arXiv:2301.09125v1 [cs.SI] 22 Jan 2023

选择适用于不相交社区检测的并行标签传播算法

Subhajit Sahu 安全、理论与算法研究中心（CSTAR）
IIIT海得拉巴，印度 - 500 032
subhajit.sahu@research.iiit.ac.in

摘要

社区检测是网络分析中的一项重要任务，有助于识别网络中的群组和模式。要在合理的时间内分析大规模网络，必须采用高速社区检测算法。研究人员在开发高速社区检测算法方面做出了重要贡献，尤其是在基于标签传播的不相交社区检测领域。这些算法已被证明能够在合理的时间内高效分析大规模网络。然而，我们有必要评估现有方法的性能和准确性，以确定哪种算法最适合特定类型的网络和具体的研究问题。本报告研究了 RAK、COPRA 和 SLPA 这三种基于标签传播的静态社区发现技术。我们在实现每种算法的单线程和基于 OpenMP 的多线程变体时，仔细考察其细节，并进行必要的调整或优化，以确定合适的参数值。研究发现，RAK算法在容差为 0.05 时表现良好；采用 12 个线程的基于 OpenMP 的严格 RAK 比顺序非严格 RAK 快 6.75 倍。对于道路网络，COPRA算法在使用单个标签时表现良好；对于其他类别的图，最大标签数设为 4 至 16 时表现良好。随着内存容量增加，SLPA 算法的表现有所提升，但总体而言，其投资回报并不理想。推荐使用 RAK算法进行基于标签传播的不相交社区检测。

1 引言

现实世界互连数据 [1] 的规模前所未有地增长，这类数据通常可表示为图。社交与生物网络等现实应用所产生的图规模庞大，因此人们采用并行化来应对规模问题。除了规模庞大之外，大多数现实世界的图在自然属性上也是动态的，边会不断被删除和插入 [2]。因此，近年来，针对动态图更新图分析的并行算法日益受到研究关注。例如：动态计算中心性得分 [3, 4, 5, 6, 7], 、动态维护双连通分量 [8, 9, 10], 、动态计算最短路径 [11, 12, 13]。

社区检测是一类热门的图问题，在电子商务、通信网络和医疗保健等领域具有实际应用。它旨在识别图中的顶点组，即所谓的社区；这些社区对应于系统中的功能单元，内部连接密集，而与网络其余部分的连接稀疏。如果仅根据网络拓扑识别这些结构，则称其为内生社区（本文研究的对象）。另一方面，外生社区则依据节点的外部信息或属性来定义，例如是否属于某个组织或所处的地理位置。图中可以识别出多种类型的社区。不相交社区（本文研究的对象）是指每个顶点恰好属于一个社区的情况。与此不同，重叠社区允许每个顶点属于多个社区 [14], ，而层次化社区则具有多层级成员结构。

社区检测问题是 NP困难。与图划分不同，社区的数量及其规模分布事先并不明确。社区检测有多种方法，包括标签传播 [15, 14, 16], 随机游走 [17], 扩散 [18],自旋动力学 [19], 适应度指标优化 [20, 21], 统计推断 [22, 23], 模拟退火 [24, 19], 团渗流 [25, 26, 27], 等。这些方法可分为两大类：分裂式和凝聚式。分裂方法也称自顶向下方法，从图中所有顶点都属于同一个社区开始，反复识别并移除桥，将其划分为连接更紧密的社区 [28, 29]。凝聚式方法，又称自底向上方法，通过合并两个或多个社区，使某个分数最大化 [30, 2]。在某些情况下，我们有一组感兴趣的相关种子顶点，并将其扩展为围绕这些顶点的社区。这称为种子集扩展 [31]。

不同的社区检测方法可能会得出不同的社区，这些社区可以通过质量函数进行评估。模块度是 Newman 和 Girvan 提出的一种常用适应度函数，它将社区内边的数量与随机零模型中的预期数量进行比较。其取值范围为 -0.5（非模块化聚类）到 1.0（完全模块化聚类）。从理论上讲，优化这一函数可以得到最佳分组。导通率是另一种适应度指标，用于衡量社区割或社区间边。常数波茨模型（CPM）是另一种质量函数，可以克服模块度的某些局限 [32]。

2 文献综述

目前已有多种用于检测网络中社区的技术。其中一些基于模块度优化、层次聚类、标签传播、区域密度、核心聚类 [33], 博弈论、信息论（Infomap） [34, 35, 36, 37, 38],以及生物进化（遗传学） [39, 40, 41]。模块度分数 [20, 42, 40],、归一化互信息指数（NMI） [43, 44], 和杰卡德指数 [43] 等指标常用于比较不同方法所得到的社区质量。

Raghavan Albert Kumara（RAK），也通常称为标签传播算法（LPA），是一种用于识别网络中社区或群组的方法：首先为每个顶点赋予唯一标签，然后让这些标签在图中传播。LPA比Louvain算法更快、可扩展性更强，因为它易于并行化，且不需要反

复执行优化步骤。不过，在某些类型的网络中，它识别高质量社区结构的效果可能不那么理想，而且生成的社区模块度往往低于Louvain算法。此外，LPA得出的解决方案可能并非唯一，也可能包含巨型社区 [45, 15]。

LPA 的改进方案包括：基于节点影响力的标签传播算法（NIBLPA），该算法在出现多个最佳标签时提供稳定（非随机）的标签选择机制 [46]；标签传播算法的标签影响策略（LP-LPA）和扩散激活标签传播算法（SALPA） [47, 48] ，这两种算法旨在解决巨型社区问题；三阶段（TS）算法，该算法首先识别中心节点，再合并社区以提高模块度（尚未提供其执行时间的信息）[49]；以及 Lie 等人的方法，该方法采用前沿和交替推拉策略，以减少访问的边数并提高解的质量 [50]。现有的重叠社区发现方法，例如社区重叠传播算法（COPRA）或说话者-听者标签传播算法（SLPA），也可用于发现互不重叠的社区，方法是为每个顶点选取与之关联的最佳标签 [14, 16]。虽然这些对 LPA 的改进能够提升社区质量，但其运行速度可能较原始 LPA 更慢（在某些情况下会慢很多）。目前已有原始 LPA 的 GPU 加速并行实现，可用于处理无法完全装入 GPU 内存的大规模数据集 [51]。

3 评估

本节首先介绍我们的实验设置，例如所使用的系统和数据集。随后，我们研究几种静态社区检测算法，即 RAK算法、COPRA算法和 SLPA 算法。在实现这些算法的单线程版本和基于 OpenMP 的多线程版本时，我们会关注其中的细节，并探索各种优化方法。

3.1 实验设置

3.1.1 所用系统

我们的实验使用一套系统，其中包括两颗 64 位 Intel Xeon Silver 4116 服务器中央处理器，主频为 2.10 GHz，以及 128GB、频率为 2666 MHz 的 DDR4同步注册式动态随机存取存储器。每颗中央处理器有 12 个 x86核心（每个核心配备 2 个超线程）和 16.5M 的 L3缓存。我们的服务器运行 CentOS 7.9。我们使用 GCC 9.3 和 OpenMP 5.0，并以 3 级优化（-O3）进行编译。所有实验中均启用同步多线程（SMT）。

3.1.2 数据集

我们在实验中使用的图如表1所示。它们均取自 SuiteSparse矩阵集合 [52]。这些图的顶点总数从 7.49 万到 1200 万不等，边总数从 81.1 万到 3.04 亿不等。我们将所有边视为无向边，并默认权重为 1；此外，还为所有图中的每个顶点添加自环。

表1：实验中使用的17个图列表。每个图的所有边都沿反方向重复添加，从而使其成为无向图，并将每条边的权重设为1。此处，$|V|$ 是顶点总数，$|E|$ 是边总数（将图转为无向图后），D_{avg} 是顶点的平均度数。顶点数和边数均按适当精度四舍五入到最接近的千（K）或百万（M）。

图	$\|V\|$	$\|E\|$	D_{avg}
网络图			
web-Stanford	282K	3.99M	14.1
web-BerkStan	685K	13.3M	19.4
web-Google	916K	8.64M	9.43
web-NotreDame	326K	2.21M	6.78
中南半岛-2004	7.41M	304M	41.0
社交网络			
社交网络-Slashdot0811	77.4K	1.02M	13.2
社交网络-Slashdot0902	82.2K	1.09M	13.3
社交网络-Epinions1	75.9K	811K	10.7
soc-LiveJournal1	4.85M	86.2M	17.8
合作网络			
coAuthorsDBLP	299K	1.96M	6.56
coAuthorsCiteseer	227K	1.63M	7.18
coPapersCiteseer	434K	3210万	74.0
coPapersDBLP	540K	3050万	56.5
道路网络			
意大利 OSM	6.69M	14.0M	2.09
英国 OSM	7.73M	16.3M	2.11
德国 OSM	11.5M	24.7M	2.15
亚洲 OSM	12.0M	25.4M	2.12

3.2 RAK算法

Raghavan Albert Kumara（RAK），也称为标签传播算法（LPA），是一种广受欢迎的基于标签传播的社区检测算法。在该算法中，每个顶点最初都被赋予一个唯一标签；在每一步中，每个顶点都会采用其大多数邻居当前持有的标签。在该算法的加权版本（我们使用的版本）中，每个顶点会采用连接权重最大的标签。通过这一迭代过程，连接紧密的顶点组会对一个唯一标签达成共识，从而形成社区。

当有多个与某个顶点相连且权重最大的社区（标签）时，我们会随机选择其中一个（非严格方法），或者只选择其中第一个（严格方法）。当至少有 $100 - n\%$ 的顶点不再改变其社区归属时，算法即收敛（容差参数）。我们通过实验确定一个适用于大多数图的平均容差参数值。此外，我们还希望比较非严格方法和严格方法在模块度（质量）和计算时间方面的表现。

在第一个实验中，我们实现了一个基于中央处理器的单线程 RAK 版本

算法 [1]。我们将非严格方法和严格方法的容差从 0.1 调整到 0.0001。根据结果，我们得出以下观察。非严格方法通常能获得更高的模块度，但也有一些例外（coAuthors*、道路网络）。我们认为这与图的结构有关。在我们看来，0.05 的容差是一个不错的选择。

在第二项实验中，我们实现了基于 OpenMP 的多线程 RAK算法 [2]。每个线程都有一个独立的哈希表，用于根据权重从邻居中选出最受欢迎的标签。最初，我们将各线程的哈希表连续存储在一个向量中。然而，我们发现，分别分配这些哈希表（通过使用指向向量的指针），相比连续分配版本，性能几乎提升了 2 倍。目前，我们采用“auto” OpenMP调度，总共使用 12 个线程。为获得最佳性能，可以通过实验或性能分析选择更合适的调度方式，并最多使用 $N-1$ 个线程（其中 N 是核心/线程数）。与前一项实验类似，我们将容差从 0.1 调整到 0.0001，并比较顺序执行和 OpenMP 实现（非严格方法、严格方法）。

根据第二项实验的结果，我们发现，基于 OpenMP 的严格 RAK 在平均表现上更好，无论是在耗时还是模块度方面。此外，总体而言，它的模块度通常优于顺序执行方法（这可能是由于随机化效果更好）。当容差为 0.05 时，基于 OpenMP 的严格 RAK（使用 12 个线程）比顺序执行的非严格 RAK 快 6.75 倍。不过，在 coAuthor* 图上，OpenMP 实现的模块度并未更高。在社交网络上，基于 OpenMP 的非严格 RAK 模块度优于严格版本。与顺序执行的 RAK 类似，我们认为 0.05 是一个合适的容差值。您可以在我们的代码仓库中找到这些实验的代码和结果。

3.3 COPRA算法

社区重叠传播算法（COPRA）是另一种基于标签传播的社区检测算法。与 RAK 不同，该算法为每个顶点使用多个标签，每个标签都对应一个归属系数（所有归属系数之和为 1）。算法步骤如下。每个顶点初始化为一个独立社区（归属系数为 1）。每次迭代开始时，每个顶点都会从其邻域收集标签。此处，我们排除每个顶点自身的标签，尽管原论文并未明确提及这一点。对于加权图，收集到的标签会根据相连边的权重进行缩放。只有高于某个阈值的标签才会被选中。该阈值与允许的最大标签数（max_labels 参数）成反比。如果所有标签都低于阈值，则随机选取一个最佳标签。如果顶点没有标签，则将其加入自身所在的社区。随后对选中的标签进行归一化，使归属系数之和为 1。重复这一过程，直至收敛。

本文作者建议使用每个社区的最少顶点数来检测收敛。但我们发现这种做法并无帮助。因此，我们采用了与 RAK 类似的收敛条件，即统计改变了最佳标签的顶点数。当这个数量低于某个比例（容差参数）时，我们就认为算法已经收敛。作者还提到使用该算法的同步版本，在这种版本中，

[1]单线程 RAK，https://github.com/ionicf/rak-communities-seq[2]基于 OpenMP 的多线程 RAK，https://github.com/ionicf/rak-communities-openmp

每个顶点的标签仅取决于上一轮迭代中的标签。尽管如此，我们发现异步方法的收敛速度更快（每个顶点的标签可以取决于当前迭代中其邻居的标签）。由于我们的重点是寻找不相交社区，因此将每个顶点的最佳标签作为最终结果。

我们进行实验，以确定一个在模块度和运行时间方面平均而言适用于大多数图的容差参数值。此外，我们调整每个顶点的最大_标签数，而这会控制标签被选取时所需的阈值权重。

在第一个实验中，我们实现了 COPRA 算法 [3] 的单线程 CPU 版本。我们将容差从 0.1 调整到 0.0001，并将最大_标签数从 1 调整到 32。根据结果，我们得出以下观察。平均而言，使用单个标签在运行时间和模块度方面似乎都是最佳选择。这对于道路网络确实如此，但不适用于其他类别的图（例如网络图、社交网络和合作网络）。例如，web-Stanford 和 web-BerkStan 等网络图在最大_标签数为 8 时达到最佳模块度，web-Google 在最大_标签数为 32 时达到最佳模块度，而 web-NotreDame 在最大_标签数为 16 时表现最佳。对于这类图，最大_标签数设为 4–16 会是不错的选择。此外，平均而言，将容差设得比 0.01 更严格，似乎不会对模块度产生任何益处。尽管如此，对于 web-NotreDame、coAuthorsDBLP 和社交网络等图，将容差设得更严格（低于 0.1 的任何值）并无帮助。无论如何，平均而言，0.01 的容差是不错的选择。

在第二个实验中，我们实现了一个基于多线程 OpenMP 的 COPRA算法版本 [4]。与 RAK 类似，我们为每个线程分配一个独立的哈希表，以便根据权重从邻居中选择一组超过特定阈值的标签。如果没有标签超过阈值，我们只选择权重最高的标签；如果仍然没有，则直接加入自己的社区。为获得更好的性能，我们分别分配哈希表，而不是将其连续存储在向量中。同样，我们暂时使用“auto” OpenMP调度和总计 12 个线程（这并非最优选择）。与前一个实验类似，我们将容差从 0.1 调整为 0.0001，并将 max_标签数量从 1 调整为 32。根据结果，我们观察到，基于 OpenMP 的方法比顺序执行更快，而且似乎还能获得更高的模块度。其余观察结果与第一个实验类似。

3.4 SLPA 算法

说话者-听者标签传播算法（SLPA）是另一种基于标签传播的社区检测算法。此算法中，每个顶点都有固定大小的标签记忆，用来记住它听到的所有热门标签。记忆大小固定为待执行的迭代总数加一。算法步骤如下：首先初始化每个顶点，使其记住自身为热门标签。每次迭代开始时，每个邻居（相对于某个顶点）都会从自己的记忆中随机选取一个标签，并将其告知该顶点。该顶点（听者）将最热门的标签添加到自己的记忆中。重复此过程，直到完成固定次数的迭代，即记忆_大小减一。为了加快算法运行速度，我们允许提前收敛

[3] 单线程 COPRA，https://github.com/ionicf/copra-communities-seq 基于 OpenMP 的

[4] 多线程 COPRA，https://github.com/ionicf/copra-communities-open mp

如果至少有 $100 - n\%$ 的顶点记住了先前标签（容差参数）。我们采用 0.05 的容差。由于我们关注的是寻找不相交社区，因此将每个顶点记忆中最热门的标签选作其所属社区。

与 RAK 和 COPRA 不同，这是一种随机化历史标签技术。我们使用随机数生成器（RNG），让给定顶点的每个邻居都从自己的记忆中随机选择一个标签进行传播。最初，我们使用 C++ 的默认随机引擎（可能是梅森旋转算法），但它似乎较慢。后来我们改用 xorshift32 随机引擎。监听顶点会听取所有邻居的意见，并根据边权重选择最流行的标签。在严格模式下，监听顶点会选择第一个最流行的标签记录到自己的记忆中；在非严格模式下，则会随机选择一个最流行的标签进行记录。它会将该标签记录在记忆的下一个索引位置。

我们进行了一项实验，调整每个顶点的记忆_大小，从而控制所需执行的最大迭代次数 [5]。这样做是为了找到一个记忆_大小参数值，使其对大多数图而言都较为合适，兼顾模块度和耗时。

从结果中可以看出，增加每个顶点的记忆_容量会延长完成所需的时间，同时也会提高最终模块度。不过，对于 soc-Slashdot* 图，模块度似乎会随着记忆_容量的增加而降低。我们还观察到，严格变体比非严格变体稍快，但两者的模块度相近。在道路网络上，严格方法似乎得到较低的模块度；而在其他类别的图上，严格方法则能取得较高的模块度。不过，SLPA 算法似乎没有带来良好的投资回报（耗时更长，却未能取得很高的模块度）。需要指出的是，我们没有对社区进行任何后处理（例如拆分不连通的社区）。

4 结论

我们研究了多种基于标签传播的静态社区检测算法，包括 RAK、COPRA 和 SLPA。在实现这些算法的单线程版本和基于 OpenMP 的多线程版本时，我们关注每种算法的细节，进行有益的调整和优化，并确定合适的参数值。容差为 0.05 时，RAK算法表现良好（该容差决定了社区归属发生变化的顶点所占的最大百分比）；在并行实现中，严格 RAK 的表现优于非严格版本（社交网络除外），而使用 12 个线程的基于 OpenMP 的严格 RAK 比顺序非严格 RAK 快 6.75 倍。对于道路网络，COPRA算法使用单个标签时效果良好；对于其他类别的图，max_个标签取 4–16 时效果较好。我们使用该算法的异步版本（其收敛速度更快），并采用与 RAK算法相同的收敛条件。平均而言，0.01 的容差是不错的选择；与顺序执行相比，基于 OpenMP 的实现速度更快，且模块度更高。对于 SLPA 算法，我们使用 xorshift32 随机引擎生成随机数，并再次采用与 RAK算法相同的收敛条件。SLPA 的严格变体

[5]单线程 SLPA，https://github.com/ionicf/slpa-communities-seq

表现优于非严格版本（道路网络除外）；随着记忆_大小增加，其表现良好，但总体而言投资回报并不理想（速度较慢）。因此，我们建议读者在进行基于标签传播的不相交社区检测时采用 RAK算法。

参考文献

[1] S. Sakr、A. Bonifati、H. Voigt、A. Iosup、K. Ammar、R. Angles、W. Aref、M. Arenas、M. Besta、P. A. Boncz 等人，《未来属于大图》，ACM 通讯，第 64 卷，第 9 期，第 62–71 页，2021 年 9 月。[2] N. Zarayeneh 和 A. Kalyanaraman，《Delta-Screening：一种快速高效的动态图社区更新技术》，IEEE 网络科学与工程汇刊，第 8 卷，第 2 期，第 1614–1629 页，2021 年 4 月。[3] K. Lerman、R. Ghosh 和 J. H. Kang，《动态网络的中心性指标》，载于《第八届图挖掘与学习研讨会论文集》，2010 年，第 70–77 页。[4] C. Li、J. Han、G. He、X. Jin、Y. Sun、Y. Yu 和 T. Wu，《静态和动态信息网络中 SimRank 的快速计算》，载于《第 13 届数据库技术拓展国际会议论文集》，2010 年，第 465–476 页。[5] E. Bergamini 和 H. Meyerhenke，《全动态网络中介数中心性的近似计算》，互联网数学，第 12 卷，第 5 期，第 281–314 页，2016 年。[6] E. Nathan 和 D. A. Bader，《一种用于更新图中 Katz 中心性的动态算法》，载于《2017 年 IEEE/ACM 社会网络分析与挖掘进展国际会议论文集》，2017 年，第 149–154 页。[7] S. C. Regunta、S. H. Tondomker、K. Shukla 和 K. Kothapalli，《用于动态接近中心性和介数中心性计算的高效并行算法》，《并发与计算：实践与经验》，第 e6650 页，2021 年。[8] Z. Galil 和 G. F. Italiano，《动态平面图中双连通分量的维护》，载于《自动机、语言与编程国际研讨会论文集》。施普林格，1991 年，第 339–350 页。[9] W. Liang、R. Brent 和 H. Shen，《并行维护 k-连通性的全动态方法》，IEEE 并行与分布式系统汇刊，第 12 卷，第 8 期，第 846–864 页，2001 年。[10] C. A. Haryan、G. Ramakrishna、K. Kothapalli 和 D. S. Banerjee，《用于全动态维护 2-连通分量的共享内存并行算法》，载于《2022 年 IEEE 国际并行与分布式处理研讨会（IPDPS）论文集》。IEEE，2022 年，第 1195–1205 页。[11] P. Narvaez、K.-Y. Siu 和 H.-Y. Tzeng，《用于计算最短路径树的新型动态算法》，IEEE/ACM 网络汇刊，第 8 卷，第 6 期，第 734–746 页，2000 年。[12] E. P. Chan 和 Y. Yang，《动态图中的最短路径树计算》，IEEE 计算机汇刊，第 58 卷，第 4 期，第 541–557 页，2008 年。[13] A. Khanda、S. Srinivasan、S. Bhowmick、B. Norris 和 S. K. Das，《用于在大规模动态网络中更新单源最短路径的并行算法模板》，IEEE 并行与分布式系统汇刊，第 33 卷，第 4 期，第 929–940 页，2021 年。

[14] S. Gregory, “Finding overlapping communities in networks by label propagation,” 《新物理学杂志》，第12卷，第103018页，2010年10月。[在线]。获取地址：https://dx.doi.org/10.1088/1367-2630/12/10/103018[15] U. N. Raghavan、R. Albert 和 S. Kumara，“一种用于检测大规模网络中社区结构的近线性时间算法”，《物理评论E》，第76卷，第3期，2007年9月。[16] J. Xie、B. K. Szymanski 和 X. Liu，“SLPA：通过说话者—听者交互动态过程揭示社交网络中的重叠社区”，载于2011年IEEE第11届数据挖掘研讨会国际会议。IEEE，2011年，第344–349页。[17] M. Rosvall 和 C. T. Bergstrom，“复杂网络中的随机游走地图揭示社区结构”，《美国国家科学院院刊》，第105卷，第4期，第1118–1123页，2008年。[18] K. Kloster 和 D. F. Gleich，“基于热核的社区检测”，载于第20届ACM SIGKDD知识发现与数据挖掘国际会议论文集，2014年，第1386–1395页。[19] J. Reichardt 和 S. Bornholdt，“社区检测的统计力学”，《物理评论E》，第74卷，第1期，第016110页，2006年。[20] M. E. Newman，“利用矩阵的特征向量发现网络中的社区结构”，《物理评论E》，第74卷，第3期，第036104页，2006年。[21] S. Fortunato，“图中的社区检测”，《物理学报告》，第486卷，第3–5期，第75–174页，2010年。[22] E. Cˆome 和 P. Latouche，“基于精确积分完全数据似然的随机块模型选择与聚类”，《统计建模》，第15卷，第564–589页，2015年3月，doi：10.1177/1471082X15577017。 [在线]。获取地址：https://doi.org/10.1177/1471082X15577017[23] M. E. Newman 和 G. Reinert，“估计网络中的社群数量”，《物理评论快报》，第117卷，第7期，第078301页，2016年。[24] R. Guimera 和 L. A. Nunes Amaral，“复杂代谢网络的功能制图”，《自然》，第433卷，第7028期，第895–900页，2005年。[25] I. Der´enyi、G. Palla 和 T. Vicsek，“随机网络中的团渗流”，《物理评论快报》，第94卷，第16期，第160202页，2005年。[26] S. Maity 和 S. K. Rath，“用于检测重叠社区结构的扩展团渗流方法”，载于2014年计算、通信与信息学进展国际会议（ICACCI）。IEEE，2014年，第31–37页。[27] S. K. Gupta、D. P. Singh 和 J. Choudhary，“基于团的重叠社区检测算法综述”，《知识与信息系统》，第1–36页，2022年。[28] M. Girvan 和 M. E. Newman，“社交与生物网络中的社区结构”，《美国国家科学院院刊》，第99卷，第12期，第7821–7826页，2002年。[29] S. Souravlas、A. Sifaleras、M. Tsintogianni 和 S. Katsavounis，“社交网络社区检测方法分类：综述”，《国际一般系统杂志》，第50卷，第1期，第63–91页，2021年1月。[30] A. Zakrzewska 和 D. A. Bader，“一种用于图中局部社群检测的动态算法”，载于2015年IEEE/ACM社交网络分析与挖掘进展国际会议（ASONAM）。IEEE，2015年，第559–564页。

[31] J. J. Whang、D. F. Gleich 和 I. S. Dhillon，“使用种子集扩展进行重叠社区检测”，载于第22届ACM信息与知识管理国际会议论文集，2013年，第2099–2108页。[32] V. A. Traag、P. Van Dooren 和 Y. Nesterov，“无分辨率极限社区检测的适用范围有限”，《物理评论E》，第84卷，第1期，第016114页，2011年。[33] Y. Ruan、D. Fuhry、J. Liang、Y. Wang 和 S. Parthasarathy，《社区发现：简单且可扩展的方法》，丛书：人机交互丛书。施普林格国际出版，2015年，第23–54页。[34] J. Zeng 和 H. Yu，“通过遗传算法有效统一优化大规模图社区检测”，载于2019年IEEE国际大数据会议（Big Data）。IEEE，2019年12月，第475–482页。[35] ——，“一种可扩展且高质量的分布式Infomap算法”，载于第47届国际并行处理会议论文集——ICPP 2018。美国纽约：ACM出版社，2018年8月，第1–11页。[36] M. A. M. Faysal 和 S. Arifuzzaman，“利用信息论方法在大型网络中进行分布式社区检测”，载于2019年IEEE国际大数据会议（Big Data）。IEEE，2019年12月，第4773–4782页。[37] M. Rosvall、D. Axelsson 和 C. T. Bergstrom，“地图方程”，《欧洲物理学杂志特刊》，第178卷，第1期，第13–23页，2009年11月。[38] L. Rita，《Infomap算法：一种社区发现算法》，2020年4月。 [在线]。获取地址：https://towardsdatascience.com/infomap-algorithm-9b68b7e8b86[39] T. B. D. P. Aditama 和 A. Sn，“使用遗传算法确定社交网络中的社区结构和模块度”，《印度尼西亚计算与控制论系统杂志》，第14卷，第219–230页，2020年。[40] A. K. Ghoshal、N. Das、S. Bhattacharjee 和 G. Chakraborty，“一种用于大型网络社区检测的快速并行遗传算法方法”，载于2019年第11届通信系统与网络国际会议（COMSNETS）。IEEE，2019年1月，第95–101页。[41] Y.-C. Lu 和 G. Chakraborty，“利用多目标优化遗传算法提高图聚类效率”，《国际应用科学与工程杂志》，第17卷，第2期，第157–173页，2020年6月。[42] V. D. Blondel、J.-L. Guillaume、R. Lambiotte 和 E. Lefebvre，“大型网络中社区的快速展开”，《统计力学杂志：理论与实验》，第2008卷，第10期，第P10008页，2008年10月。[43] A. Jain、R. Nasre 和 B. Ravindran，“DCEIL：基于CEIL分数的分布式社区检测”，载于2017年IEEE第19届高性能计算与通信国际会议；IEEE第15届国际智慧城市会议；IEEE第3届国际数据科学与系统会议（HPCC/SmartCity/DSS）。IEEE，2017年12月，第146–153页。[44] P. Chopade 和 J. Zhan，“一种利用博弈论建模进行大型网络社区检测的框架”，《IEEE大数据汇刊》，第3卷，第3期，第276–288页，2017年9月。[45] M. E. J. Newman，“检测网络中的社区结构”，《欧洲物理学杂志B辑：凝聚态》，第38卷，第2期，第321–330页，2004年3月。

[46] Y. Xing、F. Meng、Y. Zhou、M. Zhu、M. Shi 和 G. Sun，《一种基于节点影响力的标签传播算法，用于网络社区检测》，The Scientific World Journal，第2014卷，2014年。[47]K. Berahmand 和 A. Bouyer，《LP-LPA：一种基于链接影响力的标签传播算法，用于发现网络中的社区结构》，International Journal of Modern Physics B，第32卷，第06期，第1850062页，2018年3月。[在线]。网址：http://www.worldscientific.com/doi/abs/10.1142/S0217979218500625[48]M. Sattari 和 K. Zamanifar，《一种基于扩散激活的标签传播算法，用于动态社交网络中的重叠社区检测》，Data & knowledge engineering，第113卷，第155–170页，2018年1月。[在线]。网址：https://linkinghub.elsevier.com/retrieve/pii/{S0169023X16303500}[49] X. You、Y. Ma 和 Z. Liu，《一种用于社交网络社区检测的三阶段算法》，Knowledge-Based Systems，第187卷，第104822页，2020年。[50] X. T. Liu、M. Halappanavar、K. J. Barker、A. Lumsdaine 和 A. H. Gebremedhin，《方向优化的标签传播及其在社区检测中的应用》，载于《第17届ACM计算前沿国际会议论文集》。美国纽约：ACM，2020年5月，第192–201页。[51] Y. Kozawa、T. Amagasa 和 H. Kitagawa，《通过并行标签传播实现GPU加速图聚类》，载于《2017年ACM信息与知识管理会议论文集——CIKM '17》。美国纽约：ACM出版社，2017年11月，第567–576页。[52] S. Kolodziej、M. Aznaveh、M. Bullock、J. David、T. Davis、M. Henderson、Y. Hu 和 R. Sandstrom，《SuiteSparse矩阵集合网站界面》，The Journal of Open Source Software，第4卷，第35期，第1244页，2019年3月。