选题报告: 最大空元凸子集

张亚娴 软件学院 2016213643　　刘聪颖 软件学院 2016213637
宋正阳 交叉信息院 2016211701

2017 年 4 月 23 日

1 简介

在本节我们对要解决的问题进行定义，并说明其实际意义。

计算几何中的一类重要问题便是寻找给定点集中具有某种性质的子集，而空元凸子集关注的是凸性和空元性。具体来说，给定点集 S 的空元凸子集满足以下性质：该集合中的点全部落在其整体构成的凸包上，即凸包内部为空。而这里的“最大”考虑的是顶点个数，而非凸包面积。我们希望找到满足上述性质的子集中所包含的顶点个数最多的那个。

2 相关工作

在本节我们介绍相关的系列研究，包括提出的算法及对应的时空复杂度。

Chavatal 等 [CK80] 给出了关于最大凸子集（即不考虑空元性质）时间复杂度为 $O(n^3)$ 的做法。而 Avis 等 [AR85] 在此基础上进行了简单的修改给出了最大空元凸子集的时间复杂度为 $O(n^3)$，空间复杂度为 $O(n^2)$ 的解法。通过引入几何排列（Arrangement）的思想，Herbert 等人 [EG89] 把空间复杂度由 $O(n^2)$ 降至 $O(n)$。而当前已知的最优做法出自 David 等 [DEO90]，其时间复杂度正比于集合的空三角形的个数，该值在点集在单位正方形内随机取点时期望值为 $O(n^2)$；空间复杂度正比于有着相同最左顶点的空三角形的个数最大值。

3 算法描述

在本节我们对上文中提到的三种算法做概述。

3.1 时间 $O(n^3)$, 空间 $O(n^2)$

该算法 [AR85] 由以下几个部分构成：

1. 将所有可能的凸子集按其最左顶点分类，对于 S 中的每个顶点 p，将其右侧的顶点安装极角排序，并构造以其为核的扇形。
2. 计算该扇形的可见图，作为结果的候选的空凸子集的凸包只可能由该可见图中的边构成。
3. 计算可见图中以 p 为做顶点，可见图中任意顶点为右顶点的最长凸链和最长凹链，具有相同左右顶点的凸链和凹链拼合为一个空凸包。
4. 遍历右顶点求出凸链和凹链的长度和最大值，并由该最大值找出对应的空凸集。

3.2 时间 $O(n^3)$, 空间 $O(n)$

该算法 [EG89] 使用对偶改进了空间复杂度。在计算以 p 为最左顶点的凸子集时我们可以如下操作：

1. 顶点 p 被映射为 y 坐标为无穷远的点，如果子集 U 在对偶空间中被映射为一段凹链，则 $U \cup p$ 构成原图中以 p 为最左顶点的凸子集。于是计算凸子集转化为计算对偶空间中的凹链。
2. 而计算凹链时，凹链中的点相当于对偶空间中的单调凹路径中边的支撑线。于是计算最长凹链转化为计算对偶空间中的最长单调凹路径。
3. 这里单调路径是指任意垂直线与该路径只有至多一个交点，凹是指从左往右遍历时每一个拐点都是左拐。该问题可在时间复杂度 $O(n^3)$, 空间复杂度 $O(n)$ 内解决。
4. 寻找原问题的最大凸子集时使用暴力的回溯算法，时间复杂度为 $O(n^3)$.

3.3 时间 $O(\gamma_3(S))$

该算法 [DEO90] 计算给定点集中的所有 r 凸多边形时大致分为三个部分：

1. 对于 S 的每个顶点 p，舍弃掉其左边的点，将剩余点按照 p 极角排序，从而得到以 p 为核的星形多边形 P_p。使用几何排列中的对偶 [CGL83, EOS86] 该部分时间复杂度可由 $O(n^2 \log n)$ 降至 $O(n^2)$。
2. 计算 P_p 的可见图 VG_p，该可见图包含 P_p 原图中的边，但不包含以 p 为顶点的边。该部分存在时间复杂度为可见图输出大小的线性的算法 [Her87]。

3. 计算 VG_p 中所有长度为 $r-2$ 的凸链，从而对于任意一个凸链将其首尾与 p 相连便可得到一个 r 凸多边形。该部分时间复杂度也为输入可见图的大小的线性。

在枚举所有的 r 空元凸多边形的过程中我们顺便得到了其中最大值。注意到该算法的前两步与上文中的 [AR85] 是相同的，但采用了更好的算法降低了时间复杂度。

4 进度安排

在本节我们描述在课程作业中目标完成的部分及小组成员分工。

我们决定实现 [DEO90] 中描述的算法及其演示，大致做如下解耦：

1. 极角排序
   - 输入：平面上的 n 个点
   - 输出：其中每个点的右侧点以该点为极点的极角排序
2. 可见图生成
   - 输入：一个以最左侧点为核的扇形（星形多边形）
   - 输出：该扇形的可见图
3. 凸链生成
   - 输入：一个扇形的可见图
   - 输出：该可见图中除去最左侧核外长度为 $r-2$ 的凸链
4. QT 做演示界面
   - 数据点的读取（或随机生成）
   - 最终结果的显示
5. 算法执行过程中的效果演示

我们初步的分工如下：1→ 张亚娴，2→ 刘聪颖，3→ 宋正阳，4→ 张亚娴，5→ 各自负责各自部分。实现语言考虑与 QT 兼容故使用 $C++$，测试由实现人员自行进行，版本控制使用 GitHub，底层的数据结构包括 DCEL 等，分工在具体执行过程中视各部分任务量及进度会进行调整。

参考文献

[AR85] David Avis and David Rappaport. Computing the largest empty convex subset of a set of points. In *Proceedings of the first annual symposium on Computational geometry*, pages 161–167. ACM, 1985.

[CGL83] Bernard Chazelle, Leo J Guibas, and Der-Tsai Lee. The power of geometric duality. In *Foundations of Computer Science, 1983., 24th Annual Symposium on*, pages 217–225. IEEE, 1983.

[CK80] V Chvatal and G Klincsek. Finding largest convex subsets. In *Proc. 11th SE Conf. on Combin., Graph Theory and Comp*, 1980.

[DEO90] David P Dobkin, Herbert Edelsbrunner, and Mark H Overmars. Searching for empty convex polygons. *Algorithmica*, 5(1-4):561–571, 1990.

[EG89] Herbert Edelsbrunner and Leonidas J Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences*, 38(1):165–194, 1989.

[EOS86] Herbert Edelsbrunner, Joseph O’Rourke, and Raimund Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM Journal on Computing*, 15(2):341–363, 1986.

[Her87] John Hershberger. Finding the visibility graph of a simple polygon in time proportional to its size. In *Proceedings of the third annual symposium on Computational geometry*, pages 11–20. ACM, 1987.