随机增量算法

解轶伦

【摘要】

随机增量算法是计算几何中一个重要的算法，它对理论知识要求不高，编程复杂度低，而应用范围却十分广大。本文通过两个经典例题，详细阐述了本人一步一步理解随机增量算法的过程，最后是本人对随机增量算法的总结以及思考过程中的感悟。

【关键字】

增量算法　　随机化　　计算几何

【目录】

【引言】

增量法（*Incremental Algorithm*）的思想与第一数学归纳法类似，它的本质是将一个问题化为规模刚好小一层的子问题。解决子问题后加入当前的对象。写成递归式是：

$$T(n)=T(n-1)+g(n)$$

增量法形式简洁，可以应用于许多几何题目中。

增量法往往结合随机化，以避免最坏情况的出现。

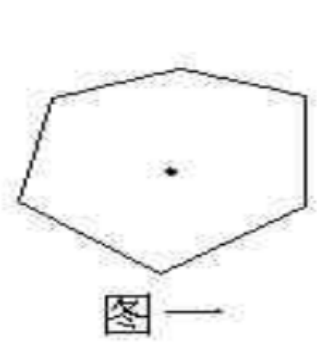
【正文】

例一 监视摄像机 （CTSC 1998 第一试）

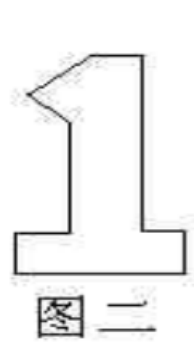
问题描述

一个著名的仓库管理公司＊ERKOI 请你的公司为其安装一套闭路监视系统。由于SERKOI 财力有限，每个房间只能安装一台摄像机作监视用，不过它的镜头可以向任意方向旋转。

首要的问题是确定摄像机的位置以确保房间的每一个角落都能被它监视到。例如，图一和图二是某两个房间的示意图，每个房间用一个封闭的多边形表示，图中的每条边表示一面墙。对于图一所示的房间，我们将摄像机安置在标黑点的位置就能满足要求；而对于图二所示的房间，无论将摄像机安置在那里都无法使其满足要求。



图一



图二

写一个程序，对于给定的房间示意图，判断是否有可能在这个房间中的某一位置安置一台摄像机，使其能监视到这个房间的任何一个角落。

输入

输入文件包含一个或多个房间示意图的描述信息。每个描述信息的第一行是一一个正整数 n（4＜=n＜=100），表示该房间的示意图为一个 n 边形。以下 n 行每行包括用空格符隔开的两个整数 x,y,按顺时针方向依次为这个 n 边形的 n 个顶点在直角坐标系中的的横纵坐标，x,y,的范围在：-1000 至 1000 之间。若 n 等于 0 则表示输入文件结束。

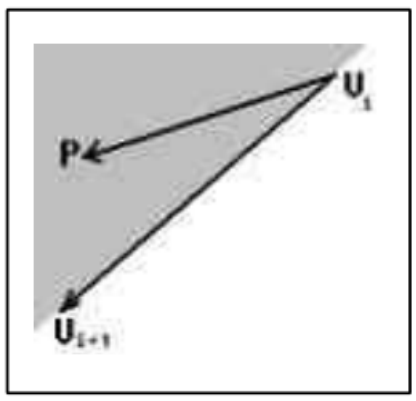
输出

对于每个房间，首先输出一行该房间的编号信息“Room ＃k：”，k 按照输入次序从 1 开始计数。紧接着一行是判断结果，如果摄像机在房间中某处安置能满足条件，输出：“surveillance is possible。”，否则输出“surveillance is impossible。” 每两个房间的输出结果之间用一个空行隔开。

申明：下面的分析与算法描述中，半平面、直线、不等式、向量都可视为等价的。

问题分析

读完题目给人的第一印象是求半平面相交，由于题目按顺时针方向给出顶点，所以我们可以用向量来表示半平面：



如图，如果点 P 在边 ViVi+1 内侧，那么从 ViVi+1 转到 ViP 为顺时针，所以它们的叉积小于等于 0，于是我们可以很容易地得到每个半平面的代数形式。

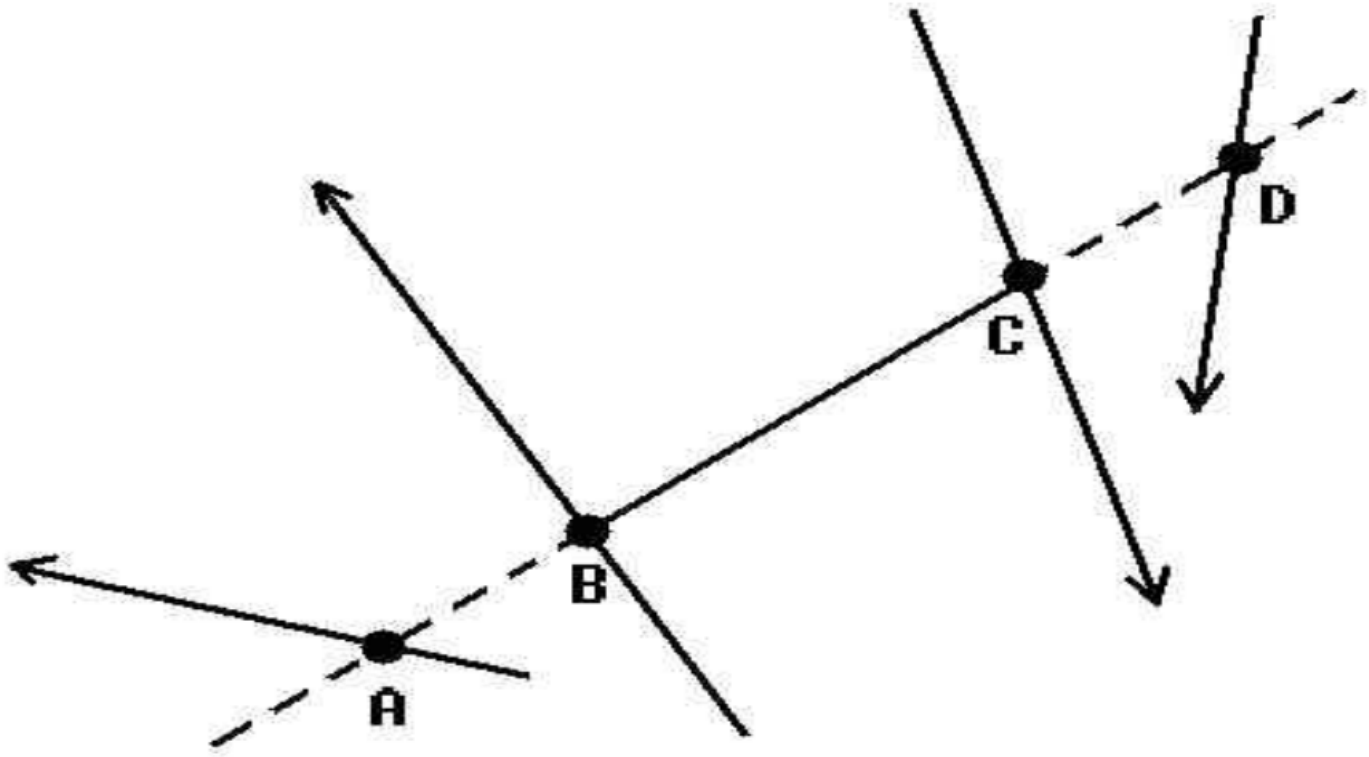
但如果直接做半平面这令人感觉杀鸡用牛刀，因为问题只要求判断是否存在可行区域，并不要求具体解的情况，因此我们应该考虑更简单的方法。

另一个直观的想法是，枚举网格上的点，看能否满足不等式组，但这显然不会是一个有效算法。上面的算法看上去“很笨”，但是会让我们产生一种改进的冲动。枚举网格上的点太盲目，完全没有利用已知条件。再来仔细分析一下问题，半平面相交最后得到的一定是个凸多边形，而我们就是要找到一个点，这个点属于这个凸多边形，那么我们可不可以加强一下命题，只考虑一些特殊的点呢？显然，凸多边形的顶点就是一类很特殊的点，它们拥有内部点所有的性质，不同之处在于，它们必是两条线的交点。那么我们可以求出所有交点，再逐个判断。

如果就此打住，我们可以得到一个$O(n^3)$的算法，足以将 1998 年国家队选拔的这道题解决了，但你看到这篇文章时，起码 2009 年已经过了一半，倘若此题重现江湖，你难道指望它还像上个世纪时那么弱小吗？新世纪，强算法，我们还得继续思考。

我们求出了所有的交点，之后逐一判断，这是不是很浪费呢？
如下图，显然 A 和 D 已经被排除了，只有 BC 之间的区域是有效的，也就是说交

点中只有 B 和 C 是有效的。于是我们很自然地想到，每条直线，最终都只有 2 个交点是有效的，那么我们最终只需要判断 2n 个点是否满足不等式组。



到这里我们已经得到了一个$O(n^2)$的算法了，似乎已经很优了，但实际上仍有更优的算法。我们一直是处理完所有直线后再判断，如果我们每处理完一条直线就判断呢？假设前 k 条直线的交点中有一个点 P 满足前 k 个不等式，那么考虑第 k+1 条线时，先看点 P 在不在第 k+1 个半平面上。如果在，那么 P 就是满足前 k+1 个不等式的点，如果不在，那么无非两种情况。

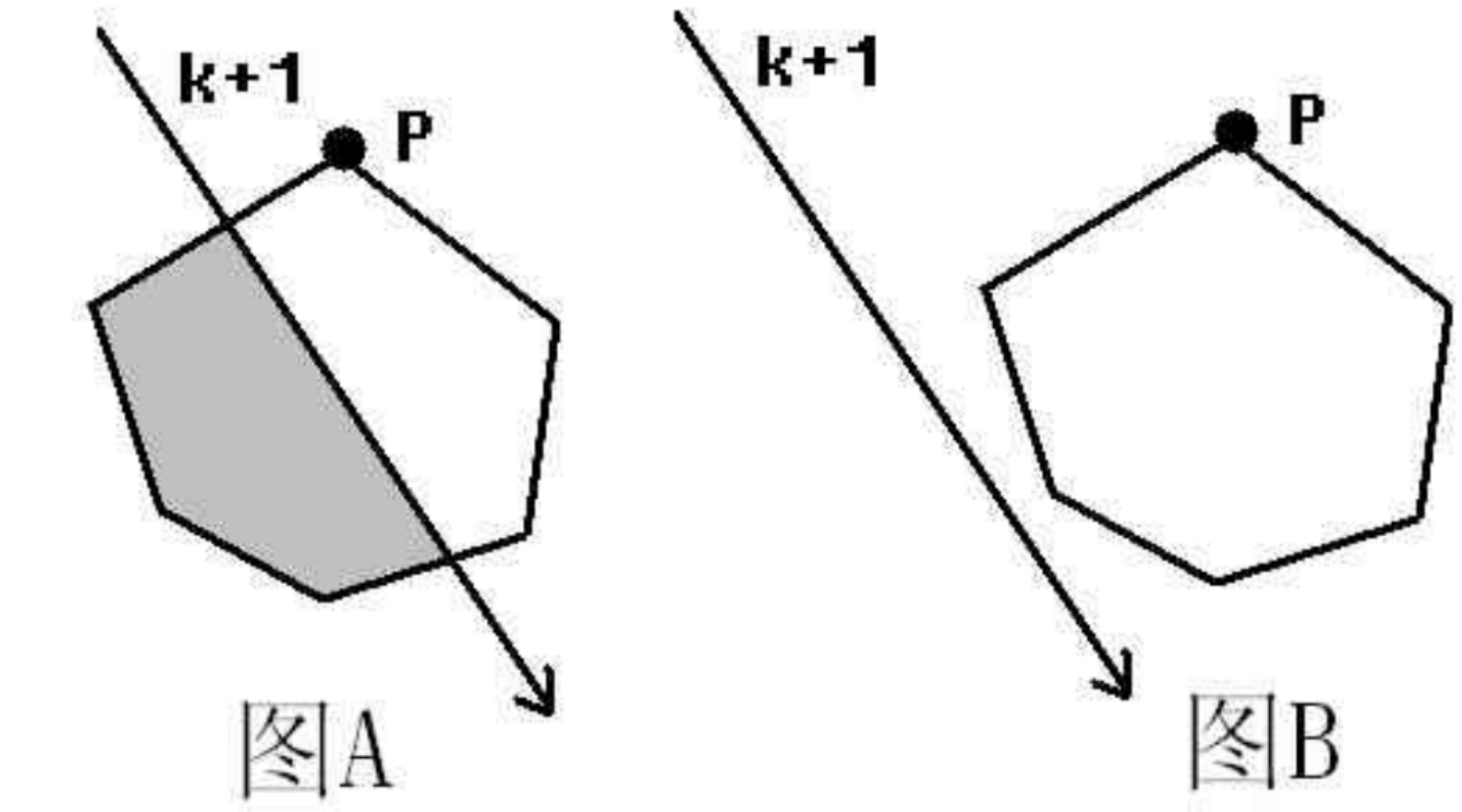


图 A 中，第 k+1 条线与前 k 条线交点中，必有满足前 k+1 个不等式的点。
图 B 显然是无解的情况，此时第 k+1 条线必被前 k 条线截空。

如果按照这个思路写出程序，似乎比刚才的更优了，因为这已经不是严格的

$O(n^2)$的算法了，但是这个算法的最坏情况仍是n^2的，在理论上并不必上个想法优。可是凭我们的直观感觉，达到最坏情况的可能性微乎其微，在平均情况下这个算法应该是很优的。这段话是不是很熟悉？在哪里听过？快排！我们只要像快排一样加入一句随机化，就可以得到一个期望效率$O(n)$的算法了。

算法描述

① 随意找一点作为初始解，并用随洗牌法打乱半平面顺序。

② 如果没有型的半平面可以加入，输出有解。否则转③。

③ 加入一个新的半平面，如果当前点在半平面上，转②，否则转④。

④ 对于新加入的第 k+1 条直线，用前个半平面去截，如果截后有剩余，保留剩余部分最下方一点，转②，否则输出无解。

复杂度分析

用 p（i）表示第 i 个半平面加上去时最优点发生变化的概率，则总的时间复杂度的期望为$\sum_{1}^{n} O(i) * p(i)$

我们可以想到，一个最下方的点是由两个半平面相交得到的，而在加上第 i 个半平面后，最下方的点发生变化，那么第 i 个半平面必是确定最下方点的两个半平面之一，由于前 i 个个半平面的顺序是随机的，所以第 i 个半平面是确定该点的两个半平面之一的概率是$\frac{2}{i}$(这就是选最下方点的原因)。因此总的时间复杂度的期望为$\sum_{i=1}^{n} O(i) * \frac{2}{i} = O(n)$

虽然只是在最后加入了随机化，但是却把时间复杂度降低了一个数量级，并且几乎没有增加编程复杂度，这可以说是最合算的优化了。

例二 最小外接圆问题

问题描述

给定平面上一组点，求一个最小的圆，包含所有这些点。

算法一 张角法

首先，易知至少有两个点在圆上（除非总共只有一个点）。我们首先枚举这两个点（设为 A, B）。

初中的平面几何知识告诉我们对于同一段弧，圆周角小于圆内角（如$\alpha < \gamma$），设 $\overline{AB}$ 上方最小的张角为α，下方最小的张角为β，则当$\alpha + \beta \geq \pi$ 时，$\triangle ABC$ 或 $\triangle ABD$ 的外接圆可以覆盖所有的点。特别地，当$\alpha + \beta = \pi$ 时 ABCD 四点共圆。

可以看到，该算法的时间复杂度为$O(n^3)$，并且是一个确定性算法。

算法二 随机增量法

在算法一基础上进行改进，先任取两点，求出覆盖它们的最小圆，然后每次增加一个点，如果该点仍在圆内，考察下一个点，否则扩张当前圆。

这里要用到一个定理：新加入的点不在当前圆内，那么必在扩张后的圆上。（证明见附录。）这样我们就可以再枚举另外一个点，再使用张角法。

若每次新加入点都在当前圆内，那么算法为$O(n)$。那么新加入点不在圆内的概率为多大呢？就像例一“两线定一点”一样，这里“三点定一圆”，所以概率为$\frac{3}{i}$。

因此期望的时间复杂度为$\sum_{i=2}^{n} O(n^2) * \frac{3}{i} = O(n^2)$

算法三

很明显算法二对增量法的应用并不彻底，实际上我们在求新圆时可以再用一次增量算法：假设在加入 pi 时重新计算，那么先以覆盖 pi 和 p1 的最小圆为开始，将 p2 到 pi-1 依次加入，如果加入的点 pk 不在圆内，说明该点必在新圆上，再从 p1 到 pk-1 枚举第三个点以确定新圆。

不难算出，当加入 pi 后，重新求出新圆的时间复杂度为 $O(i)$，于是总的时间复杂度为 $\sum_{i=2}^{n} O(i) * \frac{3}{i} = O(n)$

【总结】

增量算法蕴含一种步步为营的思想，希望改进当前解以逐步解决问题。计算几何往往是巨量代码的代名词，但是我们可以看到随机增量算法实现非常简单，如果比赛中可以使用，我觉得它应该是首选算法。增量算法与随机化的结合，更是得到了令人意外的结果，例二中竟然将一个 $O(n^3)$ 的算法优化到 O(n)，这让我们见识到了随机化在确定算法中的独特作用。

【参考文献】

1.《算法艺术与信息学竞赛》 作者：刘汝佳、黄亮

2.《计算几何--算法设计与分析》 作者：周培德

3．国家集训队 2008 年论文：《浅谈随机化在几何问题中的应用》 作者：顾研

【附录】

设 P 为平面上的一个点集；设 R 为另一个（允许为空的）点集，而且 $P \cap R = \phi$；设 $p \in P$，则下列命题成立：

① 如果存在某个圆覆盖了 P，而且其边界穿过 R 中的所有点，那么这样的圆

中必然存在唯一的最小者。我们将其记作 $md\ (P,R)$。

② 如果 $P \in md\ (P \setminus \{p\},R)$，那么 $md\ (P,R) = md\ (P \setminus \{p\},R)$。

③ 如果 $p \notin md\ (P \setminus \{p\},R)$，那么 $md\ (P,R) = md\ (P \setminus \{p\},R \cup \{p\})$

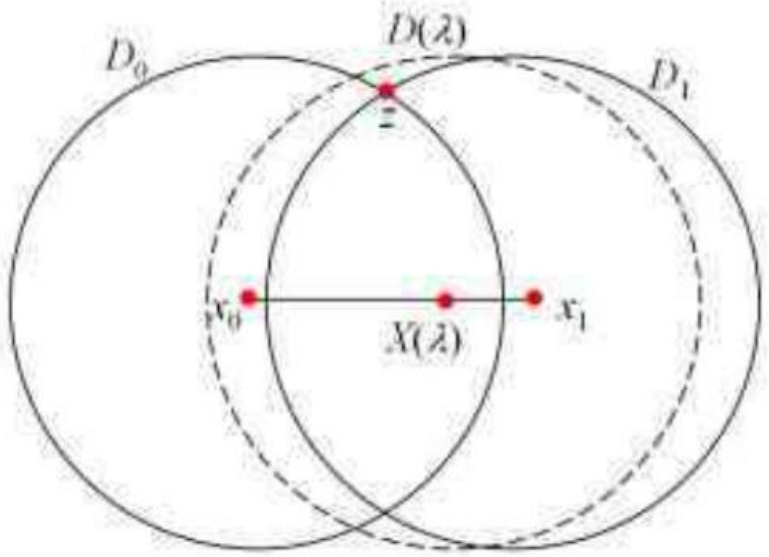


图 17

证明 ① 如图所示，假设存在半径相同的两个不同的覆盖圆 D_0 和 D_1，其圆心分别在 x_0 和 x_1。显然 P 中的所有的点必然落在交集 $D_0 \cap D_1$ 中。我们将按照下面的方法，构造出一系列连续的圆 $\{D(\lambda) \mid 0 \le \lambda \le 1\}$。取 D_0 和 D_1 的边界的一个交点 z，$D(\lambda)$ 的圆心是点 $x(\lambda) = (1-\lambda)x_0 + \lambda x_1$，其半径为 $r(\lambda) = d(x(\lambda),z)$，对于任何满足 0≤ ≤1 的 ，都有 $D_0 \cap D_1 \subset D(\lambda)$，因此作为一个特例， =0.5 时也成立。于是鉴于 D_0 和 D_1 都分别覆盖了 P 中的所有点，[illegible] 然如此。此外 $\partial D\left(\frac{1}{2}\right)$ 也必然经过 ∂D_0 和 ∂D_1 的每个交点。由于 $R \subset \partial D_0 \cap \partial D_1$，故有 $R \subset \partial D\left(\frac{1}{2}\right)$。

也就是说，$D\left(\frac{1}{2}\right)$ 不仅是 P 的一个覆盖圆，而且其边界同样会穿过 R 中的所有点。

然而，就半径而言，$D\left(\frac{1}{2}\right)$ 要严格小于 D_0 和 D_1。因此，一旦出现了半径相等的两个圆，而且它们的边界都各自穿过 R 中的所有点，就说明肯定存在另一个半径更小的覆盖圆，而且它们的边界都各自穿过 R 中的所有点。于是，最小覆盖圆 $md\ (P,R)$ 是唯一的。

② 令 $D = md\ (P \setminus \{p\},R)$，若 $p \notin D$，则 D 必然包含 P，而且其边界穿过 R 中

的所有点，不可能有任何更小的圆可以覆盖 P，而且其边界穿过 R 中的所有点。否则，这样的一个圆必然是 $P \setminus \{p\}$ 的一个包围圆，同时其边界穿过 R 中的所有点，这与 D 的定义矛盾。因此可以得到 $D = md(P, R)$。

③ 令 $D_0 = md(P \setminus \{p\}, R)$，取 $D_1 = md(P, R)$。再次考虑前面所定义的一系列 $D(\lambda)$。注意到，$D(0) = D_0$，$D(1) = D_1$。实际上，由这一系列的圆，定义了从 D_0 到 D_1 的一个连续变形的过程。根据假设，有 $p \in D_1$。因此，由其连续性，必然存在某个 $0 < \lambda^* \leq 1$，使得 p 正好落在 $D(\lambda^*)$ 的边界上。与①的证明同理，我们可以得到 $P \subset D(\lambda^*)$ 和 $R \subset \partial D(\lambda^*)$，对于任何 0< <1，$D(\lambda)$ 的半径必然会严格地小于 D_1 的半径。然而根据其定义，D_1 是 P 的最小覆盖圆，因此我们只有一种选择——$\lambda^* = 1$。也就是说，D_1 的边界必然穿过 p。