点群选取与化简算法时间复杂度研究

于艳平[1,2,3],沈　婕[1,2,3],尚在颖[1,2,3]

(1. 南京师范大学地理科学学院,江苏 南京 210046)
(2. 虚拟地理环境教育部重点实验室,江苏 南京 210046)
(3. 地理信息科学江苏省重点实验室,江苏 南京 210046)

[摘要]　点群目标作为地图的基本要素,是普通地图及专题表达的重要内容. 近年来,随着网络地图与移动地图的发展,兴趣点已成为最为重要的表达要素,其数据生产、更新与表达逐渐成为研究热点. 针对点群要素的综合,选取与化简是两种常用的操作. 传统的点群选取与化简算法主要是针对地图的自动生产,因此较侧重于点综合的质量,而随着 GIS 数据实时表达需求的增长和 LBS 服务的发展,对点综合算法的效率提出了更高的要求. 本文在调研了常见点群选取与化简算法的基础上,按照实现原理的不同将算法分类,每一类中分别选取了一种具有代表性的算法,对其时间复杂度进行分析,并初步探讨了这些算法移植到并行计算环境下的可行性. 这一研究将为点群选取与化简算法在网络地图及应急地图服务的应用与拓展奠定基础.

[关键词]　点群要素,选取,化简,算法,时间复杂度
[中图分类号] P208　[文献标志码] A　[文章编号] 1001-4616(2012)01-0111-06

A Study on the Time Complexity of Point Cluster Selection and Simplification Algorithms

Yu Yanping[1,2,3], Shen Jie[1,2,3], Shang Zaiying[1,2,3]

(1. School of Geography Science, Nanjing Normal University, Nanjing 210046, China)
(2. MOE Key Laboratories of Virtual Geographic Environment, Nanjing 210046, China)
(3. Key Laboratory of Geographic Information Science of Jiangsu Province, Nanjing 210046, China)

Abstract: Being a basic element of the map, point feature is the important content of general and thematic map representation. With the development of web maps and mobile maps, point of interest (POI) has become the most important element to be represented, and the production, updating and visualization of POI is becoming a top issue in recent years. Facing to point cluster generalization, the selection and simplification operations are often adopted. However, the traditional point cluster selection and simplification algorithms mainly aim at the automatic production of paper maps, which concern more about the quality of the generalization instead of the efficiency. This may not satisfy the needs of the real-time representation of GIS data and the development of LBS services. In this paper, the previous algorithms of point selection and simplification are reviewed and classified into four categories according to their implementation principles. One representative algorithm of each category are selected to be particularly analyzed for their time complexities. The feasibility of being applied to the parallel environment is also discussed. The study of this paper will lay a foundation for the application and development of the point cluster selection and simplification algorithms in web mapping and emergency mapping services.

Key words: point cluster, selection, simplification, algorithm, time complexity

点群要素是指地图上具有区域景观特征的点群目标,如呈点状分布的散列式居民地、密集分布的用点表示的小岛屿群及小湖泊群等等[1]. 随着地图比例尺的变化,地图上保留点的数目也在变化,而点群的特

收稿日期: 2011-12-06.
基金项目:国家自然科学基金(41071288,41171350).
通讯联系人:沈　婕,博士,副教授,硕士生导师,研究方向:地图自动综合并行计算、电子地图与网络地图设计. E-mail: shenjie@njnu.edu.cn

征信息要尽可能得到保持. 针对点群要素的综合操作主要有选取、化简、典型化、聚合与移位等[2]. 其中,选取与化简是比较常用的操作. 选取与化简都是从初始点群中选取子集,被选取的点位置保持不变,而两者的不同之处在于,选取是基于点对象的专题属性来进行,而化简则依据点群的几何属性如邻近性、密度等来选择[2]. 在实际的综合算法中,通常既考虑点群的专题属性又考虑其几何属性,难以将算法严格地归类为选取或化简算法,因此本文将针对选取与化简操作的综合算法放在一起进行分析.

在点群选取与化简方面,从上世纪80年代开始就涌现出了不少算法,如居民地空间比率算法、重力模型算法、分布系数算法等,这些算法大多是面向地图的自动生产,因此侧重于点群空间特征和相关专题信息与几何信息的保持,而对算法本身的效率关注相对较少. 近年来,随着LBS服务及网络地图服务的发展,对多尺度地图实时表达的需求不断增长,有的综合算法已经开始考虑效率问题. 在目前高性能计算技术快速发展的背景下,哪些算法适合用来做高性能计算,要经过严格的分析与实验,而其中算法的时间复杂度就是要考虑的一个重要方面. 本文调研了一些常见的点群选取与化简算法,并根据其算法原理进行分类,分别对其时间复杂度进行了研究,并对结果进行了分析与讨论.

1 常见的点群选取与化简算法及其分类

首先,针对点群要素选取与化简算法进行了调研,由于算法数目较多,对这些算法进行了分类,并从每一类中选取一种具有代表性的算法进行具体的分析.

1.1 常见的点群选取与化简算法

点群的选取与化简算法都是从原始的点集中选取一个子集,选取的点位置保持不变,而选取数量可以根据一定的模型(如开方根模型、回归模型等)来确定. 因此点群目标的选取与化简算法的重点是解决如何选取的问题.

对于如何进行点群要素的选取与化简,许多制图学者进行了大量的研究与实验. Langran 与 Poiker 提出了关于居民地选取的5种算法: 居民空间比率算法、重力模型算法、分布系数算法、分割算法和嵌套化简算法[3]. van Kreveld 针对 Langran 与 Poiker 算法无法对居民地进行重要性排序、选取过程不具有单调性的问题提出了圆增长算法,并且对圆增长算法给出了两个改进模型[4]. 毋河海在考虑点群具有聚合性质的基础上,运用凸壳工具实现了点群的结构化描述,并通过对具有嵌套结构凸壳的化简,实现了对点群目标的选取[5]. 艾廷华等运用 Delaunay 三角网来描述点群的分布特征[6]. 邓红艳等利用遗传算法进行了点群目标的选取[7]. 闫浩文等采用基本选取法则,通过 Voronoi 图来处理几何度量信息和拓扑信息,保证点状要素群的统计、专题、拓扑、几何信息得到正确传输[8].

除此之外,还有一些其他的算法. 本文整理的点群要素选取与化简算法如表1所示[3-12]:

表1 点群要素选取与化简算法

Table 1 Selection and simplification algorithms of point cluster

序号	算法名称	来源
1	居民地空间比率	[3]
2	分布系数	[3]
3	重力模型算法	[3]
4	圆增长算法	[4]
5	基于凸壳的化简方法	[5]
6	保持空间分布特征的群点化简算法	[6]
7	基于遗传算法的点群目标选取模型	[7]
8	基于 circle 特征变换的点群选取算法	[10]
9	基于 Kohonen 网络的点群综合方法	[11]
10	基于圆增长特征的点状要素选取算法	[12]
11	基于 Voronoi 图的点群综合算法	[9]

1.2 点群选取与化简算法的分类

通过对这些算法选取与化简过程实现原理进行分析,我们将这些算法分为以下几类: 基于点重要性的增量式选取、基于图(Voronoi 图、Delaunay 三角网等)的化简、基于智能算法的选取与化简与其他方式4类.

（1）基于点重要性的增量式选取.

该类选取模型通过一定的算法确定点群中各个点的重要性，然后根据点的重要程度依次确定要保留的点. 主要包括 Langran 与 Poiker 的居民地选取算法、圆增长算法、基于圆增长特征的点群选取算法等.

（2）基于图的化简.

通过构建 Voronoi 图、Delaunay 三角网或者最小生成树等实现对点群目标空间图形的表达，再基于图对点群进行化简，从而保持点群的空间特征. 主要包括文献[6]保持空间分布特征的群点化简算法、文献[9]基于 Voronoi 图的点群综合算法等.

（3）基于智能算法的选取与化简.

将智能算法引入到点群要素的综合中，来进行点群的选取与化简. 主要包括文献[7]基于遗传算法的点群目标选取模型、文献[11]基于 Kohonen 网络的点群综合方法等.

（4）其他方式.

还有一些算法的选取与化简原理与上述几类都不同，因此单独归为一类. 主要包括文献[5]基于凸壳的点群选取算法、文献[10]基于 Circle 特征变换的点群选取算法等.

2 点群选取与化简算法时间复杂度求解

时间复杂度是度量算法效率的一个重要指标. 算法的时间复杂度，即算法的时间消耗，它是关于算法输入规模 n 的函数，用 $f(n)$ 表示. 为方便计算，通常以渐进时间复杂度 $T(n) = O(f(n))$ 来进行时间复杂度的求解. 其中，大 O 符号表示函数的渐近上界，它表明了算法运行时间的最坏情况. 在分析算法的时间复杂度时，通常需要借助数学中的级数、数列求和及概率分析等工具. 对于上节中列出的点群要素选取与化简算法，本节将从每一类中各选取一种具有代表性的算法，来求解其时间复杂度.

2.1 基于点重要性的增量式选取

基于点的重要性的选取模型，是利用某种算法来确定点群中不同点的重要性程度，对较重要的点进行选取，而对不重要的点予以舍弃，点的选取是依次的、增量式进行的. Langran 与 Poiker 提出的针对居民地选取的 5 种算法中，分割算法与嵌套化简算法需要较多的人工干预，不适合用于地图的自动综合，因此本研究不作讨论，而其他的 3 种算法都是基于点的重要性进行增量式选取. 另外，圆增长算法、基于圆增长特征的点状要素选取算法也属于该类选取方式. 本文选取圆增长算法来进行分析.

圆增长算法的基本思想是：给每个点 1 个半径与其权值成正比的圆，即 $R_i = C \times I_i$（R_i 为第 i 个居民地的圆半径，I_i 为第 i 个居民地的权值，C 为常量且 $C > 0$）. C 的初始值要保证任何两个圆不会交叠，然后增大 C 值，使所有的圆都增大，直到其中 1 个圆被其他居民地的圆完全包含，此时将被包含的点列为居民地队列的最底层并且删除. 重复此过程直到只包含 1 个居民点，此时确定了所有居民地的选取队列.

圆增长算法时间复杂度的求解过程如下：

（1）权值 I_i 的计算.

居民地的权值通常为其人口数，或者综合人口数、工业产值、学校数量等数据，一般可以在线性时间内求得.

（2）初始 C 值的计算.

由于 C 的初始值是保证任何两个圆都不会交叠，因此只要令所有圆中距离最近的两个圆外切即可. 对任意两个圆（圆心分别为 P_i 点与 P_j 点）求 $k = \dfrac{R_i + R_j}{d(P_i, P_j)}$（$R_i$，$R_j$ 分别为两个圆的半径，$R_i = C \times I_i$，$R_j = C \times I_j$，$d(P_i, P_j)$ 为两圆圆心即两个居民地的距离）的值，并从求得的所有的 k 值中选取最大值，令其等于 1，即可求得 C 的初始值.

由于点群中有 n 个点，对所有的圆两两求值的次数是 $n(n-1)/2$ 次，再从这 $n(n-1)/2$ 个无序的数值中选取最大值，时间复杂度为 $O(n^2)$.

（3）点选取序列的确定.

C 的增长幅度是，每次增长使得一个圆正好被其他圆包含，即内切，此时大圆半径等于小圆半径与两圆圆心距离之和，即 $R_j = R_i + d(P_i, P_j)$（其中 R_i，R_j 分别为两个圆的半径，$R_i = C \times I_i$，$R_j = C \times I_j$，$d(P_i$，

P_j)为两圆圆心即两个居民地的距离). 因此,对任意两圆求其 $k = \frac{R_j}{R_i + d(P_i, P_j)}$ 的值,该值越大,说明大圆半径增长速度相对于小圆半径与圆心距之和的增长速度越大,大圆越容易首先将小圆包含起来.

对 n 个圆两两求 k 值,总共计算 $n(n-1)/2$ 次. 对这些数值由大到小进行排序,就可以得到最先被包含的圆的序列,其时间复杂度是:

$$T(n) = O\left(\frac{n(n-1)}{2}\log\frac{n(n-1)}{2}\right) \approx O(n^2\log n).$$

因此,圆增长算法的时间复杂度为 $O(n^2) + O(n^2\log n) = O(n^2\log n)$.

另外,通过加权 Voronoi 图来实现算法可以改进算法的时间复杂度,改进后的圆增长算法平均时间复杂度能达到 $O(n\log n)$,最坏情况下为 $O(n^2)$[4].

2.2 基于图的化简

Delaunay 三角网、Voronoi 图等工具可以很好地描述点群的空间分布特征,利用这些工具对点群进行综合,能够使点群的空间特征得到保持.

该类算法中文献[9]基于 Voronoi 图的点群目标综合方法同时考虑了统计信息、专题信息、拓扑信息、度量信息 4 类信息的描述参数,使得 4 类信息都能被正确传输,具有较强的代表性.

该算法首先构建点群的三角网并生成点群的边界多边形,并将一些新的虚拟点插入到该多边形中,形成 1 个新的点集,然后构建新点集的 Voronoi 图,以获取每个点的相对区域密度; 根据每个点的相对区域密度与重要性计算其选取可能性,并根据选取可能性与 Voronoi 邻域关系将准备删除的点标记出来; 最后如果剩余点的数量不符合根据 Radical Law 计算出来的结果,则将标记点删除并重复第二步; 反之则删除插入的虚拟点与标记点,并得到综合后的结果.

该算法的时间复杂度的求解过程如下:

(1) Delaunay 三角剖分与边界多边形构建.

进行 Delaunay 三角剖分的最快时间复杂度为 $O(n\log n)$[13]; 算法对位于外壳的三角形的边长进行判断,如果大于给定值则将边删除. 由于三角网中边的数目不超过 $3n-6$ 条[13],因此构建边界多边形的时间复杂度为 $O(n)$.

(2) 范围多边形的构建.

由于范围多边形的顶点数与边界多边形的顶点数相同,其顶点数必然不大于点群中点的个数,因此构建范围多边形步骤的时间复杂度为 $O(n)$.

(3) 借助 Voronoi 图对点进行迭代删除.

Voronoi 图的构建时间复杂度可以达到 $O(n\log n)$. 由于增加了不超过 n 个点,构建 Voronoi 图的时间复杂度仍为 $O(n\log n)$; 点选择概率的计算与排序,选择概率的计算次数为 n,而排序的时间复杂度为 $O(n\log n)$,因此步骤的时间复杂度为 $O(n\log n)$; 在点删除的过程,每一次迭代都需要重新构建 Voronoi 图,而迭代次数显然应该小于点个数 n,因此该步骤的时间复杂度小于 $O(n^2\log n)$.

因此,基于 Voronoi 图的点群目标综合方法的算法时间复杂度为

$$O(n\log n) + O(n) + O(n^2\log n) = O(n^2\log n).$$

2.3 基于人工智能算法的选取与化简

点群综合中点目标的选取,可以归结为要素整体选取的最优化问题. 由于制图综合自身的特点,这些优化目标通常难以用数学语言来表达. 而随着信息技术的发展,在计算机领域出现的智能算法为解决这些问题提供了新的思路. 遗传算法以自然选择和遗传理论为基础,将生物过程中适者生存的规则与群体内部染色体的随机信息交换机制结合,一些常用算法难以解决的优化组合问题可以取得较好的结果. 本文选取文献[7]基于遗传算法的点群目标选取模型来进行分析.

该算法首先采用自适应分类方法,将点群 M 依照密度分成若干类子群,然后再根据每个子点群的点数和最后要保留的总的点数,计算每个子点群中要保留的点数,最后结合凸壳化简方法和遗传算法对点进行选择.

其时间复杂度的求解步骤大致为:

（1）点群的自适应分类. 该步骤要根据点与点之间的相互作用力对点群进行分类，时间复杂度为 $O(n^2)$.

（2）凸壳的计算和化简.

分别对子点群求凸壳，由于所有点群总点数为 n，因此该步骤的时间复杂度小于 $O(n\log n)$；对凸壳的化简时间则取决其所采用的化简算法，以 Douglas-Peuker 算法为例，在最坏情况下其时间复杂度为 $O(n^2)$，平均情况下为 $O(n\log n)$. 由于外围凸壳点的数目一定小于 n，因此该操作的时间复杂度小于 $O(n^2)$.

（3）利用遗传算法进行点群目标的选取. 该步骤的时间复杂度为 $O(n^2)$.

因此，基于遗传算法的点群目标选取模型的算法时间复度为 $O(n^2)$.

2.4　其他方式

该类算法的实现原理与上述算法都不相同，不能归入上面的任何一类，因此单独作为一类. 基于凸壳的点群化简方法与基于 Circle 特征变换的点群选取算法即属于此类. 下面对基于凸壳的点群化简算法的时间复杂度进行分析.

对平面上的离散点集，逐层利用竖向划分条带的方法来求凸壳，得到一个逐层嵌套的多边形序列. 将原始凸壳层次进行压缩合并，形成几个不相交的多边形. 再针对每个多边形进行利用线综合的方法进行化简.

算法可以分解为以下步骤：

（1）构建嵌套凸壳.

利用竖直条带求近似解的方式来求凸壳，时间复杂度可以达到 $O(n+k)$，其中 k 为条带数目[14]. 该步骤的时间复杂度为 $O(n^2)$.

（2）凸壳的压缩合并. 时间复杂度为 $O(n)$.

（3）多边形的化简. 根据采取的线化简算法的不同，时间复杂度也不同. 若采用 Douglas-Peuker 算法，该步的时间复杂度为 $O(n^2)$.

因此基于凸壳的点群化简算法时间复杂度为 $O(n^2)$.

3　分析与讨论

上节中我们计算出了 4 种点群选取与化简算法的时间复杂度，利用同样的分析方式，可将其他算法的时间复杂度计算出来. 本文计算出的所有点群选取与化简算法的时间复杂度如表 2 所示：

表 2　点群要素选取与化简算法的时间复杂度表

Table 2　Time complexities of point cluster selection and simplification algorithms

序号	算法名称	时间复杂度
1	居民地空间比率算法	直接实现为 $O(n^2)$，利用 Voronoi 图可达 $O(\log n)$
2	重力模型算法	$O(n^2)$
3	分布系数算法	直接实现为 $O(n^2)$，利用 Delaunay 三角网可达 $O(\log n)$
4	圆增长算法	初始为 $O(n^2\log n)$；改进后平均 $O(\log n)$，最坏 $O(n^2)$
5	基于凸壳的化简方法	$O(n^2)$
6	保持空间分布特征的群点化简算法	$O(n^2\log n)$
7	基于遗传算法的点群目标选取模型	$O(n^2)$
8	基于 circle 特征变换的点群选取算法	$O(n\log n)$
9	基于 Kohonen 网络的点群综合方法	$O(n^2)$
10	基于圆增长特征的点状要素选取算法	$O(n^2\log n)$
11	基于 Voronoi 图的点群综合算法	$O(n^2\log n)$

由上表可以看出，点群的选取与化简算法普遍具有较高的时间复杂度. 这主要是因为，在进行点群的综合时，需要保持点群整体的空间分布特征，因此算法相对比较复杂，往往由多个步骤组成，而且某些步骤要进行反复的迭代或递归. 这其中，文献[3]提出的 3 种算法，进行一次选取的时间复杂度可以达到 $O(\log n)$ 或 $O(n^2)$；文献[4]的圆增长算法改进后可以达到 $O(\log n)$ 或 $O(n^2)$，而且算法可以给点群一个确定的排序，当改变选取点的数量时无须重新计算；基于凸壳和基于遗传的算法点群选取模型尽管时间复杂

度同为 $O(n^2)$,但是基于遗传算法的点群选取模型消耗时间要远远高于基于凸壳的化简[7];基于Voronoi的点群综合算法在综合后能够保持点群的多类信息,其时间复杂度也相对较高.从算法本身的特点来看,遗传算法由于每个个体相互独立并且可同时计算,具有很好的并行性.而其他大多数算法要么过于简单,要么算法步骤具有较高的时序性,因此不容易进行分解,如要进行并行可能需更多地从数据分解角度来考虑.

另外,由于本文是按照文献中提供的思想对算法进行分析,对于某些算法在实现上可能会与作者原来的实现方式有所不同,导致结果会有一定的出入.

4 结论与展望

本文针对点群要素的选取与化简算法进行了调研,并对这些算法进行分类,然后从各类中选取比较有代表性的算法对其时间复杂度进行具体分析.通过分析,得到了不同算法的时间复杂度,并基于算法时间复杂度与算法本身的特点,对其并行计算适宜性进行了初步探讨.

本文只是针对点群选取与化简算法进行了时间复杂度的分析,对于点群综合中的典型化、聚合与移位等算法的效率分析还有待于做进一步探讨.研究过程中发现,点群综合算法的设计与实现往往比较复杂,因此仅仅依据算法的时间复杂度未必能很好地代表算法的实际执行效率,而且点群综合算法的执行效率不仅与算法设计原理相关,还与算法的约束参数、计算环境、待处理的空间数据等因素相关,今后还需要在这些方面做进一步研究.

[参考文献]

[1] 王家耀. 普通地图制图综合原理[M]. 北京: 测绘出版社, 1992.

[2] Bereuter P, Weibel R. Generalisation of point data for mobile devices[C]//A problem-oriented approach: 13th Workshop of the ICA commission on Generalisation and Multiple Representation, Zurich, 2010.

[3] Langran C E, Poiker T K. Integration of name selection and name placement[C]//Proceedings of 2nd International Symposium on Spatial Data Handling. Seattle, Washington, USA, 1986.

[4] van Kreveld M, van Oostrum R, Snoeyink J. Efficient Settlement Selection for Interactive Display[C]//Proceeding of 12th Conference on Auto Carto. MD, USA, 1997.

[5] 毋河海. 凸壳原理在点群目标综合中的应用[J]. 测绘工程, 1997(1): 1-6.

[6] 艾廷华, 刘耀林. 保持空间分布特征的群点化简方法[J]. 测绘学报, 2002(2): 175-181.

[7] 邓红艳, 武芳, 钱海忠等. 基于遗传算法的点群目标选取模型[J]. 中国图象图形学报, 2003(8): 970-975.

[8] 闫浩文, 王家耀. 基于Voronoi图的点群目标普适综合算法[J]. 中国图象图形学报, 2005(5): 633-636.

[9] Yan H W, Weibel R. An algorithm for point cluster generalization based on the Voronoi diagram[J]. COMPUTERS & GEOSCIENCES, 2008, 34(8): 939-954.

[10] 钱海忠, 武芳, 谢鹏, 等. 基于CIRCLE特征变换的点群选取改进算法[J]. 测绘科学, 2006(5): 69-70.

[11] 蔡永香, 郭庆胜. 基于Kohonen网络的点群综合研究[J]. 武汉大学学报: 信息科学版, 2007, 32(007): 626-629.

[12] 高三营, 闫浩文, 陈静静, 等. 基于圆增长特征的点状要素群选取算法[J]. 测绘工程, 2008(6): 20-23.

[13] 周培德. 计算几何[M]. 北京: 清华大学出版社, 2005.

[14] Preparata F P, Shamos M I, 庄心谷, 等. 计算几何导论[M]. 北京: 科学出版社, 1990.

[责任编辑:丁 蓉]