KILOWHAT 题解

邹雨恒

1 题目大意

平面上有两个点集共N个点，你要找到一个多边形，将这两个点集分开，即

- 第一个点集所有点在多边形外或多边形上，第二个点集所有点在多边形内或多边形上
- 第一个点集所有点在多边形内或多边形上，第二个点集所有点在多边形外或多边形上

要求尽可能使多边形的周长更小。

数据有两类，生成方式如下：

- N在100到200之间随机（包含100和200）。实数Q在0.2到0.8之内随机（包含0.2和0.8）。然后随机N个点，横纵坐标分别在1到1000之间随机（包含1和1000）。每个点是黑点的概率为Q，为白点的概率为$1-Q$。
- 随机在10到14之内生成两个整数W,H（包含10和14）。随机在20到70之内生成两个整数d_W和d_H（包含20和70）。实数Q在0.2到0.8之内随机（包含0.2和0.8）。N赋值为$W\times H$，N个点分别是$(d_W\times i, d_H\times j)$，$i$从1到$W$枚举（包含1和$W$），$j$从1到$H$枚举（包含1和$H$）。每个点是黑点的概率为$Q$，为白点的概率为$1-Q$。

2 算法讨论

提示中已经告诉我们可以让多边形经过所有的点，这样的解必然是合法的。

我们可以把所有点排水平序，然后从左到右从下到上连接起来，然后最后找到一个从最后那个点到最开始那个点的一条与所有边不相交的折线（比如先向上连到一个比y坐标最大的点还大的位置，再向左连到x坐标最小点的位置，再连回来），这样是一个合法的解。

同样可以把最左边的点拿出来，按照这个点把剩余点极角排序，然后按照极角序顺次连接起来，这样得到的解要比上面那种方法更优一些。

针对上面这个做法还可以通过随机的方式得到更好的解。上述做法中极角排序的基准点是最左边的点，我们可以随机地找一点。具体方法是给第i个点随机一个权值w_i，然后定义

$$S_x = \sum_{i=1}^{n} w_i x_i$$

$$S_y = \sum_{i=1}^{n} w_i y_i$$

$$S_w = \sum_{i=1}^{n} w_i$$

然后用$(S_x/S_w, S_y/S_w)$作为基准点对剩下的点极角排序，然后顺次连接每个点。这样还可以得到更好的解。

事实上，这个问题类似于旅行商问题，可以尝试旅行商问题的随机方法。比如每次考虑随机出一个点的排列，假设这个点的序列就是多边形的点的序列。这样一来我们有两个问题：要尽可能缩小周长，并使得这个多边形不自交。每次找到两条边(a, b)和(c, d)，用$dis_(p, q)$表示p, q两点间距离。如果有$dis_{a,b} + dis_{c,d} > dis_{a,c} + dis_{b,d}$，那么就把这两条边变成$(a, c)$和$(b, d)$，直到没有这样的两条边为止。这样算法结束之后，没有两条边满足$dis_{a,b} + dis_{c,d} > dis_{a,c} + dis_{b,d}$，所以这个算法终止的时候多边形必然不自交。这个算法可以得到比较高的分数。

以上做法虽然已经得分比较高了，但是还没有用到题目中点的颜色的条件。下面考虑利用这个条件。

定义一个过程，这个过程接收一条多边形的边、一个内部点集合和一个外部点集合，返回一个包含第一个参数那条边的多边形，使得所有内部点在多边形内部，外部点在多边形外部。这里要求对这条边做两条穿过端点的垂线，所有内部点和外部点都严格在这两条垂线中间的一带。首先求出内部点和多边形那条边的两端点的凸包，然后对于每个在凸包内部的点求出一个最近的边，把这个点加入对应边的集合（内部点和外部点要分开存），然后再递归处理（递归处理的时候实际上原来的内部点变为外部点，外部点变为内部点），再把多边形的轮廓合并即可。

上述算法也可以通过加入随机的方式获得更优的解，但是效果不是特别好。

第二部分数据是一个网格图，可以通过加入一些假设使用DP处理这部分数据，可能会得到更好的解。

3 时空复杂度

时间复杂度：类似于旅行商问题的解法$O(kN^3)$，其中k为调整次数。凸包算法复杂度不好估计。

空间复杂度：$O(N^2)$。