ ,  $i \in \{1, 2, \dots, NP\}$ , 随机选择另一个体  $X_j (X_j \neq X_i)$  进行信息交互, 如式 (4) 和 (5) 所示, 生成新个体  $X_{i_{new}}$  和  $X_{j_{new}}$ :

$$\begin{cases} X_{i_{new}} = X_i + \text{rand}(0, 1) \times (X_{best} - MV \times BF_1) \\ X_{j_{new}} = X_j + \text{rand}(0, 1) \times (X_{best} - MV \times BF_2) \end{cases} \quad (4)$$

$$MV = 0.5 \times (X_i + X_j) \quad (5)$$

式中,  $\text{rand}(0, 1)$  为  $(0, 1)$  间的随机数;  $X_{best}$  为当前种群的最优个体; 获益因子  $BF_1$  和  $BF_2$  随机取值 1 或 2;  $MV$  为互利向量, 代表个体  $X_i$  和  $X_j$  之间的交互信息。当生成的新个体  $X_{i_{new}}$  和  $X_{j_{new}}$  具有相对较优的适应度值时, 将分别取代当前种群中的原个体  $X_i$  和  $X_j$ 。

2) 共栖操作: 该操作模拟了自然界中生物体间的共栖关系。生物体通过共栖交互获得其他生物体的进化信息, 在当前最优个体引导下, 提升搜索到较优区域的概率。对于种群中的目标个体  $X_i$ ,  $i \in \{1, 2, \dots, NP\}$ , 随机选择另一种群个体  $X_j (X_j \neq X_i)$ , 按式 (6) 共栖交互生成  $X_{i_{new}}$ :

$$X_{i_{new}} = X_i + \text{rand}(-1, 1) \times (X_{best} - X_j) \quad (6)$$

式中,  $\text{rand}(-1, 1)$  为  $(-1, 1)$  间的随机数;  $X_{best}$  和  $X_j$  代表了来自种群最优个体和随机交互个体的进化信息。当生成的新个体  $X_{i_{new}}$  适应度值优于  $X_i$  时, 则替代当前种群中的  $X_i$ 。

3) 寄生操作: 寄生操作模拟了生态系统中寄生者和宿主之间的寄生和博弈关系, 实现种群个体的随机化变异和优胜劣汰。对于种群中的目标个体  $X_i$ ,  $i \in \{1, 2, \dots, NP\}$ , 随机化其若干维分量构造寄生个体  $X_{para}$ ; 并随机选择另一种群个体  $X_j$  作为  $X_{para}$  的宿主, 在  $X_{para}$  和  $X_j$  的竞争中保留具有较优适应度值的个体, 将另一个体从种群中删除。特别地, 引入寄生个体的随机化成分, 有助于增强种群个体多样性, 避免进化后期陷入局部最优。

SOS 算法通过上述 3 项操作, 模拟生态系统中生物体间的多种典型共生交互行为, 实现种群个体进化信息交互, 优化子代种群  $P_{OPT}$  个体性能, 增强算法搜索到全局最优解的概率, 从而提升高阶社区划分的质量。

### 3.2.2 融合低阶与高阶信息的节点邻域社区修正

尽管元启发式算法具有较好的全局最优化能力, 但种群优化过程中仍会出现部分不合理的社区划分。如连接紧密性较弱的节点分配在同一社区, 或连接紧密性较强的节点分配在不同社区。分析该现象的主要原因为元启发式算法在优化过程中的随机性搜索特性所致。

为解决上述问题, 提出一种融合低阶与高阶信息的节点邻域社区修正 (lower-order and higher-order based neighbor community modification, LH-NCM) 操作。核心思想为通过综合利用节点邻域范围内的低阶拓扑结构信息和高阶权重信息, 将节点以较大概率调整到连接较为紧密的邻居社区中, 从而修正节点邻域范围内不合理的社区划分, 提升社区划分质量。LH-NCM 操作对网络高阶和低阶信息的利用, 主要体现在依据目标节点与各邻居社区内节点之间的权重总和, 度量与各邻居社区之间连接紧密程度。由于权重总和不仅体现了基于模体的

高阶连接强弱,也同时反应了低阶边连接强弱及邻居节点规模,因此能够综合体现目标节点与邻居社区之间的高阶和低阶连接紧密程度。

基于上述分析,设计 LH-NCM 操作主要包括目标节点选择及邻域社区隶属度计算、节点社区合理性判定以及节点社区迁移 3 部分。

1) 针对种群中的每个社区划分  $X$ , 选择各社区内的边界节点为目标节点集合  $V_{obj}$ , 即与所属社区外其他社区有边连接的节点集合。针对每个目标节点  $i \in V_{obj}$  及其任意邻域社区  $C$ , 根据节点  $i$  与社区  $C$  内邻居节点之间的总权重, 计算对社区  $C$  的隶属度:

$$U(i, C) = \sum_{j \in C \cap N_i} w_{ij} \quad (7)$$

式中,  $j$  代表社区  $C$  内节点  $i$  的任意邻居节点;  $N_i$  代表节点  $i$  的邻域。通过隶属度  $U$  计算, 度量节点  $i$  与各邻域社区之间的局部连接紧密程度。

2) 根据隶属度计算, 判定集合  $V_{obj}$  内任意目标节点当前所属社区的合理性, 选择对当前社区隶属度小于邻域内所有社区隶属度平均值的节点为不合理节点。若目标节点  $i$  的邻域社区集合表示为  $NC_i$  且  $|NC_i| \leq |N_i|$ , 则所有邻域社区的平均隶属度为

$$\sum_{C \in NC_i} U(i, C) / |NC_i|。$$

3) 对于社区归属不合理的任意目标节点  $i$ , 以其对各邻居社区的归一化隶属度  $U$  为随机选择概率, 从而以较大概率迁移至连接相对较为紧密的邻居社区, 由此提升局部社区划分质量。如图 3 所示, 4 节点经过邻域社区修正后, 从原社区 (红色节点表示) 迁移至邻域社区 (蓝色节点表示), 对应社区划分的加权模块度值增长 0.18, 表明局部社区划分质量得到提升。

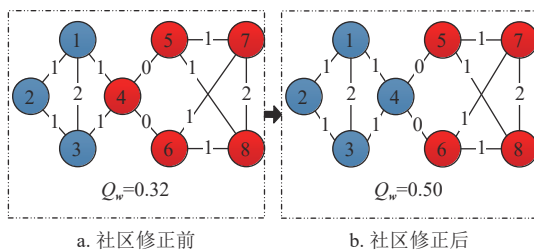


图 3 邻域社区修正结果

### 3.2.3 基于邻域社区的节点局部搜索

尽管通过融合网络局部低阶与高阶信息, 能够在节点邻域范围内修正不合理的社区划分, 提升种群个体质量。然而, 在算法 MMHCD 进化后期,

种群中通常会出现大量高质量且相似的局部最优社区划分。为了促使算法 MMHCD 跳出局部最优, 设计一种基于邻域社区的节点局部搜索 (neighbor community based local search, NCLS) 操作。不同于现有大多数依据邻居节点规模或特性的局部搜索操作, NCLS 操作在目标节点邻域范围内, 搜索模块度增益最大的邻域社区, 并作为局部最优社区。这种基于邻域社区的局部搜索, 可避免邻居节点特性引导与模块度优化目标之间的不一致, 由此快速增强算法在模块度优化中的开采能力, 提升收敛到全局最优的概率及速度。

基于上述思想设计 NCLS, 主要包括种群个体及边界节点选择、邻域社区搜索及局部最优社区迁移 3 个步骤。首先, 为保持种群个体的多样性及探索能力, 防止算法早熟收敛, 仅选择模块度值较为优异的前 25% 个体为目标个体集合。此外, 对集合中任意目标个体, 仅选择对应社区划分中的边界节点 (即与非所属社区间存在边连接的节点) 为目标节点集合; 其次, 目标节点在邻域社区内搜索, 选择对应增量模块度最大的邻居社区为目标社区。基于标准加权模块度  $Q_w$ , 推导节点从原社区  $C_{orig}$  迁移至目标社区  $C_{obj}$  时, 对应增量加权模块度  $\Delta Q_w$ :

$$\Delta Q_w = \frac{1}{2|W|} \left[ 2(\Delta W_{C_{obj}} - \Delta W_{C_{orig}}) - \frac{\Delta d(\Delta d + S_{C_{obj}} - S_{C_{orig}})}{W} \right] \quad (8)$$

式中,  $|W|$  代表网络边权重的总和;  $S_{C_{orig}}$  和  $S_{C_{obj}}$  分别代表原社区和目标社区内节点强度总和;  $\Delta d$  代表目标节点强度;  $\Delta W_{C_{orig}}$  和  $\Delta W_{C_{obj}}$  分别代表节点迁移前后对应社区的权重变化。最后, 选择对应增量加权模块度  $\Delta Q_w$  最大的邻域社区为局部最优社区, 并迁移目标节点至该目标社区。

### 3.2.4 算法复杂度分析

MMHCD 算法的计算复杂度主要取决于循环迭代中的 3 项操作, 包括基于 SOS 算法的种群个体更新、基于 LH-NCM 策略的节点邻域社区修正以及基于邻域社区的节点局部搜索操作 NCLS。不失一般性, 在包含  $n$  个节点和  $m$  条边的网络  $G$  中, 上述 3 项操作所需时间复杂度分别  $O(5n)$ 、 $O(n\langle d \rangle)$  和  $O(n\langle d \rangle)$ , 其中  $\langle d \rangle$  代表网络  $G$  中节点的平均度。此外, MMHCD 算法中最耗时部分为种群个体适应度值评价, 即加权模块度  $Q_w$  计算, 根据  $Q_w$  定义<sup>[18]</sup> 可知个体加权模块度计算时间复杂度为  $O(n^2)$ 。由于网络节点数目  $n$  较大时可认为  $n \gg \langle d \rangle$ , 因此 MMHCD 算法的时间复杂度可简化为  $O(n^2)$ 。



## 4 实验结果及分析

对提出 MMHCD 算法的性能进行了实证验证和分析, 与现有基于模体的典型高阶社区检测典型算法进行对比, 并对 MMHCD 算法中模体的影响进行了分析。

### 4.1 实验设置

1) 对比算法: 选择4种代表性的基于模体的高阶社区检测算法, 包括基于模体的谱图聚类算法 Motif-SC<sup>[9]</sup>、基于模体的网络嵌入算法 LinLog-Motif<sup>[13]</sup>、基于边增强的模体社区检测方法 EdMot<sup>[14]</sup> 以及基于模体的加权标签传播算法 MWLP<sup>[15]</sup>。除上述算法外, 还有基于典型的元启发式模块度优化算法 DECD<sup>[29]</sup> 构建的基于模体的加权模块度优化算法 Motif-DECD。

上述算法的参数设置均采用原文献中的推荐值, 其中 Motif-SC 算法中需要预设真实社区数目; EdMot 选用经典 Louvain 算法进行社区检测, 设置最大连通片数目  $K$  在  $[1,3]$  范围内; MWLP 算法中的权衡参数  $\lambda$  在  $[0,0.3]$  范围内取值; Motif-DECD 算法中变异因子  $F$  取值为 0.9, 交叉因子 CR 取值为 0.3, 阈值  $\eta$  为 0.35; Motif-DECD 与 MMHCD 算法均设置种群规模 NP 为 100, 最大迭代次数  $t_{\max}$  为 200。

2) 真实世界网络: 从 KONECT 项目<sup>[30]</sup> 和网络数据存储器<sup>[31]</sup> 中选用 10 种不同类型及不同规模的典型真实世界网络, 具体信息如表 1 所示。网络类型包括社交网络、生物网络、工程及通信网络等。节点规模在 1 000 以上的网络占比 50%, 其中规模最大的网络 PGP 包含 10 680 个节点和 24 316 条连边。

表 1 真实世界网络信息

数据集	节点数/个	边数/条	网络平均度	网络类型
Karate	34	78	4.69	社交
Macaque	47	505	13.32	生物
Dolphins	62	159	5.13	社交
Polbooks	105	441	8.40	社交
Football	115	613	10.66	社交
Email	1 133	5 451	4.81	社交
Cora	2 708	5 429	3.90	科学引文
Facebook	2 888	2 981	2.06	社交
PowerGrid	4 941	6 594	2.67	工程
PGP	10 680	24 316	4.55	通信

3) 人工合成网络: 选择两种典型人工合成网络<sup>[30]</sup>, 包括 Lancichinetti-Fortunato-Radicchi (LFR) 和 Girvan-Newman(GN), 对各算法性能进行评估。

在生成的 10 个 LFR 网络中, 节点规模均为 1 000, 混合参数  $\mu$  以 0.1 为间隔在  $[0,0.9]$  范围内取值, 网络社区规模取值范围为  $[20,50]$ , 节点平均度和最大度设置为 10 和 50。此外, 在生成的 10 个 GN 网络中, 包含 128 个节点和 4 个社区, 参数  $z_{\text{out}}$  以 1 为间隔在  $[0,9]$  范围内取值。

4) 性能评价指标: 采用加权模块度  $Q_w$  和归一化互信息 NMI<sup>[32]</sup> 为高阶社区划分的评价指标, 分别用于评估各算法检测结果的质量性和精确性。加权模块度函数<sup>[1,25]</sup> 见式 (3), 数值越大代表检测所得的高阶社区结构越显著。归一化互信息 NMI 为:

$$\text{NMI}(A, B) = \frac{-2 \sum_{i=1}^{k_{C_A}} \sum_{j=1}^{k_{C_B}} N_{ij} \log\left(\frac{N_{ij}n}{N_i N_j}\right)}{\sum_{i=1}^{k_{C_A}} N_i \log\left(\frac{N_i}{n}\right) + \sum_{j=1}^{k_{C_B}} N_j \log\left(\frac{N_j}{n}\right)} \quad (9)$$

式中,  $k_{C_A}$  和  $k_{C_B}$  分别代表划分  $A$  和  $B$  中的社区数量;  $n$  是节点数量;  $N$  为混淆矩阵, 每项元素  $N_{ij}$  代表同时隶属于  $A$  中第  $i$  个社区和  $B$  中第  $j$  个社区的节点数目;  $N_i$  是第  $i$  行元素的和, 代表划分  $A$  的第  $i$  个社区中的节点数;  $N_j$  是第  $j$  列元素的和, 代表划分  $B$  的第  $j$  个社区中的节点数。NMI 取值范围为  $[0,1]$ , 数值越大代表检测结果越接近真实社区结构。

### 4.2 人工合成网络性能测试

首先, 在生成的 GN 人工合成网络集合上, 对提出的 MMHCD 算法与 4 种典型对比算法 MWLP, EdMot, LinLog-Motif, Motif-SC 进行性能测试。为保证实验的公平性, 同一测试网络上各算法均采用相同的模体结构, 选择  $M_1 \sim M_8$  中使各算法性能达到相对最优的单个模体。所有算法在 GN 网络上的实验结果如图 4 所示, 图中数据点代表各算法在 10 个 GN 网络上独立运行 20 次所得的社区划分 NMI 的平均值。由图可知 MMHCD 算法在所有 GN 网络上相比较于其他典型算法, 均能获得相对较优的检测结果, 在精确性和稳定性上表现出一定优势。当  $z_{\text{out}} \leq 5$  网络社区结构较为清晰时, MMHCD、Motif-SC 和 Motif-DECD 算法均能够稳定地检测得到全局最优社区划分, 对应 NMI 平均值均达到 1.0。EdMot 和 LinLog-Motif 算法也能够获得近似 NMI=1.0 的检测结果。然而, 当  $z_{\text{out}} \geq 5$  对应网络社区结构较为模糊时, 各算法均表现出了不同程度的性能恶化趋势, 且性能差异逐渐凸显。与其他算

法相比, MMHCD 算法依然能够获得相对最优的检测精度。

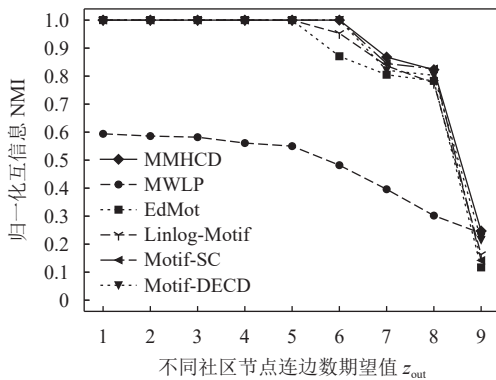


图 4 GN 网络上 MMHCD 与对比算法的 NMI 检测结果

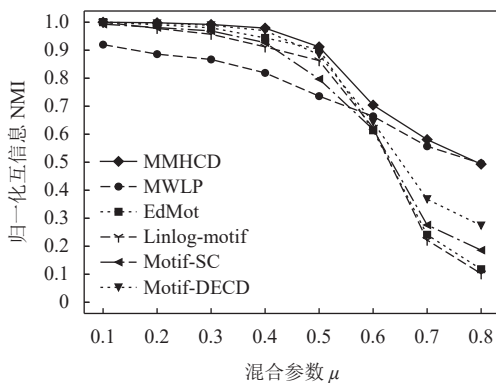


图 5 LFR 网络上 MMHCD 与对比算法的 NMI 检测结果

其次, 在规模为 1 000 的 LFR 人工合成网络集合上, 对各算法进行测试及对比分析。相同测试网络上各算法均采用相同的模体结构, 选择  $M_1 \sim$

$M_8$  中使各算法性能达到相对最优的单个模体。所有算法在 LFR 网络上的实验结果如图 5 所示, 图中数据点代表各算法在 10 个 LFR 网络上独立运行 20 次所得社区划分 NMI 值的平均值。可以看出, MMHCD 算法在 LFR 网络上依然能够获得相对最优的 NMI, 与其他典型算法相比具有较明显的性能优势。当  $\mu \leq 0.3$  时, MMHCD 算法检测所得社区划分 NMI 值均接近 1.0, 除 MWLP 算法外其他对比算法也均能获得较为精确的检测结果。当  $0.4 \leq \mu \leq 0.7$  时, 各算法检测精确性逐渐下降, 但 MMHCD 算法均能获得相对最优的检测结果。尽管在  $\mu=0.8$  时 MMHCD 算法未获得相对最高的 NMI 值, 但与 MWLP 算法获得的最优检测结果精度差仅  $3 \times 10^{-3}$ 。

以上 GN 和 LFR 网络上的实验结果表明, MMHCD 算法能在人工合成网络上检测得到较精确的社区结构, 且相较于其他基于模体的典型代表性算法, 在精确性和稳定性上能表现出一定优势。

### 4.3 真实世界网络性能测试

将 MMHCD 算法在表 1 所示 10 个不同类型和规模的真实世界网络上进行测试, 并与 5 种典型算法进行对比, 实验结果如表 2 所示。为保证实验的公平性, 同一测试网络上各算法均采用相同的模体结构, 选择  $M_1 \sim M_8$  中使各算法性能达到相对最优的单个模体。表中数据为各算法在每个网络上独立运行 20 次, 所得社区划分对应加权模块度  $Q_W$  的平均值和标准差。同一测试网络上最优检测结果用黑体加粗表示。

表 2 MMHCD 与基于模体的算法在真实世界网络上的检测结果

数据集	模体	MWLP	EdMot	LinLog-motif	Motif-SC	Motif-DECD	MMHCD
Karate	$M_1$	0.362( $1.8 \times 10^{-2}$ )	0.456( $2.63 \times 10^{-2}$ )	<b>0.484(<math>1 \times 10^0</math>)</b>	<b>0.484(<math>1 \times 10^0</math>)</b>	<b>0.484(<math>1 \times 10^0</math>)</b>	<b>0.484(<math>1 \times 10^0</math>)</b>
Macaque	$M_1$	0.061( $1.53 \times 10^{-2}$ )	0.258( $1 \times 10^0$ )	0.256( $1.19 \times 10^{-4}$ )	0.244( $1 \times 10^0$ )	0.259( $2.1 \times 10^{-3}$ )	<b>0.265(<math>1 \times 10^0</math>)</b>
Dolphins	$M_1$	0.478( $1.3 \times 10^{-2}$ )	0.637( $5.98 \times 10^{-3}$ )	0.641( $1 \times 10^0$ )	0.637( $1 \times 10^0$ )	<b>0.647(<math>1 \times 10^0</math>)</b>	<b>0.647(<math>1 \times 10^0</math>)</b>
Polbooks	$M_1$	0.323( $1.6 \times 10^{-2}$ )	0.544( $5.74 \times 10^{-4}$ )	0.546( $1 \times 10^0$ )	0.544( $1 \times 10^0$ )	0.545( $5 \times 10^{-3}$ )	<b>0.548(<math>1 \times 10^0</math>)</b>
Football	$M_1$	0.388( $1.5 \times 10^{-2}$ )	0.847( $4.87 \times 10^{-3}$ )	<b>0.853(<math>1 \times 10^0</math>)</b>	0.852( $1 \times 10^0$ )	<b>0.853(<math>1 \times 10^0</math>)</b>	<b>0.853(<math>1 \times 10^0</math>)</b>
Email	$M_1$	0.423( $4 \times 10^{-4}$ )	0.673( $5.11 \times 10^{-3}$ )	0.655( $6.66 \times 10^{-4}$ )	<b>0.677(<math>3 \times 10^{-4}</math>)</b>	0.658( $1.9 \times 10^{-3}$ )	0.673( $1.1 \times 10^{-3}$ )
Cora	$M_1$	0.638( $8.18 \times 10^{-3}$ )	0.881( $2.46 \times 10^{-2}$ )	0.879( $5.02 \times 10^{-3}$ )	0.863( $2.15 \times 10^{-4}$ )	0.816( $6.5 \times 10^{-2}$ )	<b>0.890(<math>2.34 \times 10^{-3}</math>)</b>
Facebook	$M_1$	0.439( $1.5 \times 10^{-2}$ )	0.386( $1 \times 10^0$ )	0.391( $1 \times 10^0$ )	0.235( $1 \times 10^{-3}$ )	0.421( $1.4 \times 10^{-2}$ )	<b>0.477(<math>1 \times 10^0</math>)</b>
PowerGrid	$M_1$	0.727( $1.7 \times 10^{-2}$ )	0.908( $2.94 \times 10^{-3}$ )	0.920( $1 \times 10^0$ )	0.619( $1.5 \times 10^{-2}$ )	0.817( $1.3 \times 10^{-2}$ )	<b>0.928(<math>1 \times 10^0</math>)</b>
PGP	$M_1$	0.433( $2.53 \times 10^{-2}$ )	<b>0.803(<math>1.17 \times 10^{-3}</math>)</b>	0.755( $1 \times 10^0$ )	0.748( $4.68 \times 10^{-3}$ )	0.767( $2 \times 10^{-3}$ )	0.798( $1 \times 10^0$ )
Friedman Rank		6	4	3	5	2	1

表 2 的实验结果表明, MMHCD 算法能够在 80% 的测试网络上获得最优的  $Q_W$  平均值, 具有相对较好的质量性和稳定性。在规模较小的网络上,

MMHCD 算法均能够获得最优的检测性能, Motif-DECD 和 LinLog-Motif 算法同样表现优异。当网络规模相对较大时, 不同算法之间的性能差异增大,



MMHCD 算法依然能够获得相对最优的检测结果。在 Email 网络和 PGP 网络中, Motif-SC 与 EdMot 算法分别取得了最佳检测效果, 但其  $Q_w$  平均值均与 MMHCD 算法较为接近。

采用典型的 Friedman 统计检验方法<sup>[33]</sup>, 对各算法在真实世界网络上的检测结果进行测试, 结果表明 MMHCD 算法在检测结果的质量性上性能优势显著。且 MMHCD 算法与 Motif-DECD 算法的性能表现较为接近。这一现象表明基于模体的模块度优化方法能有效检测出基于模体的社区结构, 且在社区划分的质量性上相较于其他算法具有一定优势。此外, 二者的性能差异也表明网络高阶及低阶结构信息的利用, 有助于提升加权模块度的优化性能, 从而获得更加高质量的高阶社区划分。

#### 4.4 MMHCD 算法中模体的影响分析

考虑到 MMHCD 算法采用基于模体的模块度优化实现高阶社区检测, 本节分析模体使用不同类型的网络模体对 MMHCD 算法检测性能的影响。基于 MMHCD 算法框架, 设计采用  $M_1 \sim M_8$  不同类型模体的算法, 并命名为 MMHCD-M1 至 MMHCD-M8。此外, 设计基于低阶边结构信息的模块度优化算法 MMHCD-None, 忽略 MMHCD 算法中模体高阶信息的影响, 包括基于模体的网络加权、模块度优化函数  $Q$  以及节点邻域社区修正中边权重的影响。

为分析模体使用不同类型模体对 MMHCD 算法检测精确性的影响, 将算法 MMHCD-M1 至 MMHCD-M8 和 MMHCD-None 共 9 种算法, 在生成的节点规模为 1 000 的 LFR 网络上进行测试, 实验结果如图 6 所示。图中数据点代表各算法在每个 LFR 网络上独立运行 20 次所得 NMI 的平均值。

从图 6 中可以发现, 在 LFR 网络社区结构逐渐模糊的过程中, 9 种采用不同模体的 MMHCD 算法表现出了不同的性能趋势。首先, 与依赖低阶边结构信息的 MMHCD-None 算法相比, 基于模体高阶结构信息的 MMHCD-M1 至 MMHCD-M8 算法, 在网络结构较为清晰的情况下 ( $0 \leq \mu \leq 0.6$ ) 优势并不明显。相反, 当社区结构较为模糊时 ( $0.6 \leq \mu \leq 0.9$ ), 采用模体结构信息, 能够促使 MMHCD 算法获取更高的 NMI 值, 即达到相对更高的检测精度。其次, 不同类型模体造成了 MMHCD 算法在检测性能上的差异。MMHCD-M1 和 MMHCD-M6 算法相较于其他算法, 始终能获得较高的 NMI 值, 表现出良好的精确性, 尤其

在网络社区结构较为模糊的情况下。此外, MMHCD-M8 算法也表现出了良好的稳定性, 尽管在  $0.4 \leq \mu \leq 0.6$  的网络上 NMI 值下降明显, 但在  $\mu \geq 0.7$  时获得了相对最优的检测精度。

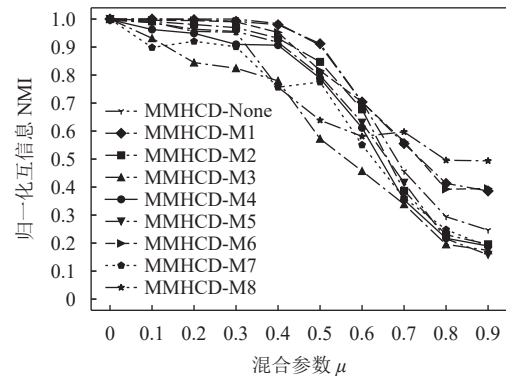


图 6 LFR 网络上不同模体的 MMHCD 算法 NMI 检测结果

上述实验结果表明, 使用不同的三阶和四阶模体结构, 对模块度优化算法 MMHCD 的社区检测精确性有一定影响。在网络社区结构较为模糊的情况下, 通过采用适当的模体结构, 如  $M_1$ 、 $M_6$  和  $M_8$  等边连接较为紧密的模体, 能够促进 MMHCD 算法的社区检测精确性提升。

## 5 结束语

在典型人工合成网络和 10 种不同规模及特性的真实世界网络上, 对 MMHCD 算法的社区检测性能进行了实验测试, 并与 Motif-SC, LinLog-Motif, EdMot 和 MWLP 等基于模体的典型高阶社区检测算法进行对比分析。实验结果表明, 相较于其他典型算法, MMHCD 算法能够在 GN 和 LFR 网络上获得更加精确和稳定的检测结果, 且在真实世界网络上同样表现出相对较好的质量性和稳定性。此外, MMHCD 算法无需预知网络真实社区数目, 且所需控制参数相对较少。

本研究提升了基于模体的网络社区检测质量, 拓展了模体理论的应用场景, 有助于加深对网络结构和功能特性的理解。未来研究将扩展到有向网络和符号网络等, 使提出的检测方法能适应不同类型的网络应用需求, 获得基于模体的高质量社区结构。

## 参考文献

- [1] NEWMAN M E J. Community structure in networks[J]. *The European Physical Journal B*, 2004, 38(2): 321-330.
- [2] BEDI P, SHARMA C. Community detection in social networks[J]. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2016, 6(3): 115-135.

- [3] FORTUNATO S, HRIC D. Community detection in networks: A user guide[J]. *Physics Reports*, 2016, 659: 1-44.
- [4] LUO J W, WU J, YANG W Y. A relationship matrix resolving model for identifying vital nodes based on community in opportunistic social networks[J]. *IEEE Transactions on Emerging Telecommunications Technologies*, 2021, 33(1): e4389.
- [5] ZHU H, JIN W, ZHOU J, et al. Nodal memberships to communities of functional brain networks reveal functional flexibility and individualized connectome[J]. *Cerebral Cortex*, 2021, 31(11): 5090-5106.
- [6] PENG X L, SMALL M, XU X J. Temporal prediction of epidemic patterns in community networks[J]. *New Journal of Physics*, 2013, 15(11): 113033.
- [7] JAVED M, YOUNIS M, LATIF S, et al. Community detection in networks: A multidisciplinary review[J]. *Journal of Network and Computer Applications*, 2018, 108: 87-111.
- [8] MITTAL R, BHATIA M. Classification and comparative evaluation of community detection algorithms[J]. *Archives of Computational Methods in Engineering*, 2021, 28(3): 1417-1428.
- [9] BENSON A R, GLEICH D F, LESKOVEC J. Higher-Order-Organization of complex networks[J]. *Science*, 2016, 353(6295): 163-166.
- [10] MILO R, SHEN-ORR S, ITZKOVITZ S, et al. Network motifs: Simple building blocks of complex networks[J]. *Science*, 2002, 298(5594): 824-827.
- [11] HUANG J Y, HOU Y, LI Y S. Efficient community detection algorithm based on higher-order structures in complex networks[J]. *Chaos*, 2020, 30(2): 023114.
- [12] TSOURAKAKIS C E, PACHOCKI J, MITZENMACHER M. Scalable motif-aware graph clustering[C]//Proc of the 26th Int Conf on World Wide Web. New York: ACM, 2017: 1451-1460.
- [13] LIM S, LEE J G. Motif-Based embedding for graph clustering[J]. *Journal of Statistical Mechanics Theory and Experiment*, 2016, 12(12): 123401.
- [14] LI P Z, HUANG L, WANG C D, et al. EdMot: An edge enhancement approach for motif-aware community detection[C]//Proc of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York: ACM, 2019: 479-487.
- [15] LI P Z, HUANG L, WANG C D, et al. Community detection by motif-aware label propagation[J]. *ACM Trans on Knowledge Discovery from Data*, 2020, 14(2): 1-19.
- [16] CAO J, BU Z, GAO G, et al. Weighted modularity optimization for crisp and fuzzy community detection in large-scale networks[J]. *Physica A: Statistical Mechanics and Its Applications*, 2016, 462: 386-395.
- [17] HAQ N F, MORADI M, WANG Z J, et al. Community structure detection from networks with weighted modularity[J]. *Pattern Recognition Letters*, 2019, 122: 14-22.
- [18] NEWMAN M E J. Analysis of weighted networks[J]. *Physical Review E*, 2004, 70(5): 056131.
- [19] PIZZUTI C. Evolutionary computation for community detection in networks: A review[J]. *IEEE Transactions on Evolutionary Computation*, 2017, 22(3): 464-483.
- [20] ATTEA B A, ABBOOD A D, HASAN A A, et al. A review of heuristics and metaheuristics for community detection in complex networks: Current usage, emerging development and future directions[J]. *Swarm and Evolutionary Computation*, 2021, 63: 100885.
- [21] SONG A, LI M B, DING X H, et al. Community detection using discrete bat algorithm[J]. *Iaeng International Journal of Computer Science*, 2016, 43(1): 37-43.
- [22] LIU Q, ZHOU B, LI S D, et al. Community detection utilizing a novel multi-swarm fruit fly optimization algorithm with hill-climbing strategy[J]. *Arabian Journal for Science and Engineering*, 2015, 41(3): 807-826.
- [23] ZHOU X J, YANG K, XIE Y F, et al. A novel modularity-based discrete state transition algorithm for community detection in networks[J]. *Neurocomputing*, 2019, 334: 89-99.
- [24] PATTANAYAK H S, SANGAL A L, VERMA H K. Community detection in social networks based on fire propagation[J]. *Swarm and Evolutionary Computation*, 2019, 44: 31-48.
- [25] CHENG M Y, PRAYOGO D. Symbiotic organisms search: A new metaheuristic optimization algorithm[J]. *Computers & Structures*, 2014, 139(7): 98-112.
- [26] AL-SHARHAN S, OMRAN M G H. An enhanced symbiosis organisms search algorithm: An empirical study[J]. *Neural Computing & Applications*, 2016, 29(11): 1025-1043.
- [27] AYALA H, KLEIN C, MARIANI V, et al. Multi-Objective symbiotic search algorithm approaches for electromagnetic optimization[J]. *IEEE Trans on Magnetics*, 2017, 53(6): 7205504.
- [28] YU V F, REDI P, RUSKARTINA E, et al. Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem[J]. *Applied Soft Computing*, 2017, 52: 657-672.
- [29] JIA G B, CAI Z X, MUSOLESI M, et al. Community detection in social and biological networks using differential evolution[C]//Proc of the Int Conf on Learning and Intelligent Optimization. Berlin: Springer, 2012: 71-85.
- [30] KUNEGIS J. Konect: The koblenz network collection [EB/OL]. [2022-04-05]. <http://konect.cc/networks>.
- [31] ROSSI R A, AHMED N K. An interactive scientific network data repository[EB/OL]. [2022-04-05]. <https://networkrepository.com/index.php>.
- [32] LANCICHINETTI A, FORTUNATO S, RADICCHI F. Benchmark graphs for testing community detection algorithms[J]. *Physical Review E*, 2008, 78(4): 046110.
- [33] JOAQUIN D, SALVADOR G, DANIEL M, et al. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms[J]. *Swarm and Evolutionary Computation*, 2011, 1(1): 3-18.