引文格式:亢晓琛,刘纪平.图划分支持下的大规模点要素并行缓冲分析方法[J].武汉大学学报(信息科学版),2023,48(6):979-987.DOI:10.13203/j.whugis20210011

Citation: KANG Xiaochen, LIU Jiping. Parallel Buffer Analysis of Large Scale Point Features Based on Graph Partitioning[J]. Geomatics and Information Science of Wuhan University, 2023, 48(6):979-987.DOI:10.13203/j.whugis20210011

图划分支持下的大规模点要素并行缓冲分析方法

亢晓琛[1] 刘纪平[1,2]

1 中国测绘科学研究院,北京,100036

2 河南省科学院地理研究所,河南 郑州,450052

摘 要:缓冲分析是解决邻近度问题的基础工具,由于算法本身包含大量的复杂运算,处理效率亟待优化。针对大规模点要素的缓冲分析,引入图表达建立了面向数据和分析过程的空间计算域,通过图划分实现了任务的均衡分割。图式化的空间计算域首先从图节点和图边两个角度定义了点要素及其空间关系的处理函数,然后对相应的时间复杂度进行拟合,获取了图节点和图边的计算权重,最后利用图划分方法实现了缓冲分析的均衡分割,从而构建与计算资源相匹配的并行任务。实验结果表明,基于图划分实现的并行缓冲分析方法在负载均衡性和整体性能方面优于主流的四叉树和规则格网划分方法,可为大规模矢量数据的空间分析优化提供参考。

关键词:矢量数据;缓冲分析;空间计算域;图划分

中图分类号:P208 **文献标识码**:A **收稿日期**:2022-12-12

DOI:10.13203/j.whugis20210011 **文章编号**:1671-8860(2023)06-0979-09

Parallel Buffer Analysis of Large Scale Point Features Based on Graph Partitioning

KANG Xiaochen[1] *LIU Jiping*[1,2]

1 Chinese Academy of Surveying and Mapping, Beijing 100036, China

2 Institute of Geographical Sciences, Henan Academy of Sciences, Zhengzhou 450052, China

Abstract: **Objectives**: Buffer analysis is a common tool of spatial analysis, which deals with the problem of proximity. Due to numerous and complex operations in the algorithm, the computational efficiency needs to be optimized. **Methods**: To process large scale point features, a graph-based representation model is proposed, which establishes the spatial computational domain for data and analysis, and develops a well-balanced task-partitioning method by partitioning the graph. First, the proposed model defines processing functions of point features and their spatial relationships from the perspectives of graph nodes and graph edges, and provides a logic description for buffer zone generation around point features. Second, the computational weights of graph nodes and graph edges are obtained by fitting the time complexity of the above processing functions. Finally, graph partitioning is adopted to divide the buffer task, which contributes to multiple parallel tasks matching with the computational resources. **Results**: The experimental results show that graph-based buffer analysis can achieve better load balance and overall efficiency, which is superior to the mainstream partitioning methods, regular-grid and quadtree. **Conclusions**: The proposed method can provide a reference for optimization of spatial analysis methods when processing large scale vector data.

Key words: vector data; buffer analysis; spatial computational domain; graph partitioning

随着数据采集精度的提高和采集周期的缩短,各类地理数据的空间分辨率与时间分辨率不断提升,数据量和计算复杂度已成为数据探索、分析、理解和呈现的巨大挑战[1-2]。在大数据背景

基金项目:国家自然科学基金(41701461);中国测绘科学研究院基本科研业务费项目(AR2001)。

第一作者:亢晓琛,博士,副研究员,主要从事高性能地理计算和地理国情监测理论与方法研究。kangxc@casm.ac.cn

下,矢量数据模型已成为空间分析、空间决策支持、空间数据挖掘等分析应用的重要数据组织形式,而这些应用一般需要超强的计算能力支持[3]。矢量要素间的各类空间关系,如距离[4]、拓扑[5]、方位[6]等,已成为数据管理、统计、插值、分析、查询等处理的基础[7-8]。利用空间关系可以对算法逻辑的内在依赖关系进行不同强度的描述[9-10],这种依赖关系在很大程度上可归因于空间相关性、空间异质性等基本规律,这为一定范畴内的复杂空间问题求解提供了分治优化的思路[11-12]。近年来,学者们相继提出了可优化矢量空间分析的并行计算框架,如CyberGIS[13]、CudaGIS[14]、HadoopGIS[15]、Streaming-RCUW[16]等。相关技术框架一般包括空间划分与计算调度两个核心部件,而对数据或任务实施均衡分割一直是实现算法优化的关键。目前,主流的空间划分方法多基于四叉树、规则格网、空间填充曲线等结构来实现。这些方法逻辑简单、易于实现,对栅格数据具有较好的适用性,但处理矢量数据时容易出现数据偏斜、边界对象冗余分配等现象,进而造成计算负载失衡、子任务通信时间过长等问题[15,17]。

在地理信息科学中,缓冲分析是以点、线、面要素为基础,在其周围建立一定宽度(即缓冲半径)的多边形,然后结合需求对重叠区域进行融合处理。多边形融合的本质是图形并集运算,该过程一般耗时较长[18],较大的缓冲半径甚至需要对全部结果进行融合。在缓冲分析处理中,任意两个不同要素的输出结果是否需要融合取决于二者之间的空间距离与缓冲半径的关系,这直接决定了要素之间是否存在输出竞争关系。因此,在对缓冲分析算法进行并行优化时,必须对可能存在的输出冲突进行预先判别,进而研究适宜的空间划分策略。现有的缓冲分析并行优化研究,如基于区域合并的并行缓冲算法[19]、面向可视化渲染的实时缓冲算法[20]、计算强度网格支持的矢量并行缓冲算法[21]等,通过采用规则格网、四叉树等划分方法取得了一定的性能提升,但仍无法避免数据偏斜和边界对象冗余分配问题。经分析可知,缓冲分析的计算负载随着缓冲半径变化呈现不同的分布特征,且计算强度取决于待融合的多边形数量,这是传统空间划分方法难以顾及的。

本文以点要素的缓冲分析为研究对象,通过预先计算点对空间距离快速筛选各点要素在一定缓冲半径内的关联要素,采用加权图对点要素及其关联关系的计算强度进行定量描述,进而利用图划分实现任务的均衡分割。由于图结构可以根据输入参数动态表达依赖关系而无须考虑矢量要素本身的几何形状,从而规避矢量要素随机分布对空间划分的干扰,这可为其他复杂空间分析的并行优化提供重要参考。

1 缓冲分析依赖关系

1.1 数据依赖性定义

在计算机科学中,数据依赖性是指计算指令间的引用制约关系。当满足如下条件之一,则认为指令j数据依赖于另外一个指令i[22]:

1)指令i输出的结果被指令j使用。

2)指令j数据依赖于指令k,而指令k数据依赖于指令i。

根据Bernstein条件[23],数据依赖性可描述为:

$$[I(S_i)\cap O(S_j)]\cup[O(S_i)\cap I(S_j)]\cup[O(S_i)\cap O(S_j)] \quad (1)$$

式中,$I(S)$代表指令S读取的数据位置;$O(S)$代表指令S写入的数据位置;$I(S_i)\cap O(S_j)$定义了反依赖性,即S_i在S_j写入某数据位置之前完成数据读取;$O(S_i)\cap I(S_j)$定义了流依赖性,即S_i写入某数据位置后S_j完成数据读取;$O(S_i)\cap O(S_j)$定义了输出依赖性,即S_i与S_j写入相同的数据位置。当一个以上指令同时写入相同位置或读与写同时发生在相同数据位置时,必须协调指令的先后顺序以解决并发冲突。

1.2 距离依赖关系提取

在点要素缓冲分析中,不同点要素的距离如果小于二倍缓冲半径,将产生重叠的多边形区域,即指令写入相同的结果多边形。因此,在一定的缓冲半径下,应通过距离比较判断不同点要素间是否存在输出依赖关系。为加速距离依赖关系提取,提出一种格网索引来辅助搜索各点要素的关联要素。点要素距离依赖关系搜索如图1所示。首先根据数据范围构建一定宽度的规则格网,计算各点要素的格网归属;然后以各点要素为查询位置,通过逆时针螺旋搜索快速筛选可能包含关联点要素的格网;最后计算查询位置与筛选格网内全部点要素的空间距离,记录小于或等于二倍缓冲半径的全部点要素,形成距离依赖关系集合。

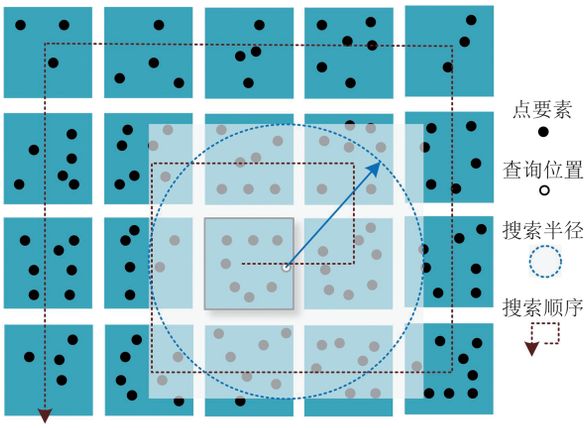


图 1　点要素距离依赖关系搜索

Fig. 1　Searching for Distance Dependent Relationships of Point Features

2　图表达与空间划分

2.1　缓冲分析图表达

根据文献[24],空间计算域被定义为描述地理计算强度分布的二维栅格空间,通过各位置的取值变化反映时间和空间复杂度的分布差异。但是,空间计算域忽视了矢量要素本身及计算所具有的空间分布特征,难以实现细粒度的精确划分。

点要素缓冲分析中,计算负载主要分布于两个部分:一是点要素本身缓冲多边形的生成,二是关联点要素缓冲多边形的融合计算。对计算强度的度量一般可借助时间复杂度来完成,以定量描述执行算法所需要的实际负载。根据缓冲分析的依赖关系,提出一种由图顶点-顶点函数-权重函数和图边-边函数-权重函数组成的计算转换方法,形成完整描述点要素缓冲分析的图表达模型(buffer graph representation, BGR),表达式为:

$$\mathrm{BGR}=(V(i,f_v(i),w_v(i)),E(j,f_e(j),w_e(j))) \quad (2)$$

式中,BGR 由节点集合 V 和关系边集合 E 组成;V 用于描述参与计算的点要素集合及其处理方法;E 用于描述缓冲分析中点要素之间的依赖关系及其处理方法。V 中各节点抽象为一个三元组,即节点标识 i、节点函数 f_v、节点权重函数 w_v,其中 f_v 为 V 中各要素的处理函数,w_v 为 f_v 在各要素上的计算强度评估。类似的,E 中各边抽象为一个三元组,即边标识 j、边函数 f_e、边权重函数 w_e,其中 f_e 为 E 中各依赖关系的处理函数,w_e 为 f_e 在各关系边上的计算强度评估。

综上,BGR 提供了描述缓冲分析依赖关系与计算负载分布的图表达结构,利用确定的数据与缓冲半径可以完成 BGR 的实例化,具体流程见图 2。

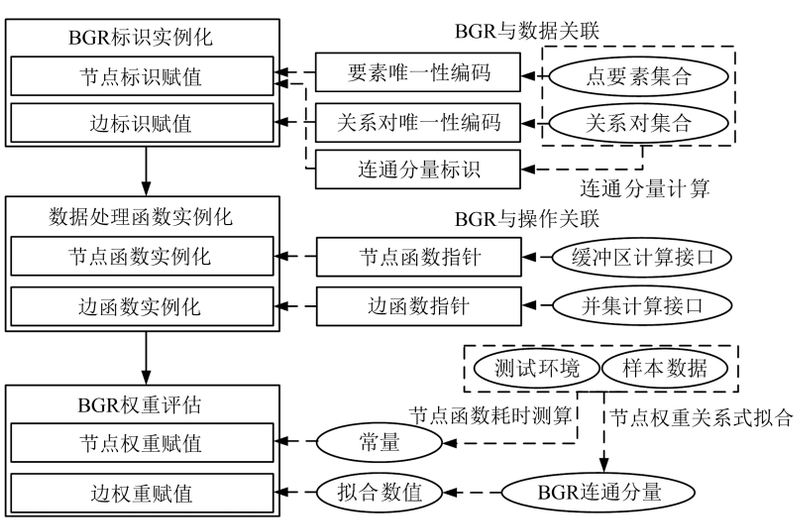


图 2　BGR 实例化流程

Fig. 2　Procedure for BGR Instantiation

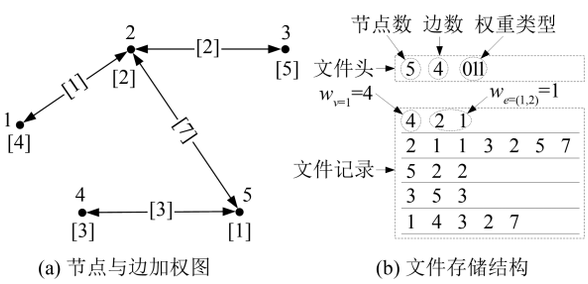
V 集合中,节点标识选用集合中各点要素的唯一性编码,节点函数 f_v 可根据需要选择成熟的算法,本文选用了几何引擎开源库(geometry engine open source, GEOS)中的缓冲区计算接口。单个点要素的缓冲区处理时间相对固定,节点权重评估函数 w_v 可视为常函数。E 集合中,边标识按提取顺序进行编码,边函数 f_e 选用了 GEOS 中的并集计算接口。在 BGR 中,所有距离小于或等于二倍缓冲半径的节点均存在连接边,整体上会形成一个或多个连通子图。换言之,通过识别 BGR 中的连通子图,可筛选出一个或多个需要进行融合处理的多边形集合。在计算多边形并集

时,实际耗时主要由参与计算的多边形坐标数决定。由于单个点要素的缓冲多边形坐标数相对固定,在对多个相同的多边形进行融合时,可以根据当前连通子图的大小对w_e值进行估计。事实上,w_v与w_e的实际耗时还与硬件环境密切相关,难以采用普适的方法进行精确定权,本文采用时间拟合函数进行预测,即通过样本数据预先测定当前环境下数据量参数与标准时间的拟合关系,进而估计实际数据的耗时情况。

2.2 BGR划分

在数学中,图划分(graph partitioning,GP)是将图形式表达的数据结构进行切割,以得到具有特殊性质的较小子图[25],一种典型的应用是将规模较大的图切割为大小近似相等的多个子图,从而为并行计算提供解决思路[26]。图划分是经典的NP-Hard问题,通常难以在多项式时间内获得最优解[27]。20世纪90年代以来,国内外学者提出多种图划分优化方法,如谱划分[28]、几何划分[29]、启发式方法[30]、多级划分[31]等。其中,谱划分通过求解矩阵的特征向量实现分割,对多类问题具有较好的适用性,但算法计算量过大,不适于处理规模较大的图划分问题。几何划分是从分布空间对图进行水平或垂直方向切分,得到多个子区域,优点是速度快,缺点是划分质量较低。启发式方法是在初始的随机划分基础上,通过局部贪心策略进行迭代优化来达到划分目的,该方法处理大图时计算性能较差。多级划分方法采用了迭代削减节点与边的策略来逐渐减小图的规模,然后通过逆向细化达到划分原图的目的。在处理规模较大的图时,多级划分方法可以同时获得较为理想的负载均衡性与处理效率,已成为产业界最常用的图划分方法之一[31]。

为区别于连通子图,本文将划分后的子图定义为划分子图。由于图的划分不受空间位置与几何形状的限制,可以避免传统空间划分方法对数据及其依赖关系的破坏。BGR本质是加权无向图,通过对其切割可间接实现缓冲分析的分治处理。本文采用了工业级强度的开源算法包Metis提供的多级划分方法,实现BGR均衡切割。Metis提供了加权图的存储结构,可实现BGR节点、关联边及其对应权重值的格式化存储,具体如图3所示。图3(a)为包含节点权重和边权重的BGR,由5个节点和4条边组成。图3(b)为文件存储结构,首行描述了节点数、边数及权重类型(011代表节点和边同时具有权重),第i行($i\geqslant 1$)存储第$i-1$个节点对应的权重值和成对存在的关联边信息。

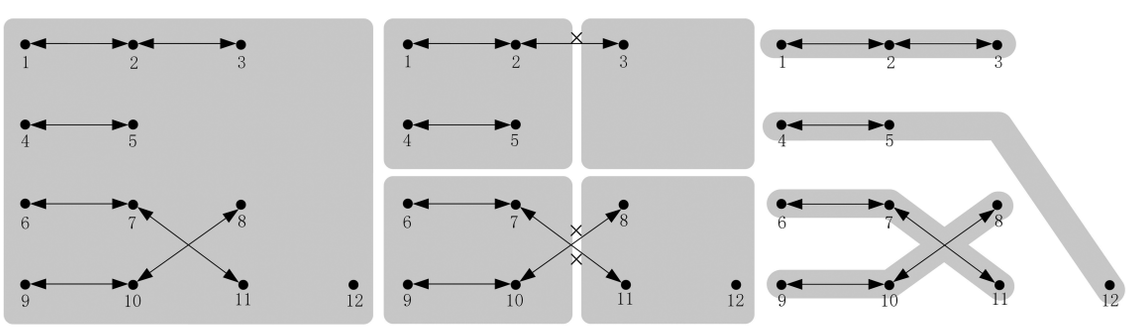


(a) 节点与边加权图 (b) 文件存储结构

图3 Metis加权图存储格式

Fig. 3 Storage Format for Metis Weighted Graph

规则格网划分与多级划分对比如图4所示。利用多级划分方法对BGR切分,一方面可以确保多个子任务的计算量相对均衡,另一方面可以尽量减少割边数,降低子任务间的通信代价[31]。图4(a)是包含12个点要素及其依赖关系的BGR,采用规则格网划分后会产生较多的割边(图4(b)),而利用多级划分方法则可以尽可能避免(图4(c))。



(a) BGR (b)规则格网划分结果 (c) 多级划分结果

图4 规则格网划分与多级划分对比

Fig. 4 Comparison of Regular-grid and Multi-level Partitioning

2.3 并行计算与融合

在对BGR实施划分后,可得到与计算资源相匹配的多个划分子图,每个划分子图对应一个可并行子任务。BGR中可能包含一个或多个连通子图,同一连通子图内的节点处理结果需要融合,不同连通子图之间则无任何依赖关系。因此,同

一连通子图内的节点如果被分配至不同的划分子图,则应对相应子任务的计算结果进行二次融合。根据BGR划分结果,并行资源会首先调用f_v处理各划分子图内的点要素,然后调用f_e处理划分子图内的依赖关系,最后处理子图之间的割边。

本文采用一种初始状态与BGR同构的日志图BGR-Log来记录与更新图节点与图边的标识状态,进而引导并行计算与结果融合。BGR-Log包括节点集合、边集合、缓冲半径、子任务集合、连通子图集合、中间结果集合与最终计算结果7个基本参数,计算过程包括内部计算、割边标记与外部计算3个阶段。

1)内部计算

在子任务计算过程中,根据划分子图处理进度,实时合并BGR-Log节点,待子任务处理结束后,各划分子图内具有相同连通子图标识的节点全部合并为一个节点。

2)割边标记

内部计算完成后,BGR-Log中每个节点代表了同一子任务内部的连通子图,此时,BGR-Log中具有相同连通子图标识的关系边全部为割边,应进一步对子任务处理结果进行外部融合计算。

3)外部计算

BGR-Log中割边代表了未完成的融合计算,该任务仍具有较高的计算强度。通过顺序拾取BGR-Log中的非共享节点边可筛选出当前无输出竞争关系的计算任务,进而利用并行线程实现二次融合。非共享节点边拾取包括3个步骤:(1)创建非共享节点边集合,从BGR-Log中随机选取一条边作为起始边;(2)更新非共享节点边集合,循环判断BGR-Log中各剩余边是否与当前的非共享节点边存在共享节点,如不存在则纳入集合;(3)驱动计算,对非共享节点边集合所标识的任务进行并行处理,并将BGR-Log中对应节点合并。循环执行步骤(2)与(3),至BGR-Log中不存在关系边结束任务。

基于BGR-Log的缓冲融合区并行生成如图5所示,其中BGR-Log包含2个连通子图,通过迭代拾取非共享节点边和并行融合处理,最终得到与连通子图个数相同的结果多边形。

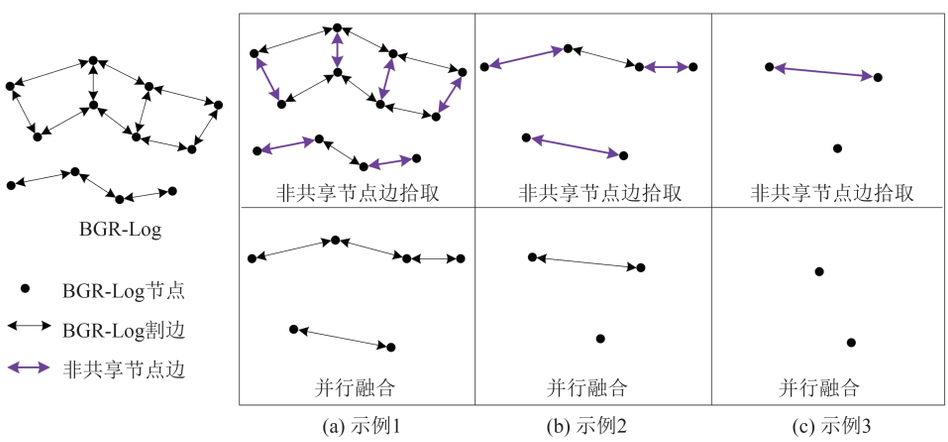


图5　基于BGR-Log的缓冲融合区并行生成

Fig. 5　Parallel Dissolved Buffer Zone Generation Based on BGR-Log

3　实验与分析

3.1　实验数据与环境

在自然资源统计分析中,行政村点位通常作为学校、医院、社会福利机构等设施邻近度评估的基础,分析半径通常设置为500～10 000 m,形成缓冲区后再与相关设施点位进行叠加分析。该过程中缓冲区生成最为耗时,且难以直接并行化处理,利用BGR表达与划分可以解决大规模行政村点位缓冲区的快速生成难题。为测定BGR中w_v与w_e,本文构建了包含10 000个规则分布点要素的模拟数据,为降低点要素间距变化对计算耗时的影响,水平和垂直间距均设置为1 000 m。真实数据选用了第一次全国地理国情普查获取的31个省(市、区)的行政村点位数据,共计680 942个,整体呈东南稠密、西北稀疏的偏斜分布特征。

测试环境为单个节点的联想工作站,配置包含2个至强CPU(主频2.80 GHz,共计20个物理核心,支持40个超线程),32 GB内存,4 TB硬盘存储。操作系统为Windows 7,编程语言为C++。

3.2　权重计算

单个点要素的缓冲区生成及少量简单多边形的融合计算均耗时较低,难以准确测算。针对模拟数据,实验以100次运算作为计时单元,缓冲

区生成仅耗时9.85 s,得到w_v=0.098 5。按照多边形融合计算的最高复杂度$T=O(n^2)$(n为多边形边数)估计,算法的处理时间取决于参与计算的多边形边数,融合时间在理论上可以根据数据规模进行拟合。根据点个数与模拟数据测试结果,经最小二乘法拟合得到w_e。拟合函数如图6所示,w_e与实际耗时的拟合优度达到0.998 9。

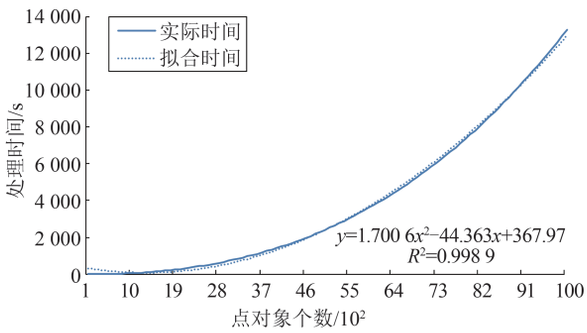


图6 等间距点要素缓冲区融合的拟合函数时间

Fig. 6 Fitting Function Time of Dissolved Buffer Zone Generation with Equidistant Point Features

3.3 空间划分性能

3.3.1 空间划分时间

一般情况下,空间划分数由可并行计算资源决定,如CPU核心线程数。为实施空间划分性能对比,分别实现了规则格网与四叉树划分方法,统计了5种不同划分数(8、16、32、64、128)下的空间划分时间,并以最大划分数128为例,对比了3类空间划分方法的负载均衡性。

图7为5种缓冲半径下3类空间划分方法的性能对比情况。从变化趋势来看,规则格网划分效率最高,平均5 s内可完成一次数据划分。四叉树划分效率最低,且随划分数增加耗时呈显著增加趋势,划分数达到128时平均耗时超过65 s。BGR划分耗时介于规则格网划分与四叉树划分之间,且当缓冲半径较大时与规则格网划分较为接近。

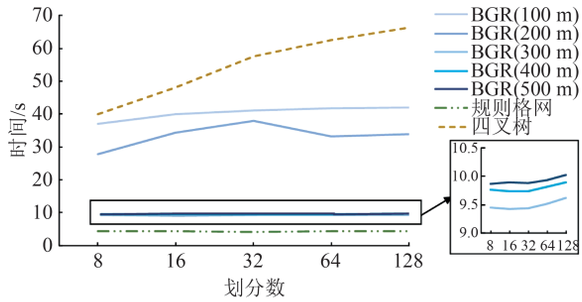
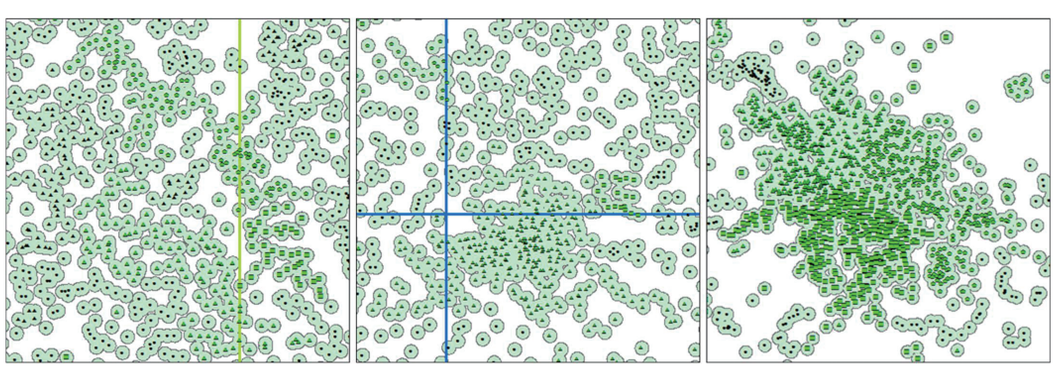


图7 不同缓冲半径下3类空间划分方法时间对比

Fig. 7 Partitioning Time Comparison of Three Methods with Different Radii

规则格网划分须构建与数据范围相同的系列格网单元,进而通过遍历一次数据判别每个点要素的空间归属来完成划分。BGR划分耗时主要包括BGR构建与图划分两部分。BGR构建中采用了格网索引,通过过滤与判别计算可快速提取任意点要素在一定邻域半径内的关联要素,测试中平均耗时3~4 s。测试还发现,BGR划分耗时与图中的连通子图数有关,随着连通子图的减少,多级划分方法的迭代处理时间逐渐减少。当缓冲半径大于300 m时,BGR划分耗时基本维持在10 s左右。四叉树划分采用迭代思路,对当前划分中点要素最多的子空间进行二次划分,直至子空间个数满足要求为止,整体耗时随划分数增加而延长。

图8为500 m缓冲半径下BGR划分的局部效果,其中同一种符号代表相同的划分归属。四叉树划分(图8(a))与规则格网划分(图8(b))将同一连通子图内的节点分配至不同的划分子图,划分策略存在一定的盲目性;BGR划分(图8(c))规避了规则边线对依赖关系边的盲目切割现象,使得BGR中同一连通子图的节点尽可能分配至同一划分子图内,而部分过大的连通子图仍会被切分以实现负载均衡的核心目标。



(a) 四叉树划分 (b) 规则格网划分 (c) BGR划分

图8 不同划分方法局部效果图

Fig. 8 Local Effects of Different Partitioning Methods

3.3.2　空间划分均衡性

图 9 为 5 种缓冲半径下 3 类空间划分方法对应的划分时间、最长子任务时间与割边处理时间。为避免计算资源竞争对时间统计的干扰,实验采用串行方式对各子任务的耗时及割边处理时间进行单独统计。从各组统计结果来看,规则格网划分下最长子任务耗时最长,均衡性最差,其次为四叉树划分,BGR 划分下最长子任务耗时最短。根据划分时间、最长子任务时间与割边处理时间的合计结果分析,BGR 划分下的并行缓冲分析时间应低于规则格网与四叉树划分,且随着缓冲半径的增加,计算优势更加明显。

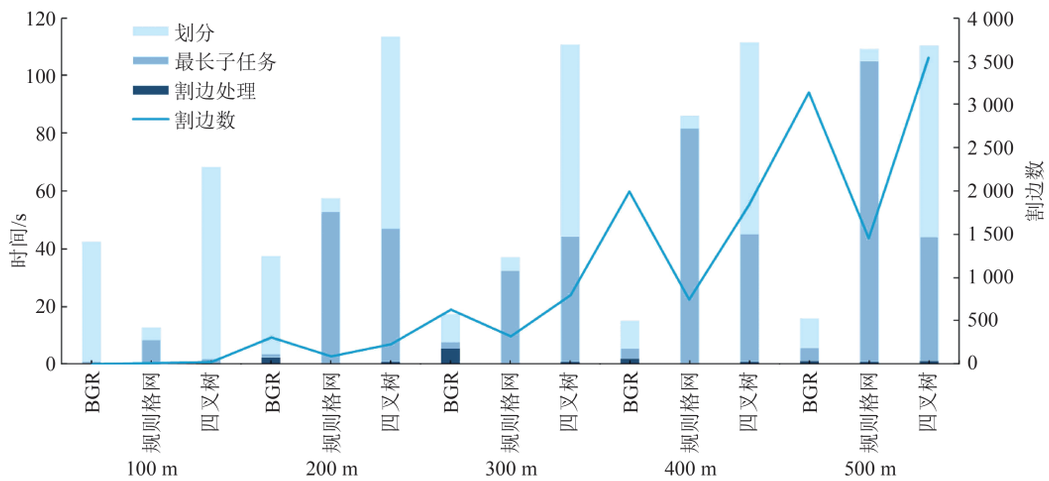


图 9　空间划分均衡性与割边处理时间

Fig. 9　Balance of Spatial Partitioning and Cut-Edge Processing Time

3.4　整体性能

通过与规则格网、四叉树两种划分方法进行对比,初步验证了 BGR 划分方法的有效性。在当前测试环境下,构建了 20 种不同缓冲半径的 BGR,并根据 CPU 核心线程数将各 BGR 划分为了 40 个并行子任务,计算完成后,再根据 BGR-Log 对割边进行并行处理,测试结果见表 1。

由表 1 可知,随着缓冲半径的增加,融合比例

表 1　多种半径下串行 BGR 与并行 BGR 计算时间

Tab. 1　Computation Time of Serial BGR and Parallel BGR with Different Radii

序号	缓冲半径/m	结果多边形数量	融合比例/%	串行时间/s	并行时间/s	加速比
1	50	672 151	1.29	106.47	81.65	1.30
2	100	660 333	3.03	109.74	92.01	1.19
3	150	638 495	6.23	116.10	103.21	1.12
4	200	610 145	10.40	127.32	112.67	1.13
5	250	579 401	14.91	140.21	94.98	1.48
6	300	548 306	19.48	152.51	98.84	1.54
7	350	516 911	24.09	165.01	102.51	1.61
8	400	484 913	28.79	179.45	112.78	1.59
9	450	452 125	33.60	194.92	118.52	1.64
10	500	419 060	38.46	209.65	120.52	1.74
11	550	385 672	43.36	233.66	127.92	1.83
12	600	352 021	48.30	289.91	138.79	2.09
13	650	319 934	53.02	392.93	148.13	2.65
14	700	289 453	57.49	1 038.99	296.12	3.51
15	750	261 232	61.64	2 925.87	409.83	7.14
16	800	235 772	65.38	6 883.81	576.84	11.93
17	850	212 723	68.76	18 782.26	1 082.78	17.35
18	900	192 058	71.80	26 595.51	1 392.72	19.10
19	950	173 827	74.47	31 561.23	1 541.99	20.47
20	1 000	157 414	76.88	36 191.83	1 518.41	23.84

逐渐升高,结果多边形个数逐渐减少,整体计算耗时相应增加。对比串行与并行模式发现,随着缓冲半径的增加,加速比得到稳定提升,这得益于空间划分、割边处理及数据读写的耗时占总体时间的比例呈降低趋势。当缓冲半径增加至800 m时,加速比达到10倍以上;当缓冲半径增加至1 000 m时,加速比提升至23倍左右。与ArcGIS软件的Buffer工具对比发现,串行模式下BGR方法的计算性能至少达到ArcGIS软件的25倍以上。当缓冲半径超过700 m时,ArcGIS抛出异常错误。

为进一步验证BGR方法对高强度运算的适应性,分别以10 000 m、20 000 m、40 000 m为缓冲半径进行测试,结果见表2。根据测算,此时99%以上的点要素满足距离依赖关系。从表2可知,大半径的融合计算强度极高,并行模式下BGR方法的加速比分别达到22.80、22.67、23.21。

表2 BGR高强度运算测试

Tab. 2 Intensive Computation Test of BGR

序号	缓冲半径/m	串行时间/s	并行时间/s	加速比
1	10 000	54 516.13	2 391.54	22.80
2	20 000	56 593.58	2 496.18	22.67
3	40 000	133 143.17	5 735.34	23.21

4 结 语

点要素的缓冲区为简单多边形,对于更复杂的线状、面状要素而言,可考虑利用多边形的边数完成w_v和w_e测算。从空间划分结果来看,BGR划分在寻求子任务均衡的前提下,尽可能减少割边的产生,划分结果的外部形态呈集聚分布,这与BGR中的连通子图保持一致。从空间划分耗时来看,BGR划分的时间代价介于规则格网与四叉树划分之间,并行子任务的最长处理时间明显低于这两种方法,但割边数及相应处理时间略高。这是因为BGR划分会主动切割一些较大的连通子图,将部分计算量由较大子任务迁移至较小子任务,以达到负载均衡目的,从而导致部分高权重关系边被切割。

依赖关系分析是实现复杂空间分析分治优化的关键,采用图式化空间计算域的定量表达与划分可实现一种面向并行计算资源的空间划分方法。针对大规模点要素缓冲分析的高耗时问题,本文设计与实现了面向距离依赖关系的空间计算域BGR及其分治处理方法。实验中,BGR划分的负载均衡性优于传统的规则格网与四叉树划分方法,在一定程度上解决了矢量数据分布偏斜造成的负载失衡问题。本文方法可推及至拓扑、方向等类型的空间关系计算,并用于相关算法的并行优化。

参 考 文 献

[1] Li Qingquan, Li Deren. Big Data GIS[J]. *Geomatics and Information Science of Wuhan University*, 2014, 39(6): 641-644. (李清泉, 李德仁. 大数据GIS[J]. 武汉大学学报(信息科学版), 2014, 39(6): 641-644.)

[2] Liu Jiping, Dong Chun, Kang Xiaochen, et al. National Geographical Conditions Statistical Analysis in the Era of Big Data[J]. *Geomatics and Information Science of Wuhan University*, 2019, 44(1): 68-76. (刘纪平, 董春, 亢晓琛, 等. 大数据时代的地理国情统计分析[J]. 武汉大学学报(信息科学版), 2019, 44(1): 68-76.)

[3] Zhao Chunyu. Studying on the Technolgies of Storage and Processing of Spatial Vector Data in High-performance Parallel GIS[D]. Wuhan: Wuhan University, 2006. (赵春宇. 高性能并行GIS中矢量空间数据存取与处理关键技术研究[D]. 武汉: 武汉大学, 2006.)

[4] Nekola J C, White P S. The Distance Decay of Similarity in Biogeography and Ecology[J]. *Journal of Biogeography*, 1999, 26(4): 867-878.

[5] Egenhofer M J. A Formal Definition of Binary Topological Relationships[M]//Foundations of Data Organization and Algorithms. Berlin, Heidelberg: Springer, 1989: 457-472.

[6] Guo Renzhong, Chen Yebin, Zhao Zhigang, et al. Scientific Concept and Representation Framework of Maps in the ICT Era[J]. *Geomatics and Information Science of Wuhan University*, 2022, 47(12): 1978-1987. (郭仁忠, 陈业滨, 赵志刚, 等. ICT时代地图的科学概念及表达框架[J]. 武汉大学学报(信息科学版), 2022, 47(12): 1978-1987.)

[7] Huang Liangke, Li Chen, Xie Shaofeng, et al. Spatial Interpolation of Atmospheric Weighted Mean Temperature Grid Products in China with Consideration of Vertical Lapse Rate[J]. *Geomatics and Information Science of Wuhan University*, 2023, 48(2): 295-300. (黄良珂, 李琛, 谢劭峰, 等. 顾及垂直递减率的中国区域Tm格点产品空间插值[J]. 武汉大学学报(信息科学版), 2023, 48(2): 295-300.)

[8] Ying Shen, Jin Fengzan, Li Lin, et al. Hierarchical Block of Vector Data Based on ArcGIS Engine[J]. *Journal of Geomatics*, 2014, 39(6): 50-53. (应申,

靳凤攒, 李霖, 等. 基于ArcGIS Engine的矢量数据分层分块技术研究[J]. 测绘地理信息, 2014, 39(6): 50-53.)

[9] Longley P A, Tobón C. Spatial Dependence and Heterogeneity in Patterns of Hardship: An Intra-urban Analysis[J]. *Annals of the Association of American Geographers*, 2004, 94(3): 503-519.

[10] Rodríguez A M, Egenhofer M J. Comparing Geospatial Entity Classes: An Asymmetric and Context-dependent Similarity Measure[J]. *International Journal of Geographical Information Science*, 2004, 18(3): 229-256.

[11] Yang C W, Wu H Y, Huang Q Y, et al. Using Spatial Principles to Optimize Distributed Computing for Enabling the Physical Science Discoveries[J]. *Proceedings of the National Academy of Sciences of the United States of America*, 2011, 108(14): 5498-5503.

[12] Werner M. Parallel Processing Strategies for Big Geospatial Data[J]. *Frontiers in Big Data*, 2019, 2: 44.

[13] Wang S W. A CyberGIS Framework for the Synthesis of Cyberinfrastructure, GIS, and Spatial Analysis[J]. *Annals of the Association of American Geographers*, 2010, 100(3): 535-557.

[14] Zhang J T, You S M. CudaGIS: Report on the Design and Realization of a Massive Data Parallel GIS on GPUs[C]//The 3rd ACM SIGSPATIAL International Workshop on GeoStreaming, Redondo Beach, USA, 2012.

[15] Aji A, Wang F S, Vo H, et al. HadoopGIS: A High Performance Spatial Data Warehousing System over MapReduce[J]. *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*, 2013, 6(11): 1009.

[16] Kang X C, Liu J P, Lin X G. Streaming Progressive TIN Densification Filter for Airborne LiDAR Point Clouds Using Multi-core Architectures[J]. *Remote Sensing*, 2014, 6(8): 7212-7232.

[17] Guo Mingqiang, Xie Zhong, Huang Ying. Content Grid Load Balancing Algorithm for Large-scale Vector Data in the Server Cluster Concurrent Environment[J]. *Geomatics and Information Science of Wuhan University*, 2013, 38(9): 1131-1134. (郭明强, 谢忠, 黄颖. 集群并发环境下大规模矢量数据负载均衡算法[J]. 武汉大学学报(信息科学版), 2013, 38(9): 1131-1134.)

[18] Shen J X, Chen L, Wu Y, et al. Approach to Accelerating Dissolved Vector Buffer Generation in Distributed In-memory Cluster Architecture[J]. *ISPRS International Journal of Geo-Information*, 2018, 7(1): 26.

[19] Fan J F, Ji M, Gu G M, et al. Optimization Approaches to Mpi and Area Merging-based Parallel Buffer Algorithm[J]. *Boletim De Ciências Geodésicas*, 2014, 20(2): 237-256.

[20] Wu Y, Ma M Y, Chen L. HiVewshed: An Interactive Online Viewshed Analysis System for Multiple Observers[C]//The 3rd International Conference on Computer Science and Application Engineering, Sanya, China, 2019.

[21] Guo M Q, Han C D, Guan Q F, et al. A Universal Parallel Scheduling Approach to Polyline and Polygon Vector Data Buffer Analysis on Conventional GIS Platforms[J]. *Transactions in GIS*, 2020, 24(6): 1630-1654.

[22] Hennessy J L, Patterson D A, Arpaci-Dusseau A C. A Quantitative Approach[M]. Boston: Morgan Kaufmann, 2011.

[23] Bernstein A J. Analysis of Programs for Parallel Processing[J]. *IEEE Transactions on Electronic Computers*, 1966, EC-15(5): 757-763.

[24] Wang S W, Armstrong M P. A Theoretical Approach to the Use of Cyberinfrastructure in Geographical Analysis[J]. *International Journal of Geographical Information Science*, 2009, 23(2): 169-193.

[25] Lipton R J, Tarjan R E. A Separator Theorem for Planar Graphs[J]. *SIAM Journal on Applied Mathematics*, 1979, 36(2): 177-189.

[26] Skiena S S. The Algorithm Design Manual[M]. London: Springer, 2008.

[27] Xu Baogang. Graph Partitions: Recent Progresses and some Open Problems[J]. *Advances in Mathematics*, 2016, 45(1): 1-20. (许宝刚. 图的划分:一些进展与未解决问题[J]. 数学进展, 2016, 45(1): 1-20.)

[28] Wieling M, Nerbonne J. Bipartite Spectral Graph Partitioning for Clustering Dialect Varieties and Detecting their Linguistic Features[J]. *Computer Speech & Language*, 2011, 25(3): 700-715.

[29] Plimpton S J, Moore S G, Borner A, et al. Direct Simulation Monte Carlo on Petaflop Supercomputers and Beyond[J]. *Physics of Fluids*, 2019, 31(8): 086101.

[30] Stanton I, Kliot G. Streaming Graph Partitioning for Large Distributed Graphs[C]// The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, USA, 2012.

[31] Wang Xin, Chen Weixue, Yang Yajun, et al. Research on Knowledge Graph Partitioning Algorithms: A Survey[J]. *Chinese Journal of Computers*, 2021, 44(1): 235-260. (王鑫, 陈蔚雪, 杨雅君, 等. 知识图谱划分算法研究综述[J]. 计算机学报, 2021, 44(1): 235-260.)