感受随机的美

—— 浅谈随机化思想在几何问题中的应用

广东中山一中 顾研

引入

随着信息学的发展，近几年，各种各样灵活的几何题目层出不穷。因此随机算法和随机化思想便有了表演的舞台。

随机算法的特点是：简单、快速、灵活和易于并行化，这些特点都会在论文中得到体现。

概览

第一部分 随机算法简介

数值概率算法

拉斯维加斯算法

蒙特卡罗算法

舍伍德算法

第三部分
模拟退火算法

第二部分
随机增量算法

随机增量算法的一个例子

Expensive Drink **(Beijing Site, 2007)** （经过抽象）

maximize $ax + by + cz$

s.t. $l_1 \le a_1x + b_1y + c_1z \le r_1$

$l_2 \le a_2x + b_2y + c_2z \le r_2$

……

$l_n \le a_nx + b_ny + c_nz \le r_n$ **(*n*≤100)**

单纯形法、内点法?

随机增量算法的一般步骤

发现问题的本质

↓

提出算法

↓

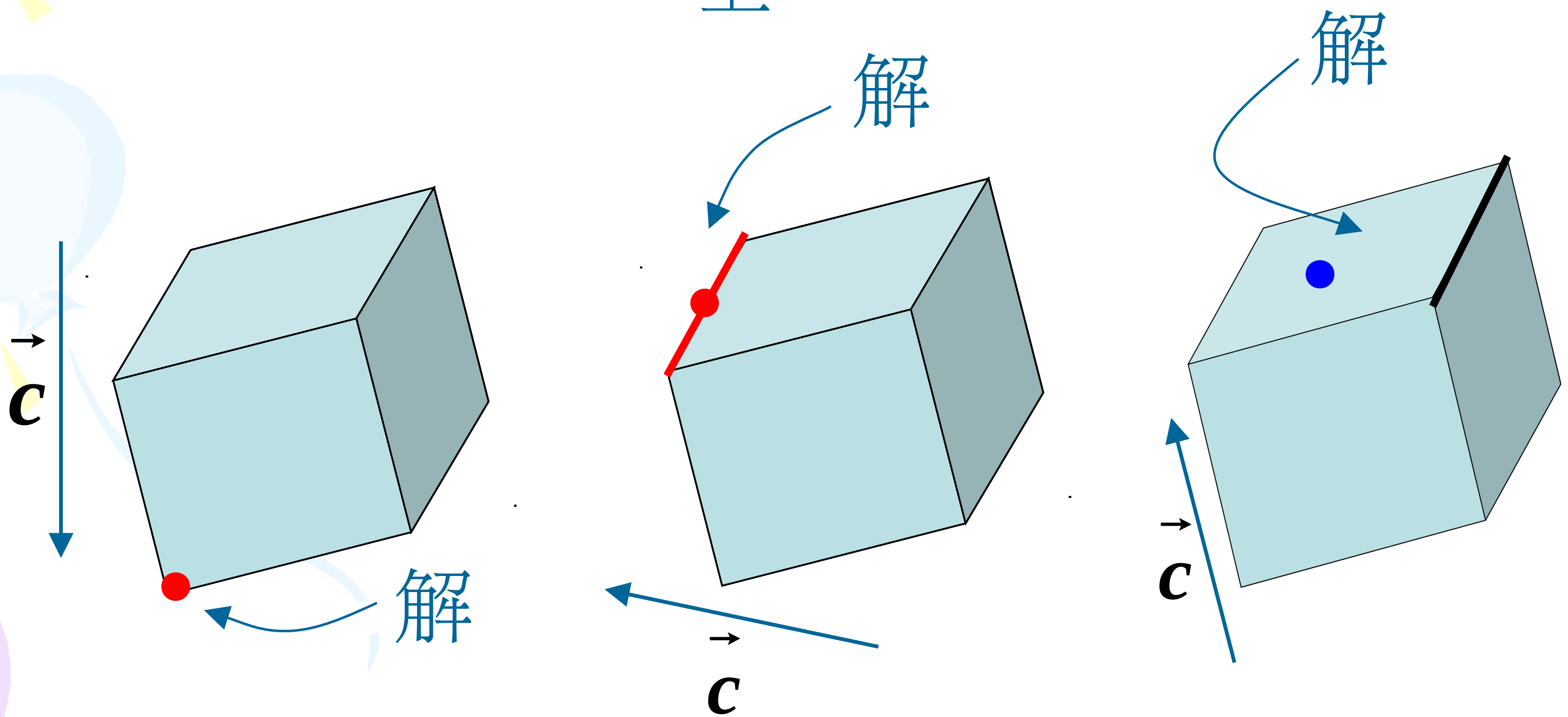
改造成增量算法

↓

加入随机

Expensive Drink

结论 **1**： 如果存在解，必然存在于三个平面的交点上。

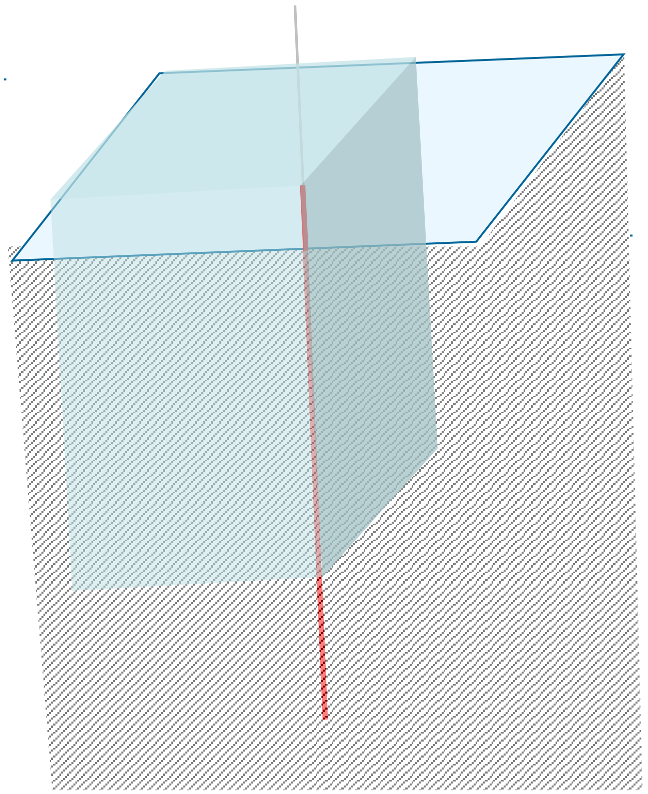


Expensive Drink

结论 **1**：如果存在解，必然存在于三个平面的交点上。

想法：枚举两个平面，
得到一条直线

枚举其余约束，
切割该直线。



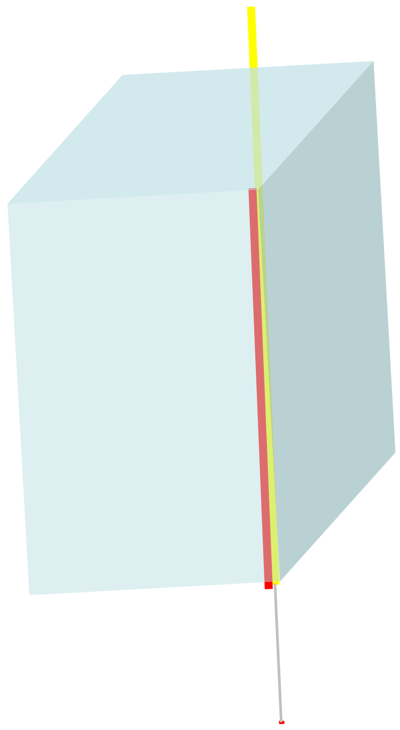
Expensive Drink

结论 **1**： 如果存在解，必然存在于三个平面的交点上。

想法：枚举两个平面，
得到一条直线。

枚举其余约束，
切割该直线。

直到最后剩下一
条线段。



Expensive Drink

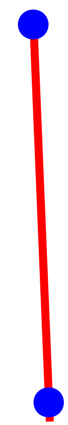
结论 **1**： 如果存在解，必然存在于三个平面的交点上。

结论 **2**： 只有线段的两个端点可能成为解。

直线数量 $O(n^2)$

切割复杂度 $O(n)$

总复杂度 $O(n^3)$

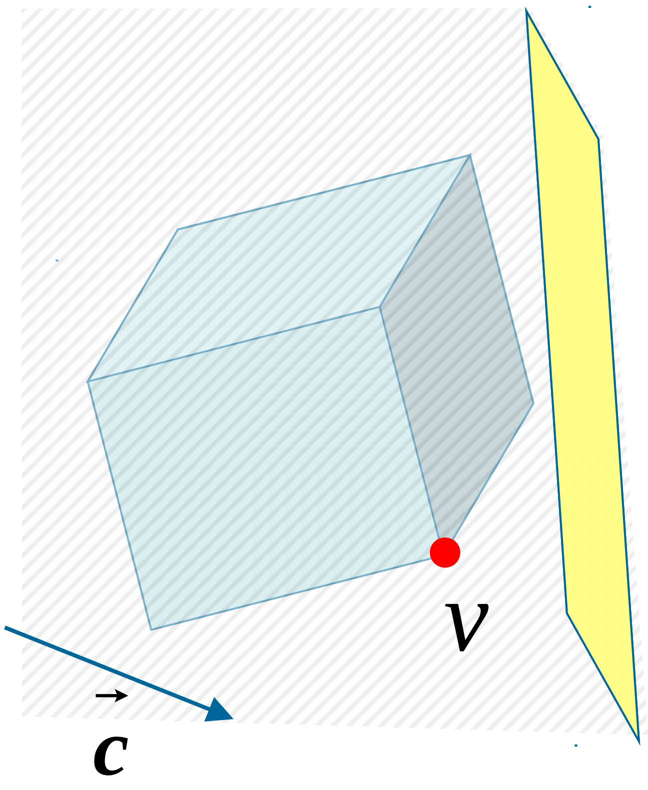


仍需要提高

Expensive Drink

症结：没有利用到之前已经计算的结果

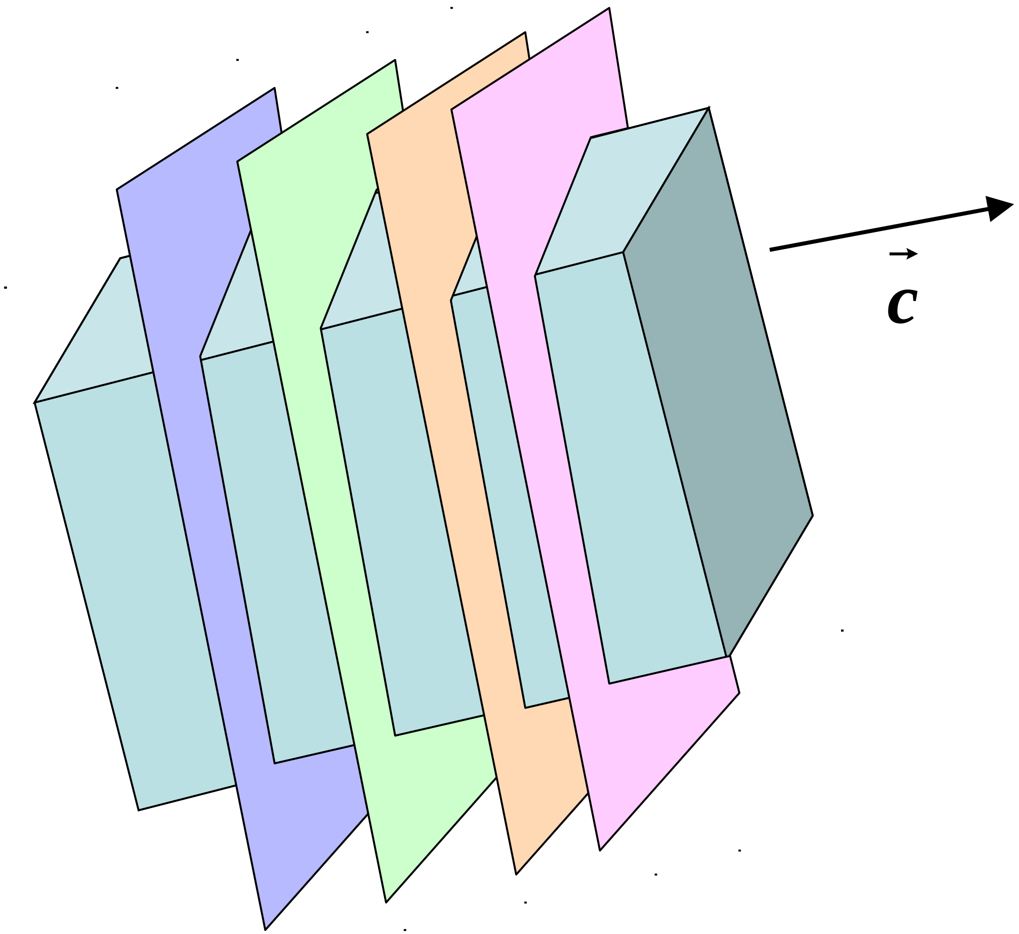
对症：引入增量算法。依次加入半空间的时候，若原先的最优解为 $\boldsymbol{v}$，且满足当前的约束，就没有必要枚举平面上的直线了。



Expensive Drink

复杂度仍旧为 $O(n^3)$

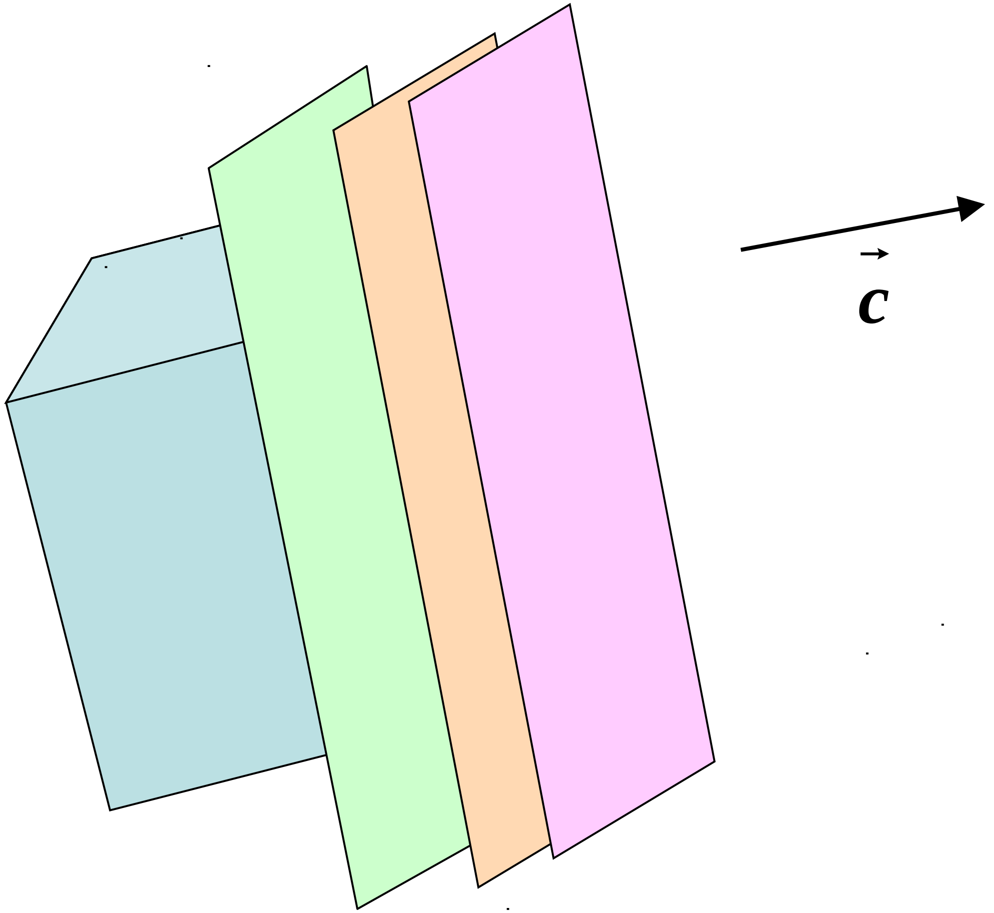
对策：随机插入半空间的顺序



Expensive Drink

复杂度仍旧为 $O(n^3)$

对策：随机插入半空间的顺序



复杂度分析

取随机变量 X_i，若满足前 **i-1** 条约束的最优解满足第 i 条约束，则 $X_i=0$，否则 $X_i=1$。

时间复杂度为 $\sum_{i=1}^{n} O(i^2)\cdot X_i$

根据期望的线性率有 $E\left[\sum_{i=1}^{n} O(i^2)\cdot X_i\right]=\sum_{i=1}^{n} O(i^2)\cdot E[X_i]$

$E[X_i]$ 是多少呢？最优解由 **3** 个约束构成，恰好包括第 i 条约束的概率就是 $\frac{3}{i}$。

$$\sum_{i=1}^{n} O(i^2)\cdot\frac{3}{i}=O(n^2)$$

Expensive Drink

- 在本题中，增量算法架筑起了线性规划问题与经典几何知识的桥梁，随机化思想则消除了输入数据的顺序对于复杂度的影响。本题也体现出随机算法简单、快速（相对于单纯形法）的特点。

下面将介绍论文中的第二个算法：模拟退火算法。

模拟退火算法简介

模拟退火（*Simulated Annealing*）算法是模仿自然界中固体退火的原理的一种元启发式（*Meta-Heuristics*）算法。

① 初始化：初始充分大的温度 ***T***，初始解状态 ***S***，迭代数 ***L***

② ***for k=1 to L*** 做③至⑤

③ 产生新解 ***S'*** 并计算评价函数 ***C(S')***

④ 若 ***C(S')<C(S)*** 则接受 ***S'*** 作为新的当前解，否则以概率 $e^{-\frac{\Delta t}{T}}$ 接受 ***S'*** 作为新的当前解

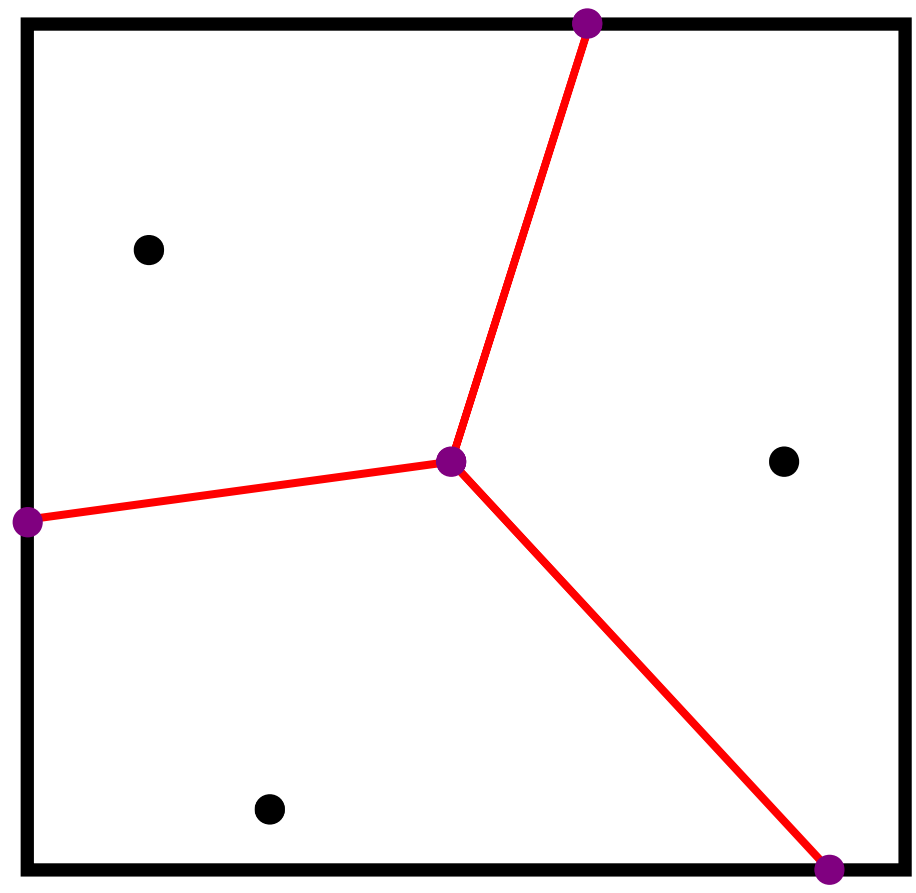
⑤ 如果满足终止条件则输出当前解作为最优解，结束程序

⑥ ***T*** 逐渐减少，然后转②

最小距离问题

求区域中一点，到某个点集中的点的最小距离最大

经典方法：构造**Voronoi**图解，并对顶点集合进行判断。

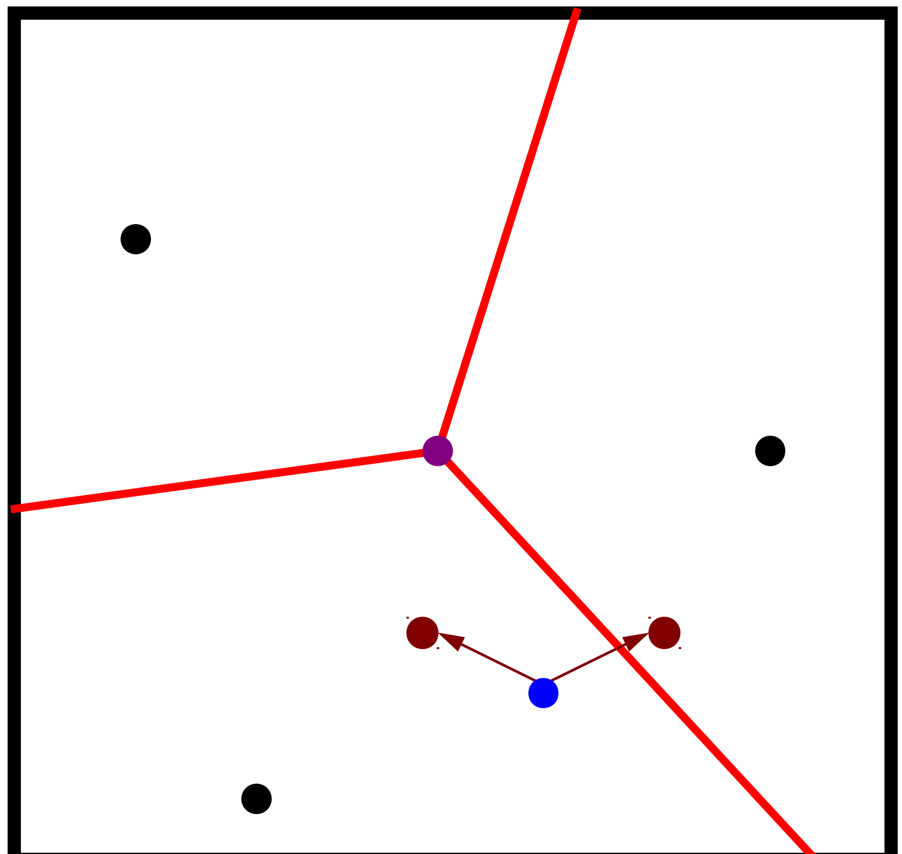


最小距离问题

求区域中一点，到某个点集中的点的最小距离最大

通过类比的思想，引入模拟退火算法：随机初始解，温度 ***T*** 定义为调整向量的模长。估价函数定义为到最近点的距离。

如果函数值变大，则更新原



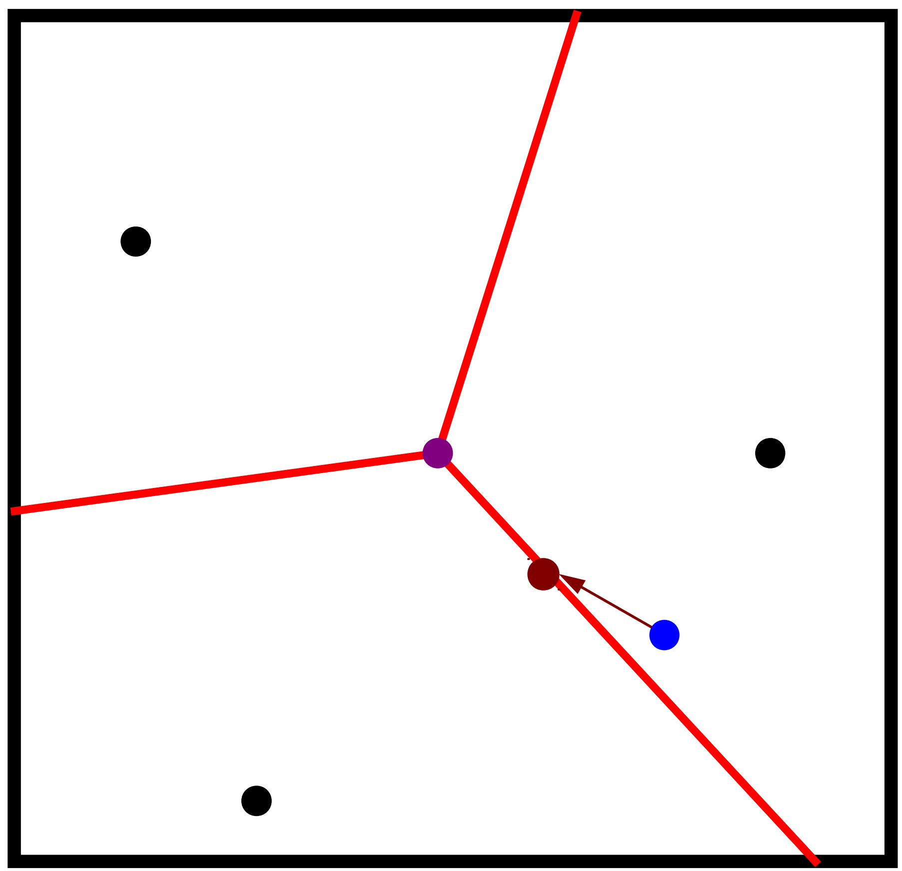
最小距离问题

求区域中一点，到某个点集中的点的最小距离最大

通过类比的思想，引入模拟退火算法：

随机初始解，温度 ***T*** 定义为调整向量的模长。估价函数定义为到最近点的距离。

如果函数值变大，则更新原



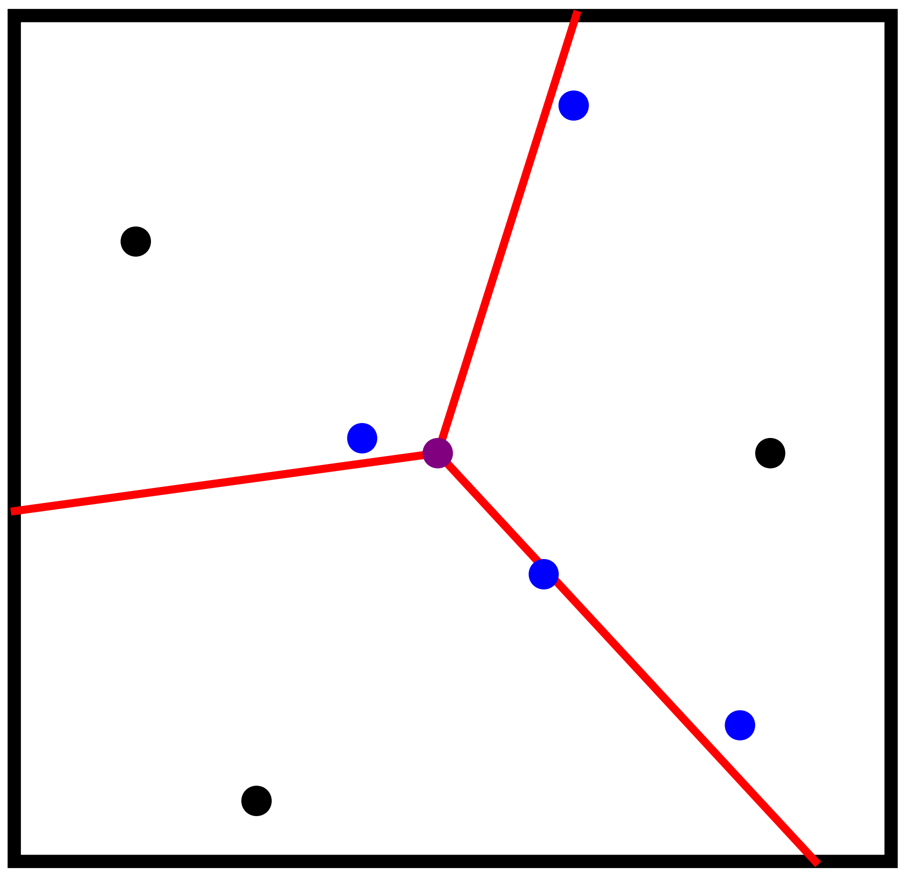
最小距离问题

求区域中一点，到某个点集中的点的最小距离最大

通过类比的思想，引入模拟退火算法

：模拟退火算法有并行性。

不断重复这一过程，直到步长足够小。取当前最优解作为答案。



模拟退火算法的应用

模拟退火算法有很强的可移植性。

对应于

最小距离最大 ⟷ 最近点 **Voronoi** 图解

最大距离最小 ⟷ 最远点 **Voronoi** 图解

第 ***k*** 大距离最小 ⟷ ***k*** 阶 **Voronoi** 图解

经过反射后距离最小

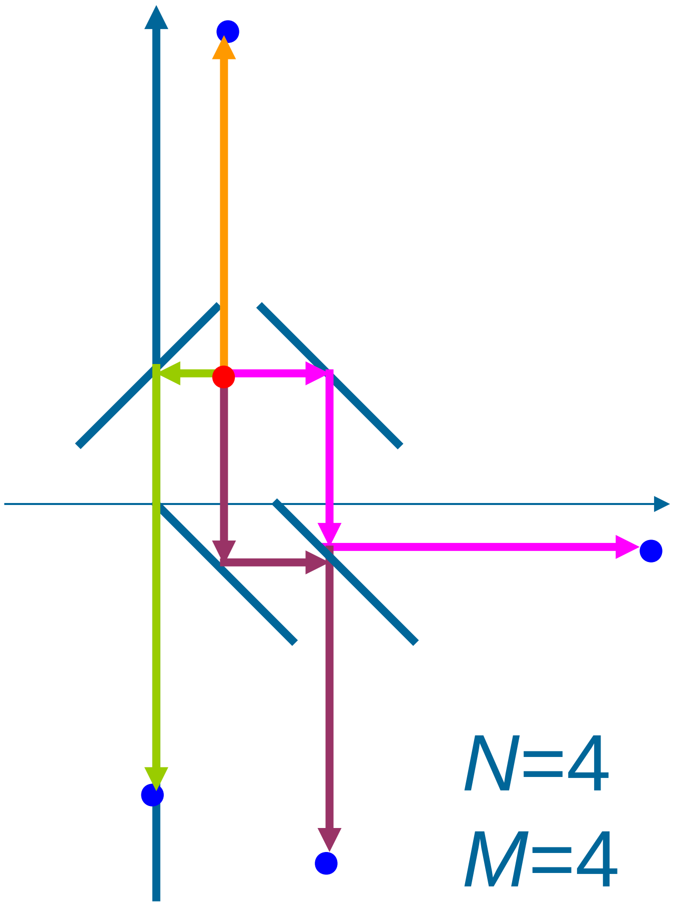
和距离最小 ⟷ 

倒数和距离最小

模拟退火算法的例子

激光坦克（**CTSC2007**）

在平面上有 ***N*** 个坦克，***M*** 个镜子。要求在平面内放置一个激光发射器，使得它在发出的每束激光经过不超过 ***k*** 次反射后击中所有目标的前提下，距离的最大值最小。

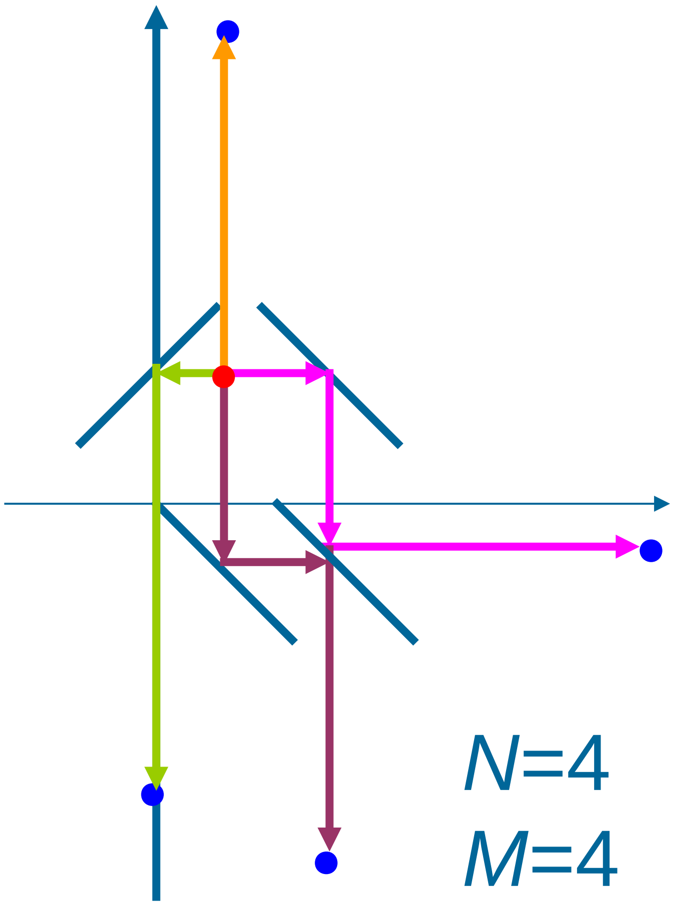


N=4
M=4
k=2

模拟退火算法的例子

激光坦克（**CTSC2007**）

本题是一个最大距离最小的问题，如果不考虑镜子的因素，可以使用最远点 **Voronoi** 图或前面的随机增量算法来解决，但是镜子的存在使得问题非常棘手。

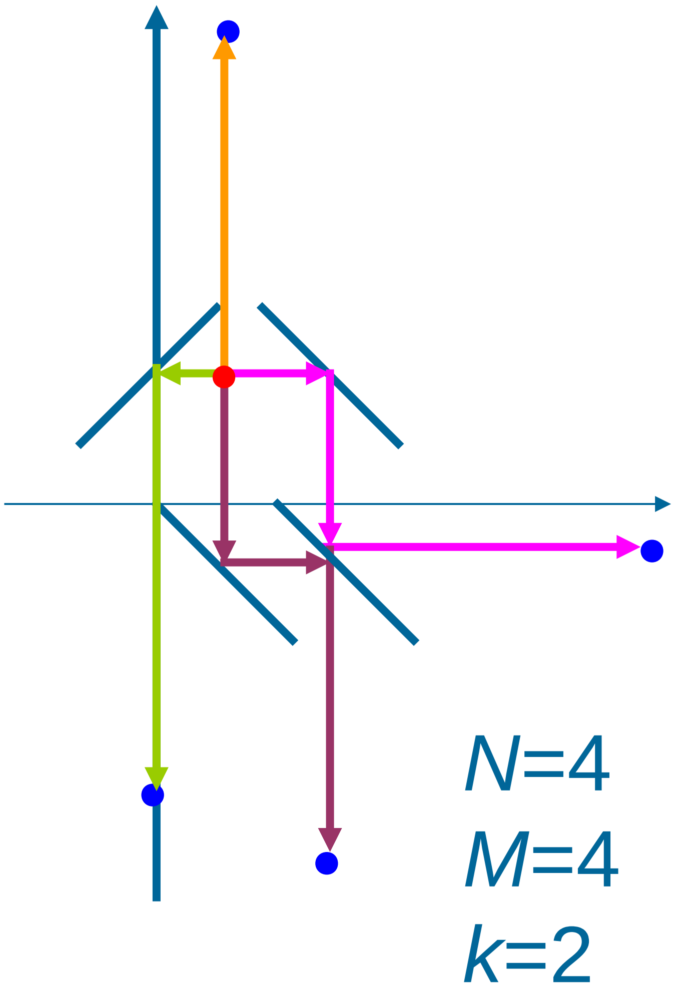


N=4
M=4
k=2

模拟退火算法的例子

激光坦克（**CTSC2007**）

此时，模拟退火算法的可移植性的优势就体现了出来，我们可以在主算法的框架上，分别独立编写与镜子不同次数相交的评价函数。



激光坦克的得分与代价

Testcase	*k*	不处理反射	处理一次反射	处理两次反射
6	0	10	10	10
2	1	10	10	10
3	1	0	10	10
7	1	1	10	10
5	2	10	10	10
8	2	6	10	10
1	3	9	10	10
4	3	10	10	10
9	3	0	0	10
10	5	0	0	0
总得分		56	80	90
代码长度	90	160	240	300

总结

- 本文通过几道例题，以及体现出的一种思想，希望能为大家打开一扇窗，在遇到几何问题的时候多一种思路。当然，随机化思想的灵活运用，是在对于经典问题熟练掌握的前提下的，因为创新永远建立在扎实的基础之上。

谢谢！

Expensive Drink 题目描述

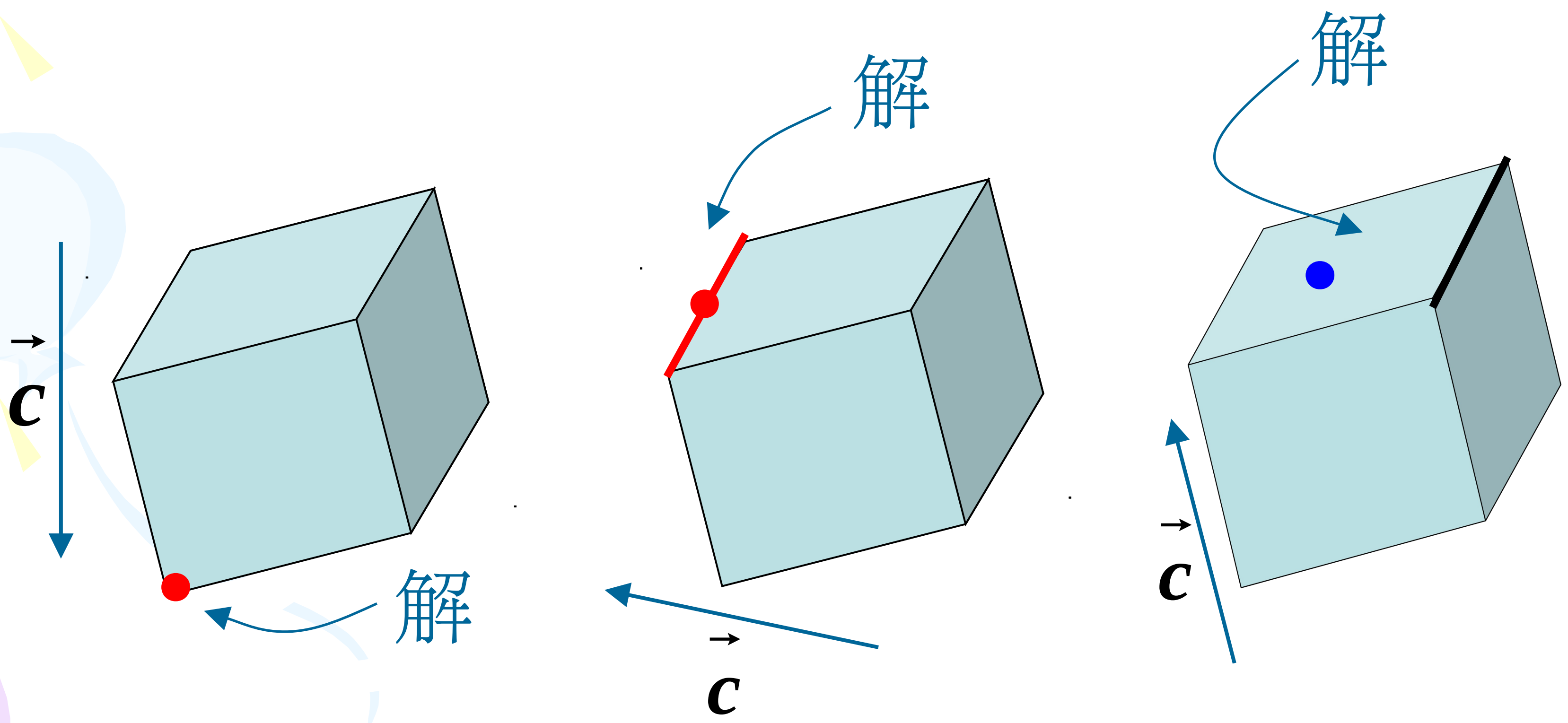
有 **3** 种物品的价格（设为 ***x, y, z*** ）要满足 **n** 组约束

$$L_i \le a_i x + b_i y + c_i z \le R_i$$

且 $$0 \le x \le y \le z$$

求 $ax + by + cz$ 的最大值

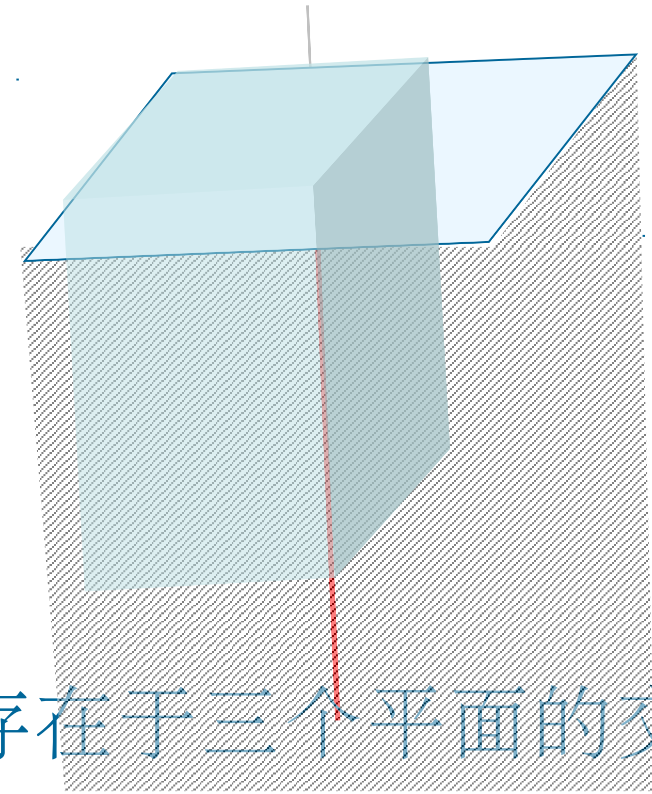
Expensive Drink



结论 1： 如果存在解，必然存在于三个平面的交点上。

Expensive Drink

想法：枚举两个平面，
得到一条直



枚举其余约束，
切割该直线。

结论 **1**：如果存在解，必然存在于三个平面的交点上。

Expensive Drink

结论 **1**： 如果存在解，必然存在于三个平面的交点上。

想法：枚举两个平面，

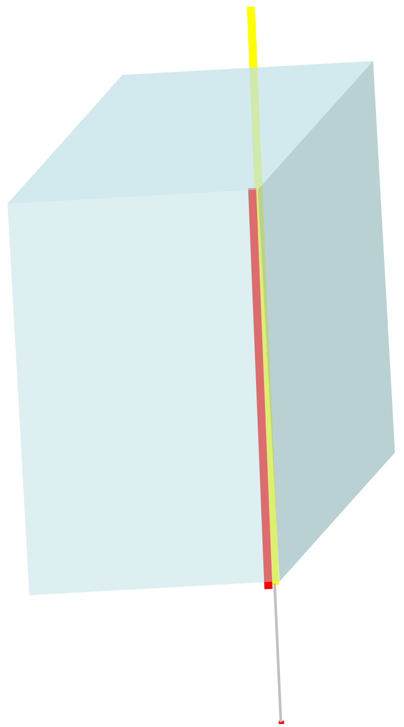
得到一条直线。

枚举其余约束，

切割该直线。

直到最后剩下一

条线段。



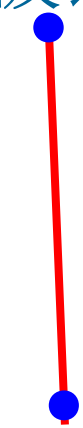
Expensive Drink

结论 **2**：只有线段的两个端点可能成为解。

引理 **1** 只有线段的两个端点可能是的目标函数的

最大值。

引理 **2** 不会有某三个平面的交点被遗漏。



Expensive Drink

引理 1　只有线段的两个端点可能是的目标函数的最大值

证明　设线段的两个端点分别在 p_0 和 p_1，我们用 $\{P(\lambda)|0 \le \lambda \le 1\}$ 表示线段上的点，其中 $P_0 = p_0$、$P_1 = p_1$、$\overline{P(\lambda)P_0} = \lambda\overline{P_0P_1}$。因为 $\{P(\lambda)\}$ 是空间中的一条直线，求导后，x、y、z 对于 λ 的变化量都是常数。设 $C(\lambda)$ 为 $P(\lambda)$ 的目标函数的值，则 $C'(\lambda)$ 也为常数。不妨设 $C'(\lambda) = k$。

若 $k=0$，则线段上每个点的解相同。

若 $k>0$，则 $C(1)$ 是线段上的最优解。

若 $k<0$，则 $C(0)$ 是线段上的最优解。

Expensive Drink

引理 2 不会有某三个平面的交点在计算中被遗漏。

证明 枚举两个平面，并用其余约束切割公共棱，设线段的两个端点分别在 p_0 和 p_1，设该三个平面的交点为 p，分三种情况：

① p 为 p_0 或 p_1 中的一个。已枚举到。

② p 在 $\overline{p_0 p_1}$ 上。由**定理 1** 知必然不会比 p_0 和 p_1 的解中较大的优。

③ p 不在 $\overline{p_0 p_1}$ 上。则必然有某个（些）约束条件不能满足，解不合法。

具体的实现

因为空间中的直线情况比较多、比较复杂，因此我们可以使用参数方程进行统一表示。

$$\begin{cases} x = x_0 + x_1 \cdot t \\ y = y_0 + y_1 \cdot t \\ z = z_0 + z_1 \cdot t \end{cases}$$

这样，我们对直线的切割就转化成为对于参数值求交的过程。

最后是求解参数方程的过程。首先我们假设枚举的两个平面不平行，我们任意消去 ***x*** 、 ***y*** 、 ***z*** 中的一个，得到一个二元（一元）一次方程。取任意一个自由元的方程的系数，经过两次回代即可求出直线的参数方程。

三维线性规划 $O(n)$ 的算法

- 这题理论上存在 $O(n)$ 复杂度的方法。但是该算法有两点弊病：
- **1)** 时间复杂度中隐藏的常数巨大。本题中在时间上的优势微小。（n 仅 **100** 。）
- **2)** 编程复杂度过大。其实 $O(n)$ 的算法并不难想：每次加入一个半空间后，如果先前的解不成立需要更新，此时就是要将目标向量在平面上的投影作为新的目标向量，将其他半空间转换成半平面做一次二维线性规划。几次空间和平面间的转换与旋转，将该算法仅仅保留在理论上。
- 我们使用随机思想是希望事半功倍、化繁为简，因此本算法有悖于我们的初衷。而且无论在信息学还是 **ACM** 赛场上比赛的时间都是有限的，因此本算法虽然存在，但并不值得推广。

数值概率算法

- 数值概率算法常用于数值问题的求解。这类算法所得到的往往是近似解。而且近似解的精度随计算时间的增加不断提高。在许多情况下，要计算出问题的精确解是不可能或没有必要的，因此用数值概率算法可得到满意的解。
- 举个例子：计算π的近似值时，我们可以在单位圆的外接矩形内随机撒 n 个点，设有 k 个点落在单位圆内，可以得到π????4 k/n 。

舍伍德算法

- 舍伍德算法总能求得问题的一个解，且所求得的解总是正确的。当一个确定性算法在最坏情况下的计算复杂性与其在平均情况下的计算复杂性有较大差别时，可以在这个确定算法中引入随机性将它改造成一个舍伍德算法，消除或减少问题的好坏实例间的这种差别。舍伍德算法精髓不是避免算法的最坏情况的发生，而是设法消除这种最坏行为与特定实例之间的关联性。舍伍德算法的一个最广泛的应用就是快速排序的随机化实现。

随机洗牌算法

- 这个问题不复杂，以下代码就可以以线性的时间复杂度得到一个 **1~n** 的随机排列。（记录在数组 ***O*** 中。）

Algorithm Random_shuffle
for i ← 2 to n
  交换 **O[i]**，**O[random(i)]**
（其中 **random(n)** 返回一个 **1~n** 的随机数。）

蒙特卡罗抽样

- 它的基本思想是，对于所求的问题，通过试验的方法和大样本来模拟，得到这个随机变量的期望值，并用它作为问题的解。它是以一个概率模型为基础，按照这个模型所描绘的过程，通过模拟实验的结果，作为问题的近似解的过程。

模拟退火算法的原理

- 模拟退火算法是一种元启发式（ ***Meta-Heuristics*** ）算法，来源于固体退火原理，将固体加温至充分高，再让其徐徐冷却。加温时，固体内部粒子随温升变为无序状，内能增大，而徐徐冷却时粒子渐趋有序，在每个温度都达到平衡态，最后在常温时达到基态，内能减为最小。根据 ***Metropolis*** 准则，粒子在温度 ***T*** 时趋于平衡的概率为 $e^{-\frac{\Delta t}{T}}$
- ，其中 ***E*** 为温度 ***T*** 时的内能， ***ΔE*** 为其改变量， ***k*** 为 ***Boltzmann*** 常数。

元启发式算法

- 元启发式算法（*Meta-Heuristics*）是一种启发式策略，意思就是指导启发式算法进行工作的方法。常见的元启发式算法有：
- 模拟退火算法
- 遗传算法
- 蚁群算法
- **PSO(** 粒子群优化 **)**

精确度分析的一个例子

- 最优解附近（如点 ***A***， ***B***）的点非常稀少且距离很远，因此有候选解在它周围（所在的 **Delaunay** 三角剖分区域内）的概率是很大的。而且此时的距离比较大，我们对方向进行多次尝试，因此调整出去的概率也很小。

精确度分析的一个例子

- 如图，假设一次随机调整成功的概率为 ***P***，则

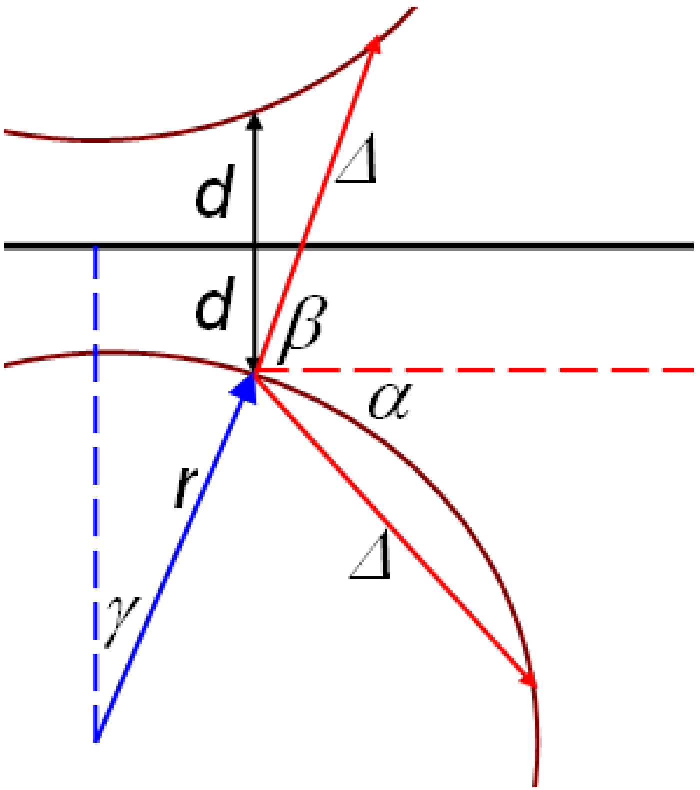
$$P \geq \frac{\alpha+\beta}{2\pi}$$。（有可能向左调整）。

$$\alpha = \pi - \left(\frac{\pi}{2} - \gamma\right) - \arccos\left(\frac{\Delta}{2r}\right)$$

$$= \frac{\pi}{2} + \gamma - \arccos\left(\frac{\Delta}{2r}\right)$$

$\beta \geq \alpha$（当 $d = 0$ 时 $\beta = \alpha$）

$$\Rightarrow P \geq \frac{1}{2} + \frac{\gamma}{\pi} - \frac{\arccos\left(\frac{\Delta}{2r}\right)}{\pi}$$



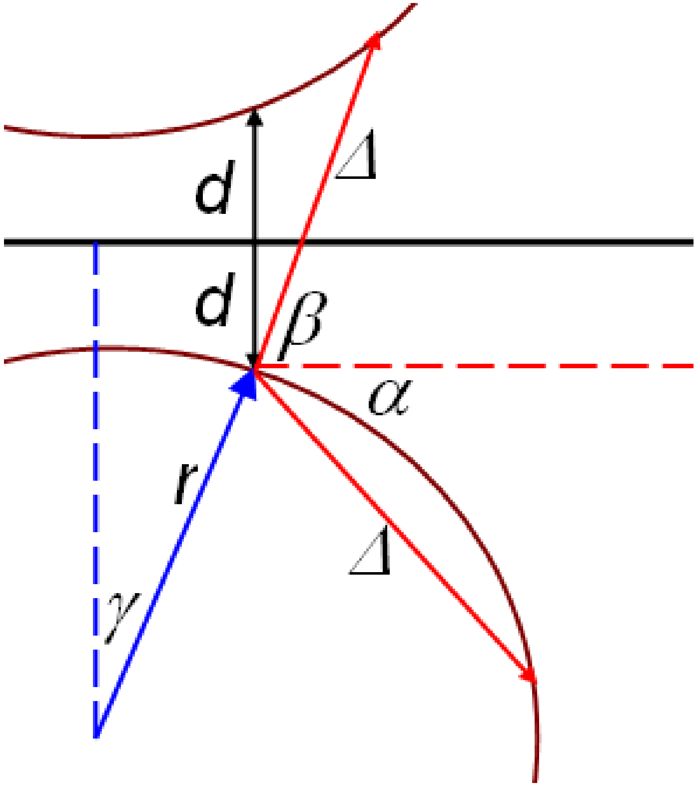
精确度分析的一个例子

- 若我们的 L 取 30

➔…………(d=0,γ=0)

➔Δ ??最小值 0.085r 。

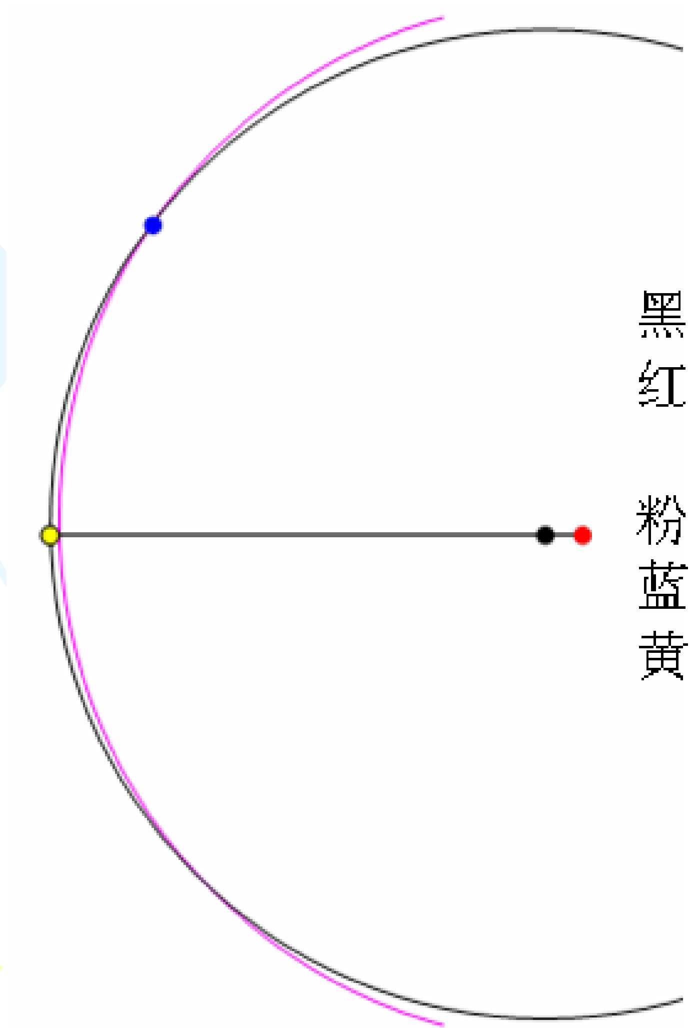
- 1） 因为此时两者垂直，因此对于答案的影响很小。
- 2） 我们使用了放缩过程，把γ、*d* 都当成 0 计算，因此实际的调整概率还要更高。



精确度分析的一个例子

- 但是如果题目的精度要求非常高，怎么办呢？既然很难随机到向量和 **Voronoi** 边平行，我们可以直接枚举平行于 **Voronoi** 边的向量，虽然在时间上付出一点代价，但是在调整成功的概率和解的精度无疑将大大提高。当然对于普通的题目（本节三道例题 **Run Away** 、 **Empire Strikes Back** 、激光坦克），普通的随机调整就可以了。

URAL1520： Empire Strikes Back



黑色圈： 边界

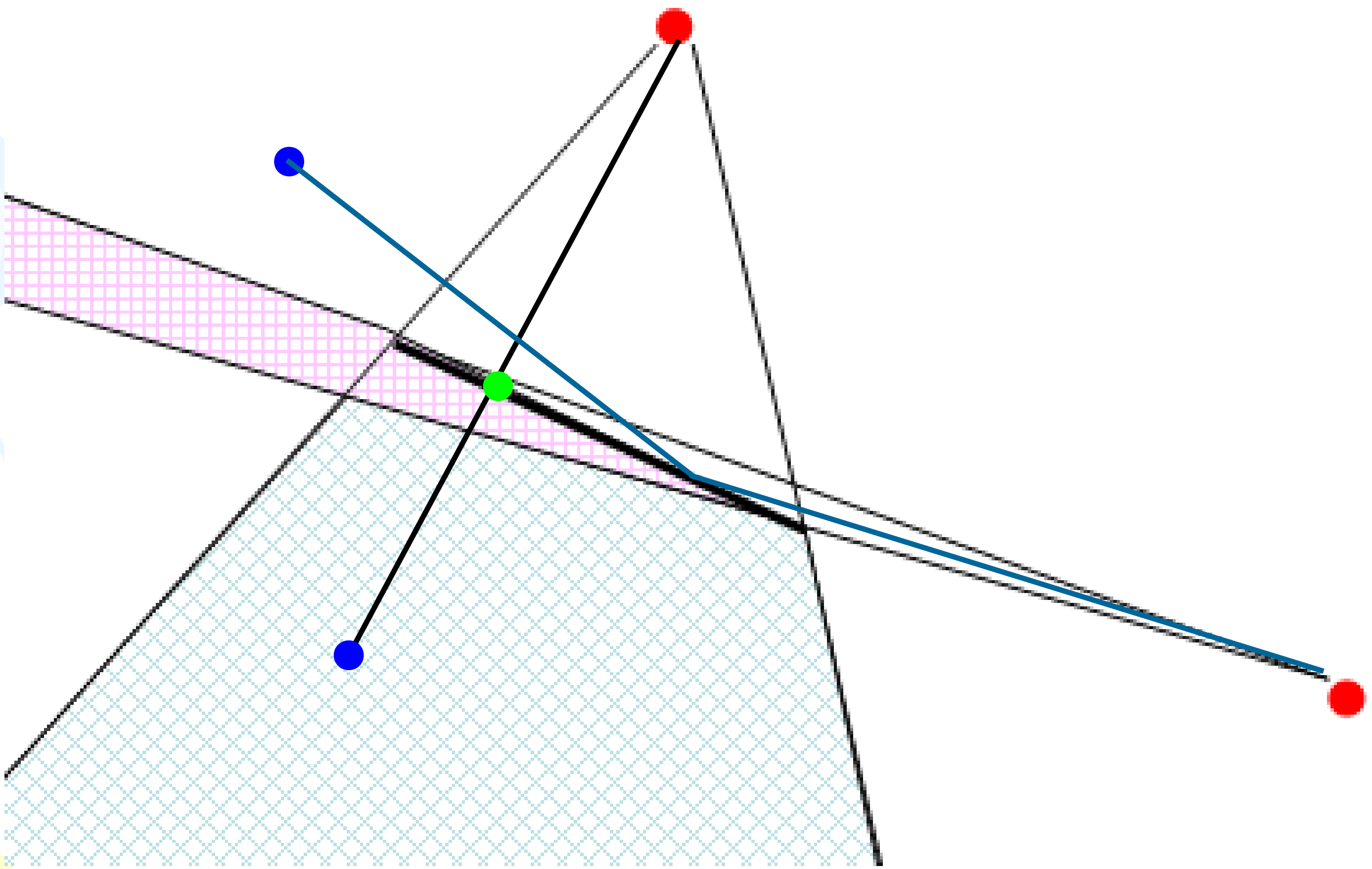
红色点： 左边边界上的
距离最大的点

粉色圈： 圆心为红色点的圆

蓝色点： 某候选点

黄色点： 答案所在的位置

激光坦克



激光坦克

多次迭代
逐步求精

Testcase	*k*	*N*	*M*
6	0	200000	0
2	1	2	50
3	1	2	1000
7	1	10000	3
5	2	3	30
8	2	3	70
1	3	4	4
4	3	2	35
9	3	18	30
10	5	25	100

可以使用随机化思想的几何题目

- 1 ***Expensive Drink***
- 2 最小外接圆 / 球
- 3 ***Run Away***
- 4 ***Empire strikes back***
- 5 激光坦克
- 6 ***A star not a tree?***
- 7 ***Mammoth Hunt***
- ***TopCoder Marathon*** 中数道几何题目

随机增量算法

模拟退火算法

调整法