分　数：	
评卷人：	

華中科技大學

研究生（数据中心技术）课程论文（报告）

题 目：图划分技术研究综述

学　　号 M202173707

姓　　名 王源博

专　　业 计算机计术

课程指导教师 施展 童薇

院（系、所） 计算机科学与技术学院

2021 年 12 月 25 日

图划分技术研究综述

王源博[1)]

[1)](华中科技大学 计算机科学与技术学院，武汉 430074)

摘 要 图划分是大规模分布式图处理中的一个重要步骤。在划分现实世界幂律图时，边缘划分算法优于传统的顶点划分算法，因为它可以将单个顶点分割成多个副本来分摊计算量。许多高级的边划分方法都是为静态图从头开始划分而设计的。然而，现实中的图结构不断变化，导致分区质量下降，影响了图应用程序的性能。一些研究专注于离线重分区或批处理增量分区，但如何实时处理动态分区仍值得深入研究。讨论了分区的动态变化对分区的影响，发现插入和删除都会导致局部的次优分区，这是分区质量下降的原因。为了解决这一问题，提出了一种动态边缘划分算法。通过分布式流处理动态问题，并通过重新分配一些紧密连接的边来提高分区质量。实验表明，该方法对初始划分质量、动态尺度和类型以及分布尺度都具有鲁棒性。平衡图划分是许多具有关系数据的大规模分布式计算的关键步骤。随着图数据集的规模和密度的增长，出现了一系列高度可伸缩的平衡划分算法，以满足不同领域的不同需求。在 3 篇研究图划分问题的 CCF A 类期刊、会议的基础上概括出图划分技术的问题背景，总结了当前的研究现状，以及并指明了未来的研究方向。

关键词 图计算；图划分；动态图；分布式系统；边划分；

A Survey on Graph Partition

Wang Yuanbo[1)]

[1)](School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, Hubei 430074)

Abstract Graph partitioning is an important step in large-scale distributed graph processing. When dividing real-world power-law graphs, the edge division algorithm is better than the traditional vertex division algorithm because it can divide a single vertex into multiple copies to share the amount of calculation. Many advanced edge division methods are designed for dividing static graphs from scratch. However, in reality, the structure of the graph is constantly changing, resulting in a decrease in the quality of the partition, which affects the performance of the graph application. Some research focuses on offline repartitioning or batch incremental partitioning, but how to deal with dynamic partitioning in real time is still worthy of in-depth study. Discussed the impact of the dynamic changes of the partition on the partition, and found that both insertion and deletion will lead to the partial sub-optimal partition, which is the reason for the deterioration of the partition quality. In order to solve this problem, a dynamic edge partitioning algorithm is proposed. Deal with dynamic problems through distributed streams, and improve the quality of partitions by redistributing some tightly connected edges. Experiments show that this method is robust to the quality of initial division, dynamic scale and type, and distribution scale. Balanced graph partitioning is a key step for many large-scale distributed computing with relational data. With the growth of the scale and density of graph data sets, a series of highly scalable and balanced partitioning algorithms have emerged to meet the different needs of different fields. On the basis of 3 CCF Class A journals and conferences that study the problem of graph division, summarize the problem background of graph division technology, summarize the current research status, and point out the future research direction.

Key words Graph Computing; Graph Partition; Dynamic Graph; Distributed System; Edge Division;

1、引言

图是计算机科学中最常用的一类抽象数据结构，在结构和语义方面比线性表和树更为复杂，更具有一般表示能力。许多大数据应用都作用于图上，不仅有最短路径，网页排序等传统应用，还包括社交网络分析、生物信息网络分析、道路交通管理等新兴应用。随着图数据量的增长，图的存储和处理要在分布式环境中执行。为了满足这一需求人们提出了许多分布式图处理系统。图划分是分布式图处理的基础，它将一个图划分成多个部分。划分的好坏对分布式图处理的性能有很大的影响。首先处理速度取决于最慢的部分，因此要划分平衡。其次在分布式图应用中，一些顶点需要与远端的邻居通信来传递消息，这导致了部件间(inter-part)通信。由于网络延迟，部件间通信比部件内通信成本高。因此，基于分区负载均衡的图划分的目标是最小化子图间通信开销。然而，平衡图划分是一个 NP 难题

图划分是 Pregel 等系统进行分布式计算分前提。由于图计算通常按照拓扑结构访问数据，所以每次迭代处理都会引入巨大的通信开销，成为制约分布式处理性能的关键因素。因此一个良好的划分算法应保证划分后的子图在负载均衡的前提下，减少子图之间交互边（切分边）的规模，从而减少网络通信。另一方面云计算资源会随着并发处理作业数目的变化和集群中节点的增删而动态变化。Pregel 等系统使用集群总任务数目来定量描述计算资源，使用空闲任务数目表征当前可用计算资源。，并提供 setTaskNum 接口，允许用户在提交图处理作业时，根据处理需求和计算资源灵活定义本作业的任务数。因此同一个作业在在不同时刻被提交时，其分布式任务数目不尽相同，这称之为分布式处理力粒度的弹性变化。这导致图数据需要按照当前的任务数目重新划分，划分结果的不可重用性使图划分的执行效率也成为影响总计算代价的重要因素。

然而均衡图划分本身是一个经典的 NP—complete 问题。早期的图划分算法以 Kernighan-Lin 算法为代表，主要用于超大规模集成电路设计中的电子元器件布局。其主要思想是首先将一个网络图分割成两个大小相等的顶点集合 A 和 B，在集合 A 和集合 B 中的顶点，除了和本集合内顶点的有边连接外，还可能和另外一个集合内的顶点有边连接，对于后者，用 T 表示所有这种连接边的权重之和，作为衡量集合 A 和集合 B 之间连通性的指标。K-L 算法在执行过程中，不断调整集合 A 和集合 B 内的顶点，直到 T 值最小。但是 K-L 算法的时间复杂度过高，随着图顶点数据的增大，将超出目前的计算处理能力。为降低时间复杂度，有文献提出了基于 Quick_Cut 计算的 K-L 算法，利用邻居搜索的特点，避免了不必要图顶点的遍历。还有文献提出了针对分布式 P2P 网络，提出了一种基于图顶点的度序列和广度优先搜索 k-图分割技术，能够将一个大型的 P2P 网络分割成 k 个自网络并能做到各个子网络的任务负载均衡。文献提出了通过 3 步处理来实现大规模图的分割：1.建立带权重的深度优先搜索树 2.将大图分割成若干均衡的子图 3.迭代处理尽量减少子图之间的关联。

近年来，相关的研究工作可以分为两类，一类以 METIS 算法为代表的离线划分算法和以 LDG 为代表的在线划分算法。前者可以显著优化切分边规模，降低迭代计算过程中的通信开销，因此受到学术界和工业界的广泛关注。然而离线划分过程需要频繁访问图顶点，引入了昂贵时间开销。面对海量图数据，例如 Twitter 图（4700 万顶点，15 亿出边），其执行时间超过 8 小时，效率低下。另一方面，在线划分算法可以在图处理系统的数据加载阶段完成图划分，仅扫描一次图数据。与离线划分算法相比，在线划分算法通过一定程度上牺牲划分效果，来获得较高的执行效率。但是此类算法通常是集中式算法，以便于维护复杂的启发式规则，保证相对较好的划分效果，其拓展性显然受到了单机处理能力的限制。虽然已经存在分布式在线划分算法的相关研究工作，但是启发式规则的维护开销仍然限制影响了算法的运行效率。

本文总结了当前的图划分问题的研究现状和进展，详细分析了存在的挑战性问题，并探讨了未来的研究方向。

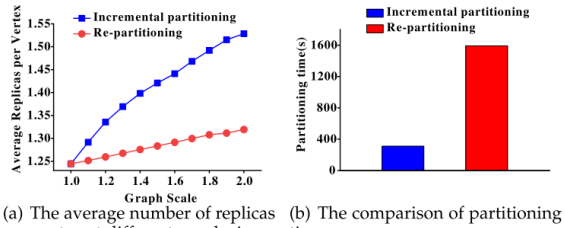
2. 原理和优势

许多大数据应用都发挥着图的作用，不仅是传统的最短路径确定、论文引文关系挖掘、网页排名等应用，还有一些新兴的应用，如社会网络分析、生物信息网络分析、道路交通管理等。随着图形数据量的增长，需要在分布式环境中进行图形存储和图形处理任务。为了满足这一需求，提出了许多分布式图处理系统。图划分是分布式图处理的基础，它将一个图划分为多个部分。分区的质量对分布式图

处理的性能有很大的影响。首先，处理速度取决于最慢的部分，因此分区应该是负载均衡的。其次，在分布式图应用中，一些顶点需要与远端邻居通信来传递消息，从而导致部件间通信。由于网络延迟，部件间通信的成本高于部件间通信。因此，图分区的目标是在分区负载均衡的基础上使各部分之间的通信开销最小。然而，平衡图划分是一个 NP-hard 问题。

近年来，图的划分经历了一个从顶点划分到边划分的转变。之前的研究已从理论上证明，对于自然幂律图，边缘划分在分布式图计算中具有较好的性能。在边划分中，每条边被分配到一个部分中，一个顶点的相邻边可能不在同一个部分中，因此需要在对应的部分中创建顶点副本，避免了超顶点造成的负载不平衡。由于计算是沿着边进行的，所以零件的负载由边的数量决定。为了降低副本间同步的部件间通信开销，边划分需要基于分区负载均衡最小化顶点副本。边划分问题的解决已经有了很多先进的工作，但大多数算法都是从头开始对图进行划分，如全局边划分算法和流边划分算法。

现实世界的图总是动态的，图的结构不断变化。Twitter 的月活跃用户在 10 年内从 4.82 亿增长到 25 亿研究在动态图上表明，真实世界图的密度会逐渐增大，直径会逐渐减小。在初始划分的基础上，动态划分会导致部件之间的连接更加紧密，从而增加部件之间的通信成本。最近，提出了一些先进的全局分区算法的增量版本，如 DNE 的 IncDNE 算法，将一批动态更改累加在一起进行分区。然而，它不仅需要额外的空间(可能与初始图形一样大)，而且不支持某些实时图形应用程序。流和基于散列的分区算法支持实时分配新边，但它们的质量很差。随着图的变化，需要进一步提高分区质量。一种可以想象的提高划分质量的方法是通过一个高质量的全局边缘划分算法对整个图进行重新划分。但由于其成本高、数据迁移量大，在动态环境下是不可行的。在图 1 中，比较了处理十亿规模动态图的两种不同方式:1)利用 HDRF 实时增量分割图的动态;2)利用全局算法 NE 在每个时间点对整个图进行重新分割。在图 1a 中，随着图的规模逐渐增大到 2 倍，重新划分算法只增加了约 0.07 的平均顶点复制，而增量算法增加了约 0.3 的平均顶点复制。这个 0.23 的差距表示在十亿规模的图中有大量的冗余顶点副本，这将导致图任务中有大量额外的同步开销。当图的规模达到 2 倍时，增量算法只需要 300 秒，而重分区算法需要近 1600 秒，这是无法接受的。它们都不能很好地解决动态边界划分问题。此外，一些轻量级的重新划分算法试图通过在部分之间迁移边或顶点来降低时间成本，而不是对整个图执行全局划分算法。然而，其中一些算法是为顶点划分而设计的，而另一些算法则没有考虑图结构的动态特性。文章介绍了一种动态边缘划分算法 GR-DEP，该算法在分区动态变化的情况下，能够实时提高分区质量。作者发现插入和删除都会导致局部次优分区，从而产生冗余副本。



(a) The average number of replicas per vertex at different graph sizes. (b) The comparison of partitioning time.

图 1.增量划分算法与重划分算法比较。

文章介绍了一种动态边缘划分算法 GR-DEP，该算法在分区动态变化的情况下，能够实时提高分区质量。我们发现插入和删除都会导致局部次优分区，从而产生冗余副本。为了消除局部次优划分，作者将一些紧密连接的边作为一组来处理，通过重新分配与其他部分连接较紧的部分的组来减少局部顶点切割。在 GR-DEP 中，作者通过一个分布式流对每个传入的动态更改进行分区，并通过一种局部重分配方法实时提高分区质量。作者的主要贡献如下:提出了一种动态边缘划分算法来处理在线图结构的变化。它还可以通过重新分配少量的边来提高分区的实时性。讨论了不同的动态变化对分区质量的影响，发现局部次优分区是不可避免的，这是动态图中存在大量冗余顶点副本的主要原因。证明了一些紧密连接的边作为一组迁移可以改善局部次优划分。设计了一种结构感知的方法和一种自我中心的方法来搜索和迁移组。将它们应用于不同的顶点可以获得更好的效果。在动态分区之前和之后运行图计算任务。当图的大小增加 40%时，与最先进的动态分区算法相比，可以减少增加的通信成本和任务总时间，分别减少 41.5%和 71.4%。

3 研究进展

3.1 边组重分配的动态划分技术

在边划分中，所有的动态变化都被视为边。一个

顶点可以看作是它所有相邻边的集合。对于每条新边，我们必须确定它将被分配到哪个部分。新边具有以下四种类型:

(1)两个端点都是新顶点;

(2)一个端点是一个新顶点，另一个在分区中有副本;

(3)两个端点在分区中都有副本，它们被相同的部分覆盖;

(4)两个端点在分区中都有副本，但它们不被相同的部分覆盖。

如何实时分配每个传入的新边可以参考流边界划分。插入一条新边可以看作是流边界划分的中间状态。每个决策都需要在分区平衡的基础上最小化创建的副本。对于给定的一个边(v_i, v_j),每个划分$P_s \in P$，之前的文献提出了一个得分函数$C(v_i, v_j, P_s)$来定义变得分配。

$$C(v_i, v_j, P_s) = C_{REP}(v_i, v_j, P_s) + C_{BAL}(P_s),$$

作者将边组（EG）定义为迁移之后可以使图的局部次优划分变为局部最优划分的一组边。然后定义了迁移边组可以获得的值$gain$

$$gain_{i\to j} = |evc| - |ivc|,$$

作者将每次动态变化视为减少顶点复制的机会，并在分配新边或删除旧边后，在局部结构中实时搜索 EG。GR-DEP 的概况如图 2 所示。动态更改从分布式流进入分区。每个部分都有一个动态处理器和一个 EG 处理器。前者处理每个到达的动态更改，后者搜索动态更改附近的 EG。每个顶点拥有一个邻接表和一组复制位置，每条边都连接到两个端点顶点。

Dynamic 处理器。Dynamic 处理器用于接受和处理每个传入的动态更改(算法 1)。对于删除，动态处理器将查找当前部分的边缘。如果边不在当前部分中，它将被发送到端点所在的部分中。被删除的边的两个端点将从彼此的邻接表中删除，没有邻接边的顶点将从该部分中删除。对于插入，动态处理器将根据当前部件中的信息确定其赋值，并将其发送给目标部件。如果端点没有被目标部分覆盖，则应该创建一个副本，并通知其他副本更新信息。

Algorithm 1. Dynamic Processor

Input: Graph G, dynamic edge e
1: $P_t \leftarrow$ the assignment of e;
2: send e to P_t;

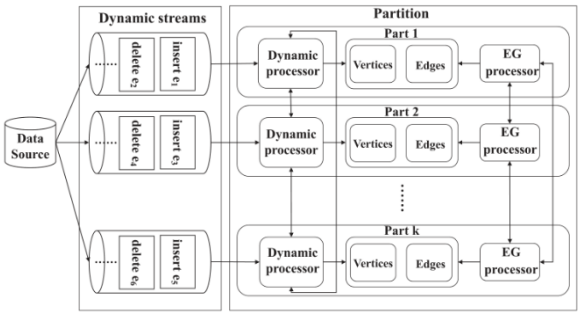


图 2.GR-DEP 模型

EG 处理器。每个部分都有一个 EG 处理器，在每个接收到的动态变化附近寻找 EG(算法 2)。EG 将迁移到其目标部分。具体的搜索方法将在下面提到。但是，对每个动态更改执行搜索的开销是不可忽略的。实际上，在重新分配 EG 后，所涉及的局部结构可以认为是最优的。换句话说，在这种地方结构中不太可能找到新的 EG。因此，在这样一个最近的重新分配结构中，我们跳过了动态变化。我们纪念的端点附近发现,例如,使用参数 T 控制每个标记顶点搜索的次数可以跳过。事实表明,当 T=5,与不设置 T 相比,可以提高搜索成功率的 2 - 5 倍,而 EG 只有 15%– 25%减少。如果分区不平衡，过载部分的 EG 处理器会随机选择边界点，搜索其附近的 EG。

Algorithm 2. *EG* Processor

Input: Edge e, parameter T
1: **if** the endpoint of e appears in the dynamic change for the first time or the Tth time after reassignment **then**
2: search *EG* near e;
3: **if** an *EG* is found **then**
4: migrate *EG* to its destination part;
5: **end if**
6: **end if**

首先边组的结构是不固定的，所以一个固定的搜索方法深度优先搜索或者广度优先搜索都无法找出边组。然后作者提出了一个基于结构感知的优先级搜索算法。提出了一个基于结构感知的优先级搜索算法。首先低于最低负载的子图是不能迁移组出去的，因为这会导致负载均衡。然后假设动态变化是边(v_i, v_j)插入或者删除到P_s。

然后从端点处开始搜索，初始化一个优先队列 Q，集合 VG 表示已经选择的点，集合 EG 表示已经选择的边。在整个搜索过程中，保持记录两个值 evc 和 ivc。只有队列 Q 为非空搜索就不停止，每次取出队头的点 P，然后把 P 和 VG 中相连的边都加入 EG，再把 P 加入集合 VG。然后更新 evc 和 ivc。如

果得到的最大 gain 值是正值就停止搜索，现在得到的集合 VG 就是要迁移的边组。

Algorithm 3. Structure-Aware Search Method

Input: Vertex v, part P_s, parameter M_1;
Output: EG;

if $|P_s| < (2-\varepsilon)\frac{|E|}{k}$ **then**
  return null;
end if
priority queue $Q = \{v\}$, $EG = \emptyset$, $VG = \emptyset$;
$evc\_num\_arr = \{0, \ldots\}$, $ivc\_num = 0$;
while $Q \neq \emptyset$ & $|EG| < M_1$ **do**
  $v_q \leftarrow Q.pull()$;
  push all edges between v_q and VG into EG;
  $VG \leftarrow VG \cup \{v_q\}$;
  update evc_num_arr and ivc_num;
  for all other parts $P_d \in P$, calculate $gain_{s\to d} = evc\_num\_arr[d] - ivc\_num$;
  $MaxGain \leftarrow \max_{|P_d| < \varepsilon\frac{|E|}{k}} gain_{s\to d}$;
  if $MaxGain > 0$ **then**
    return EG;
  end if
  if v_q is not a high-degree vertex **then**
    calculate priority for all neighbor vertices of v_q and push them into Q;
  end if
end while
return null;

结构感知的搜索方法可以根据局部结构的特点使得搜搜出 EG 的可能性最大化，然后还是存在一些问题。一是对于子图内部的改变搜索的成功率较低，二是对于度数较高的边界点，搜索的开销很大。为了改善这两种情况下的搜索性能，提出了一个中心搜索算法。它指定一个边的每个端点都只能在以自我为中心的网络中寻找边组。自我中心搜索算法只在一个固定小区域里面搜索边组，虽然搜索的结果可能不全面，但是开销较小并且稳定。所以适合中心变化的情况，以及度数较高的节点。

实验证明作者提出的一种动态边缘划分算法 GR-DEP，可以提高图结构变化时的划分质量。作者发现，局部次优分区会导致冗余的副本。为了减少冗余顶点，提出了两种搜索边缘组的方法。实验表明，GR-DEP 方法在质量和性能上都优于现有的实时动态分区方法。应用于图形应用程序，GR-DEP 可以将总时间从 6%减少到 21%。GR-DEP 适用于实时动态环境，其效果可与批量划分算法相媲美。

3.2 平衡图划分的优先重流算法

作者首先观察到最近引入的两组迭代划分算法——基于重流和基于平衡标签传播(包括 Facebook 的 Social Hash 分区)可以通过设计决策的公共模块化框架来看待。在这种模块化视角的帮助下，作者发现设计决策的关键组合导致了一个新的算法家族，其经验性能明显优于任何现有的高可伸缩性算

Algorithm 4. Egocentric Search Method

Input: Vertex v, part P_s, parameter M_2;
Output: EG;

if $|P_s| < (2-\varepsilon)\frac{|E|}{k}$ **then**
  return null;
end if
$evc\_num\_arr = \{0, \ldots\}$, $ivc\_num = 0$;
$EG = \emptyset$, $VG = \{v\}$;
if $|N_s(v)| > M_2$ **then**
  \\cr $N_s(v)$ is the set of neighbor vertices of v in P_s;
  Add the M_2 vertices with the most replicas in $N_s(v)$ into VG;
else
  $VG \leftarrow VG \cup N_s(v)$;
end if
for each v in VG, update evc_num_arr by $R(v)$;
$P_d \leftarrow \arg\min_{|P_d| < \varepsilon\frac{|E|}{k}} evc\_num\_arr[d]$;
remove the vertex v that $P_d \notin R(v)$ from VG;
check if the remaining vertices in VG are ivc, and update ivc_num;
if $evc\_num\_arr[d] - ivc\_num > 0$ **then**
  add all edges between vertices in VG into EG;
  return EG;
end if
return null;

法在广泛的现实世界的图上。由此产生的优先级流媒体算法借鉴了流媒体文献，采用了基于乘法权的约束管理策略，同时采用了平衡标签传播中的优先级概念来优化流媒体流程的顺序。我们的实验结果考虑了一系列的流顺序，其中基于称之为矛盾的动态排序在最终平衡分区的切割质量方面大体上是最具表现力的，而基于程度的静态排序也几乎一样好。

一种常见的节点集划分方法是对节点集进行简单的哈希处理，有效地在集群之间均匀随机地分布节点。但更智能的分区方法可以大大提高这些分布式算法的运行时间。与通用图集群任务相比，这个分区任务的一个重要需求是，作者寻求平衡与分区的每个集群相关的计算负载。在本工作中，作者将重点关注每个节点的计算负载是恒定的环境，但我们考虑和介绍的算法都可以很容易地修改，以考虑非均匀/加权负载

进入我们的问题，平衡图划分:给定一个输入图，我们如何将节点集划分为(1)在 k 个集群 s 之间保持负载均衡，同时(2)最小化某个目标函数。我们关注边切目标最小化跨越多个分片的边的数量——因为它与文献密切相关。需要认识到，在实际的问题中，最小化边可能不能完全描述工作负载性能，但我们使用这个目标作为代理，并对各种设计决策对结果的影响进行分类。目标函数的其他例子包括超图的扇出最小化和图群随机化中的方差最小化。

不幸的是，对于中等的图来说，找到边切问题的精确解是不可行的:当碎片的数量是 2 时，这个问题等同于最小平分问题，它是典型的 NP-hard，并且没有已知的有效算法具有良好的逼近保证。也就是

说，有大量的工作是关于实际的，尽管是启发式的算法，这些算法在一系列相关的大规模图数据集上表现得很好。

最近关于图划分的可扩展实用算法的研究主要是由管理一些世界上最大的关系数据集的公司所推动的。在这项工作中，围绕三种最近的算法构建了一个共同的框架，这三种算法通常被认为是或接近于针对不同目标的最先进的算法:平衡标签传播(BLP)、重新流线性确定性贪婪(reLDG)和社会哈希分配器(SHP)。在作者的实验评估中，我们还以最近的一种基于线性嵌入的高性能算法为基准，这种方法与其他方法没有明显的关联。BLP 和 SHP 属于一类基于标签传播的算法。从初始分配开始，他们迭代地进行节点重定位，以实现更高质量的分区。ReLDG 是所谓的重流算法[24]的一个例子，该算法在重复传递中串行地处理节点集，每个节点的放置都按照旨在实现平衡的分配规则进行。流算法通常是由高度受限的计算框架驱动的，在此框架中，当图在传输、移动和/或加载(在 ETL 期间，在数据仓库语言中)时，试图进行节点分配。因此，在此工作之前测试的节点集的流排序只有随机广度优先搜索(BFS)和深度优先搜索(DFS)，以模仿网络爬虫或等效进程获得的顺序。因此，我们的工作是第一个:(1)将后者与可伸缩的非流算法进行基准测试，(2)探索战略流顺序，即算法考虑的节点集的顺序。我们称这些算法为平衡图划分的优先重流算法。

作者的主要贡献有 3 点：

1) 提供了文献中没有的基准测试，表明现有的 restreaming 算法 reLDG 在现实世界的一系列图上优于 BLP 和 SHP1。
2) 在讨论中对这三种算法进行了模块化，并注意到它们实际上是在一个公共框架中设计决策的三种不同组合，涉及到它们如何管理约束、节点优先级和一个我们称为 incumbency 的概念。
3) 同时引入了静态和动态流排序，后者在流迭代之间可能会有所不同，作为一种将优先级注入流算法以实现平衡图划分的方法。特别地，一个这样的动态排序，矛盾排序，在所有的测试用例中产生最好或接近最好的结果，紧跟着是一个静态程度排序。

同步分配与流分配。BLP 和 SHP 利用前面分区的静态快照中的信息，同时进行所有节点的重定位。在 reLDG 中，节点从节点列表的连续传递中一次分配一个，在流中稍后更改节点的分配情况。在文中，将这种区别表示为同步赋值与流赋值

基于流的和成对的约束处理。在两个同步算法之间，BLP 使用一个线性程序来保持平衡，在每个分片的节点净流入等于净流出的约束下，最大化重定位增益，直到一个期望的不平衡参数。另一方面，KL-SHP 只是确保相同数量的节点在分片对之间移动。由于前者有流体动力学的解释，我们称之为基于流的约束处理策略。后者称之为成对约束处理。

优先级排序。在这项工作中，将优先级定义为如何考虑非在职节点的重新定位的顺序。BLP 和 KL-SHP 在进行节点迁移时都优先考虑增益。他们利用重定位队列中的排序，首先移动收益最高的节点，从而直接优化边割。另一方面，重新 reLDG 在节点迁移时不优先考虑任何度量;节点按照流顺序随机排列优先级。这一事实突出了这一算法家族的改进机会。

最终作者通过实验发现，提出的优先重流划分算法具有最先进的划分性能。

	Synchronous				Streaming (reLDG)						
Graph	SHP-I	SHP-II	KL-SHP	BLP	Random	CC	BFS	Degree	Ambivalence	Gain	METIS
pokec	0.578	0.595	0.585	0.532	0.675	0.681	0.698	**0.716**	0.712	0.618	0.827
livejournal	0.626	0.648	0.625	0.617	0.674	0.666	0.731	0.745	**0.749**	0.671	0.899
orkut	0.535	0.555	0.534	0.531	0.650	0.628	0.665	**0.689**	0.679	0.626	0.711
notredame	0.783	0.635	0.652	0.612	0.882	0.864	**0.929**	0.902	0.924	0.878	0.982
stanford	0.737	0.711	0.697	0.629	0.856	0.844	0.891	0.900	**0.916**	0.793	0.973
google	0.670	0.603	0.616	0.606	0.848	0.814	0.868	0.959	**0.964**	0.799	0.989
berkstan	0.701	0.652	0.658	0.585	0.858	0.805	0.895	0.913	**0.918**	0.766	0.988

图 3.reLDG 算法性能

3.3 可伸缩，多约束，复杂目标图划分

作者提出了 XTRAPULP，一种用于处理不规则万亿边图的分布式内存图划分程序。XTRAPULP 基于可扩展标签传播社区检测技术，该技术在之前的各种作品中已经被证明是一种可行的方法，可以以最小的计算时间生成高质量的倾斜和小世界图的分区。作者的 XTRAPULP 实现还可以一般化，以计算具有任意数量约束的分区，并且可以计算跨所有部分的通信负载均衡的分区。在大型稀疏图的集合上，我们证明了 XTRAPULP 分区比最先进的分区方法快得多，同时也证明了 XTRAPULP 可以在几分钟内生成具有超过 10 亿个顶点和超过 1000 亿个边的真实图的分区。

在并行应用程序中，图分区是确保负载均衡计算和减少节点间通信的必要预处理步骤。随着在线社交网络、网络图表和其他非传统图表数据(如脑图)的规模以指数级的速度增长，可扩展和高效的算法对它们进行分割和分析是必要的。这些网络的典型特征是高度倾斜的顶点度分布和低平均路径长度。其中一些图可以使用“小世界”图模型、建模，其

他的则称为“幂律”图。

对于高度并行的分布式内存图分析，在十亿个顶点或十亿个边缘的小世界图上，任何计算和通信的不平衡都可能导致显著的性能损失。因此，可以使用图分区来提高平衡性。虽然有些人认为，共享内存图处理可以比分布式处理在图的简单分析上更好，这些图适合一台机器的 RAM 或磁盘，超大规模网络，内存密集型动态规划算法，而其他计算复杂的分析算法在许多情况下仍然需要分布式处理模型。

传统的图划分工具通常是为规则网格上的科学计算应用程序设计的，但它们在能够划分的图的大小或支持的划分目标度量方面都受到了限制。此外，与科学计算应用程序相比，许多分析本身往往相当快，因此在图处理应用程序中有效使用图分区程序的时间和可伸缩性要求更加严格。本质上，针对新兴图分析的分区方法应该比当前最先进的方法快得多，支持多个目标指标，并且在大规模的不规则结构输入上有更好的扩展性。作者还希望该方法(1)比传统的分区方法(用于科学计算)更有效的内存;(2)具有良好的强伸缩性能，因为可以处理固定大小的问题;(3)相对容易实现，(4)不需要调优。最近在这种分区方法方面已经取得了一些进展。有使用随机或基于边的分布的方法，基于标签传播的图分区方法，传统分区方法对小世界图实例的适应性，以及分布式图框架的适度规模的方法。万亿边缘尺度的图问题最近才开始尝试，最近的工作实现了万亿边缘尺度的基于边的划分。

Algorithm 1. XtraPuLP Overview: $parts \leftarrow$ PuLP-MM $(G_g(V_g, E_g), p, I_{\text{out}}, I_{\text{bal}}, I_{\text{ref}}, Imb_v, Imb_e)$

```
G(V,E) ← distributeGraph(G_g(V_g,E_g))   ▷read and distribute input graph to MPI ranks
I_tot ← I_out × (I_bal + I_ref)          ▷total per-phase iteration count
iter_tot ← 0                             ▷running per-phase iteration count
parts ← XtraPuLP-Init(...)
for i = 1 ... I_out do
  parts ← XtraPuLP-VertBalance(...)
  parts ← XtraPuLP-VertRefine(...)
iter_tot ← 0                             ▷reset for next phase
for i = 1 ... I_out do
  parts ← XtraPuLP-EdgeBalance(...)
  parts ← XtraPuLP-EdgeRefine(...)
```

标签传播团体检测算法是一种快速、可扩展的大型网络团体检测方法。该算法的工作原理如下:图中的所有顶点都被初始化到它们自己的社区中，在每次迭代中，一个顶点将自己置于社区中，与它的多个邻居一起，并随机打破连接。由于它相当容易并行化，并且其优化目标与平衡切边的目标相似，标签传播作为一种寻找小世界和不规则网络的高质量分区的有效方法，已经被广泛采用。在分区器中使用标签传播有两种主要方法。

在算法 1 中给出了 XTRAPULP-MM 算法。作为该算法的主要输入，我们有未实现的全局视图对于一个图$G_g(V_g, E_g)$。对每个 MPI 等级赋一个$G(V, E)$的局部图表示;其中包括一些顶点的不相交划分和它们的一跳邻域边。该算法的附加输入包括划分数量(p)、目标不平衡约束和迭代计数。这和下面的算法描述都是从单一等级的角度给出的。

这个算法有三个阶段。第一阶段是快速初始化策略，它允许部分之间存在一些不平衡(XTRAPULP-Init)。它可以被认为是无约束标签传播和基于宽度优先搜索的图增长两种共享内存 PULP 初始化策略的混合。具体来说，我们从给定的p个种子集合执行一个水平同步的宽度优先搜索。顶点 v 是通过一个随机的绘图来解析的，这个绘图的权重是 v 在每个部分中的邻居数。

第二阶段平衡每个部分的顶点数量，同时最小化切割边的全局数量(XTRAPULP-VertBalance, XTRAPULP-VertRefine)。我们分别将 Imbv 和 Imbe 定义为顶点和边方面的目标最大零件尺寸(例如，$Imb_v = (1 + Rat_v) \times \frac{|V|}{p}$)。

第三阶段平衡顶点和边，最小化全局切边和任意部分的最大切边(XTRAPULP-EdgeRefine)。在之前的工作中，作者已经注意到，在尝试解决更难的边缘平衡问题之前，用较低的切割来解决较容易的顶点平衡问题，往往会在切割方面产生比其他方向处理更好的最终解决方案质量。然而，首先平衡边，然后再平衡顶点，对于具有倾斜度分布的网络来说，可以导致更好的最终不平衡，因为前一种方法不能有效地重新平衡部分。请注意，我们本质上不能保证任何一种方法都能实现不平衡约束。

在作者的实验中，在后两个阶段(Iout)的每个阶段中交替进行 3 次外部迭代的平衡和细化。平衡算法运行 5 次迭代，而优化算法运行 10 次迭代。这些迭代计数是根据经验选择的，以在计算时间和分区质量之间提供一个合理的折衷。对于某些应用程序，这些迭代可以被修改;将这样做的能力作为参数包含在独立的可执行文件和库接口中。标签传播通常为固定的迭代计数而运行，因为已经观察到收敛并不能快速保证，收敛也不一定保证更高的解决方案质量。将该算法推广到任意数量约束的情况。

4 总结与展望

图划分在分布式大规模网络图分析中起着至关重要的作用，如页面排名和标签传播。分区策略的质量和可伸缩性对这种通信和计算密集型应用程序有着强烈的影响，因为它驱动了分布式计算节点之间的通信成本和工作负载平衡。对于静态图，图划分算法主要分为离线算法和在线算法，对于现实世界的动态图，其划分算法主要采用动态划分算法。现实世界图的动态变化主要分为两个方面：一是内部动态性，即参与计算的图数据是动态变化的。内部动态性的变化使得初始的图划分可能产生负载倾斜。二是外部动态性，真实世界的实体之间的关联关系本身处于动态变化之中，这种动态变化会引起图的拓扑结构发生变化，破坏原有的划分结果。

本文分析了 3 篇研究动态图划分的文章。第一篇文章针对图的外部动态性，提出了一种边组迁移的方法，通过将一组边迁移到其他的划分部分来减少图动态变化引起的负载不均衡。第二篇文章提出了一种混合算法，即内存划分与流式划分混合的划分算法来提高划分的效率。第三篇文章通过研究网络图的聚类性质，提出了一种顶点划分算法。

未来图划分算法的研究方向仍然在于如何提高划分质量。由于现实世界的图的变化存在一定规律，所以利用这些规律来优化划分算法是未来的一个研究方向。

参 考 文 献

[1] H. Li et al., Group Reassignment for Dynamic Edge Partitioning, IEEE Transactions on Parallel and Distributed Systems, 2021, 32(10), pp. 2477-2490.

[2] Amel Awadelkarim and Johan Ugander. Prioritized Restreaming Algorithms for Balanced Graph Partitioning. Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. New York, USA, 1877–1887.

[3] G. M. Slota, C. Root, K. Devine, K. Madduri and S. Rajamanickam, "Scalable, Multi-Constraint, Complex-Objective Graph Partitioning," in IEEE Transactions on Parallel and Distributed Systems, 2020, 31(12), pp. 2789-2801.

1