Abstract

We prove a geometric version of the graph separator theorem for the unit disk intersection graph: for any set of n unit disks in the plane there exists a line such that intersects at most O((m + n)logn) disks and each of the halfplanes determined by contains at most2n/3 unit disks from the set, where m is the number of intersecting pairs of disks. We also show that an axis-parallel line intersecting O(√m + n) disks exists, but each halfplane may contain up to 4n/5 disks. We give an almost tight lower bound (up to sublogarithmic factors) for our approach, and also show that no line-separator of sublinear size in n exists when we look at disks of arbitrary radii, even when m = 0. Proofs are constructive and suggest simple algorithms that run in linear time. Experimental evaluation has also been conducted, which shows that for random instances our method outperforms the method by Fox and Pach (whose separator has size O(√m))

.**Keywords：**Disk,Graph,Halfplanes

1引言

平衡分隔符是一种基本工具，用于许多分治型算法，以及通过归纳法证明定理。给定一个无向图G = (V, E)与V的顶点集合和E作为其边缘集合和一个非负实数α∈(1/2,1],我们说一个子集S⊆V是一个α-separator如果G的顶点集\ S可以分割成两组a和B,每个大小最多αV | |这样没有边缘a和B之间的参数α决定如何平衡两组a和B的大小。均衡分离器有用我们希望的大小S | |分隔符和α≥1/2小。

2 论文描述问题

在一般稀疏图中，人们做了大量的工作来证明具有某些性质的分离子的存在。为

例如，著名的Lipton-Tarjan平面分隔定理指出，对于任何n个顶点的平面图，都存在大小为O(√n)的2/3分隔符。类似的定理已经证明了有界亏格图，无极小图，低密度图和多项式展开图。这些分隔符结果适用于不包含任意大小的完整图的图类，并且每个图都包含类只包含O(n)条边，其中n是顶点的数量。因为任何α-separator的完全图(n)顶点，对于包含完全图的图类的分隔符的研究似乎是无用的。然而，目前还不清楚对于可能稠密的图，分隔符相对于边的数量有多小。我们感兴趣的焦点可能是稠密的几何图形，它常常编码其他的几何信息邻接。即使可以在几何图形中使用分隔符工具，通常也会丢失几何信息在这个过程中。因此，部分文献集中在寻找平衡的分隔符，也保存了几何图的几何性质。这种分色器称为几何分色器。

其中，我们重点介绍了Miller等人的工作，以及Smith和Wormald的工作。They Rd intersection 图 n 的 球 , 证明 了 如果 采用 空间 中 每 一 个 点 是 最多 被 k 的。那么存在一个O(k1/dn1 - 1/d)大小的(d + 1)/(d + 2)-分离器(在确定性中可以找到这样的分离器)线性时间)。更有趣的是，分隔符本身和它创建的两个集合有很好的属性;他们表明,存在一个(d - 1)维球，它与至多O(k1/dn1 - 1/d)个球相交，最多包含(d + 1)n/(d + 2)个球内部有球，外部最多(d + 1)n/(d + 2)个球。在这种情况下，球体充当分隔符(适当地)，而集合A和集合B是球的内外分别分离器球体。注意，由集合A所导出的图是由其中的球的交点图组成的隔符(类似地，分隔符外的球用B表示，与球体相交的球用S表示)。即使对于平面图来说，分隔符的大小大于Lipton-Tarjan中的分隔符(特别是对于高值的d)，它的主要优点是它创建的三个子图是相同的几何图

族(Rd中球的交点图)。当k较大时，即使对于d = 2: 如果 √n 磁盘 重叠 在 一 个 单 点 , 另 一 个 磁盘 形式 路径 k = √n 和 m = (n), m 是 哪里交集图的边数。因此，分隔符的大小为O(√)kn) = O (m3/4)。Fox和Pach给出了另一个遵循相同精神的分隔符结果:一组Jordan的交集图如果每一对曲线相交于一定数量的点，则平面上的曲线有一个大小为O(√m)的2/3分隔符。R2中的一组磁盘满足这个条件，因此这个定理适用于磁盘图。他们的证明可以变成一个多项式时间算法。然而,我们需要构建磁盘的安排,需要O (n22α(n)),在那里α(n)是阿克曼的逆函数,并在实践中一个有效的实现是不平凡的。从几何角度看，这两个结果表明，给定平面上的一组单位圆盘，我们总能找到平面上的闭合曲线(分别为圆[24,27]和约当曲线[9])对集合进行划分

曲线是分隔符中的曲线结果与论文组织。In 本文 我们 继续 的 想法 几何 分隔 符 和 显示 平衡 separator 总是 exists, 即使 我们 限制 分离器 一 行 (see Fig. 1 ). Given 一 组 n 单元 与 m 双 磁盘相交 disks, 我们 显示 一 行 2/3-separator 大小 O((m + n)logn) 可以 在 预期 O(n) time, 和在确定的O(n)时间内，可以找到大小为O(√m + n)的轴向平行线4/5分隔符。

3如何解决问题

将结果与之前的工作进行比较，算法在四个方面匹配或改进。简单的形状:圆[24,27]与约旦曲线[9]与我们的线，(ii)集A和B的平衡:3/4[24,27]与/3[9]和我们的大小分离器:O (m3/4)与O(√)[9]与我们O˜(√)。最后，(iv)我们的算法简单和渐近速度:O (n)和O˜(n2)与我们的O (n)。请注意，这些算法需要几何表示。

例如，单位磁盘图是由一组单位磁盘给出的，而不是组合成图。事实上，找到一个单位磁盘图形的几何表示并不是一项简单的任务:这个问题是NP-hard和∃R-complete ，两个不相交的集合分别是曲线内外的圆盘。

我们强调，我们的结果侧重于单元磁盘图，而其他结果适用于任意磁盘图半径。实际上，如果我们想要用一条线来分隔具有任意半径的磁盘，我们可以显示分隔符的大小

一样 大我们还证明了对于单元磁盘，我们的算法可能无法找到更好的2/3行分隔符

比 O(m log(n/√m)) case; 最差确切的陈述可以在第3节中找到。在这个意义上，我们的大小分离器是渐进的，几乎是紧的。第四部分给出了实验结果。我们评估了

将我们的算法与Fox和Pach[9]的方法进行比较，得到的随机分色器的大小实例，并得出结论，我们的算法在单位磁盘的交集图上的性能优于他们的算法。使用线分隔符处理交叉磁盘有些困难。If 我们 选择 独立 的 两 两 不 相交 的 geometric 对象 由 Jordan curve, 然后 我们 可以 雇佣 curve. 的 内部 体积 参数然而,我们不能为行分隔符使用卷参数，因为该行不确定有界区域。其他相关工作。在不同的背景下，也研究了双不相交单元圆盘的线分离器。自圆盘是两两不相交的，交点图一般是空的，可以很容易地分离出来。相反，现在的重点是找到一条与少量磁盘相交的闭合曲线，使它定义的两个连接的组件包含大致相同的内容数量的磁盘。Alon等人证明了对于给定的n对不相交的单位圆盘集合D，存在这样一个斜率a，使得每一个线 与 边坡 相交 O(n logn) 单位 磁盘 D. In particular, 减半 的 直线 的 斜率 将 是 一 个 不错 的 分离器(each halfplane 将 最多 n/2 − O(n 完全 包含 in). logn) 磁盘他们的证明是概率性的转化为期望O(n)时间随机化算法。然后给出了一个确定的O(n)时间算法

由 Hoffmann 等 人 , 还 展示 了 如何 找到 一 个 相交 的 线 最多 O(√n/(1 − 2 α)) 磁盘 和 单位每个halfplane包含最多(1−α)n磁盘(0 <α< 1/2)。Loffler和Mulzer[21]证明了存在一个axis-parallel 线 这样 相交 O(√n) disks, 和 每个 halfplane 包含 最多 9 n/10 单位 disks, 可以在O(n)时间内找到。我们的结果更一般(因为它允许交叉)，并且有更好的平衡参数(他们的9/10对我们的4/5)。为了便于比较，这三个结果也显示在表1中。

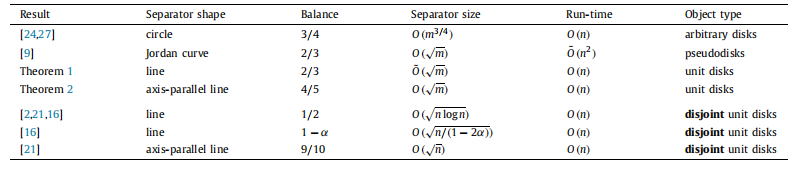


图3.1 表1

我们通过分离器的尺寸来评估分离器算法的质量。定理1提出了一个简单的方法

算法:找到一个中心点(这可以在线性时间)，并尝试随机线通过该点，直到找到了一个好的分离器。由于实现中心点算法并不简单，所以我们使用了另一种方法是渐进较慢，但更容易实现:为斜率a均匀随机选择找到一个2/3分隔符与最小磁盘数相交的斜率a(通过对磁盘排序，此步骤可以在O(n log n)时间内完成)在a的正交方向上作平面扫描)。可见，改进算法找到的一个分隔符相交于通过中心点的直线与通过中心点的直线一样多。因此，在定理1中，是一个随机方向将是好的，有正的概率。我们重复这个过程多次，以获得良好的行分隔符。我们将我们的算法与Fox和Pach[9]的方法进行了比较，该方法保证了O(√m)的分隔符大小。为我们的算法的实现我们使用上面描述的更简单的变体。Fox和Pach证明了平面上的一组约当曲线的交图具有大小为2/3的分隔符O(√m)如果每一对曲线相交的点个数相同。他们的证明是建设性的，如下所述。图3.2所示。q = 11的snake实例。在这种情况下，n = (q2−1)/2 + q = 71。因此，实例包含71个单元磁盘。First, 我们 构建 的 安排 curves, 并 获得 一 组 平面 图形 的 顶点 的 顶点 arrangement 和 连续 顶点 由 edge. 曲线 连接我们对所得到的平面图形进行三角剖分使之成为平面图形最大的平面。然后，我们找到一个简单的循环2/3分隔符C(即在图中形成一个循环的2/3分隔符)的大小O(√m + n)总是存在于[23]。我们输出C中包含一个顶点的所有曲线。在我们的实现中，我们用蛮力的方式构造了圆的排列，并使用了一个简单的循环分离器。

算法由Holzer等人实现，称为基本循环分离器(FCS)算法。虽然FCS算法没有

Fox-Epstein等人最近的实验研究表明，理论保证了所得分离器的尺寸它具有与最先进的循环分离器算法相当的性能，并在理论上保证了大多数的性能的病例。我们在实验中使用了两组实例。我们称第一个集合为随机实例集，称第二个集合为随机实例集一组snake实例。随机实例是随机产生的，就像通常在传感器上做的实验工作一样网络。我们固定一个边长为L的平方S，并在S中独立且均匀地随机生成n个单位磁盘。如果图是不连接的，我们丢弃它并再次生成。直观地说，对于任何它将n个磁盘放在sizelength√2n的平方中。圆盘的放置方式是它们的交集图是路径，因此是稀疏的。这个实例的一个特殊特性是它有一个固定大小的分隔线:一条垂直线可以做一个平衡的切割，只与一个圆盘相交。然而,当我们随机选择一条线时，很难找到分隔符。随着实例的大小，概率趋于0生长。snake实例的目的是观察算法在不利条件下的行为。snake实例的具体构造如下:为了便于生成，我们将单元磁盘的半径设置为4/3，固定一个奇数q，设n = (q2−1)/2 +q。我们将n个单位磁盘中心放在整数坐标(x, y)∈{2i−1 | 1≤i≤(q + 1)/2}×{j | 1≤j≤q}∪{(2i, qi mod 2) | 1≤i≤(q−1)/2}处。注意，当i是偶数时，qi mod 2等于1当i是奇数时等于q。因此，对于每个q, snake实例都是唯一确定的。通过结构观察边数m为n−1 = (q2−1)/2 + q−1。

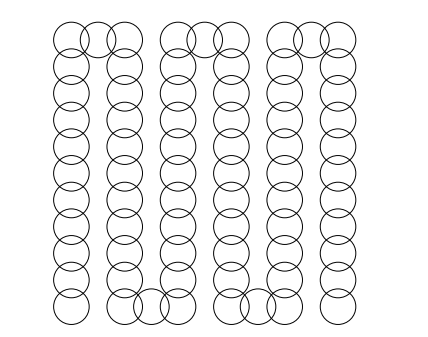


图3.2 The snake instance with q = 11

All 实验 已经 进行 Intel CoreTM i7 - 5600 U CPU @2.60 GHz × 4 和 7.7 GB 内存976.0 GB硬盘，运行Ubuntu 14.04.3 LTS 64位。在第一个实验中，我们实证地检验了我们提出的算法所得到的分隔符的大小本节开头提出的修改。对于随机实例，我们固定L = 100，并将磁盘数量n从10,000更改为30,000，增量为50. 由于我们的算法是随机的，所以我们运行算法k次，其中k∈{1,2,10,15,20}，然后计算平均值分离器 尺寸 (because Theorem 1 , 我们 预计 平均 收敛 O(m logn) infinity).

4 小结

这篇论文留下了一些有待解决的问题。定理1的证明依赖于中心点的存在和平衡

2/3看起来是固有的。在一个行分隔符的平衡上给出一个更好的界是一个开放的问题，

可能会以分隔符大小的更糟糕边界为代价。这可能会导致平衡和平衡之间的权衡

线分隔符的分隔符大小。

本文主要研究磁盘，特别是单位磁盘。在2.2节末尾解释的方法可以是用于表示与定理1相似大小的胖凸物体的相似结果。然而，我们不知道是否该方法可推广到任意大小的凸目标。这是另一个有待解决的问题。