

# 计算几何课大作业实验报告

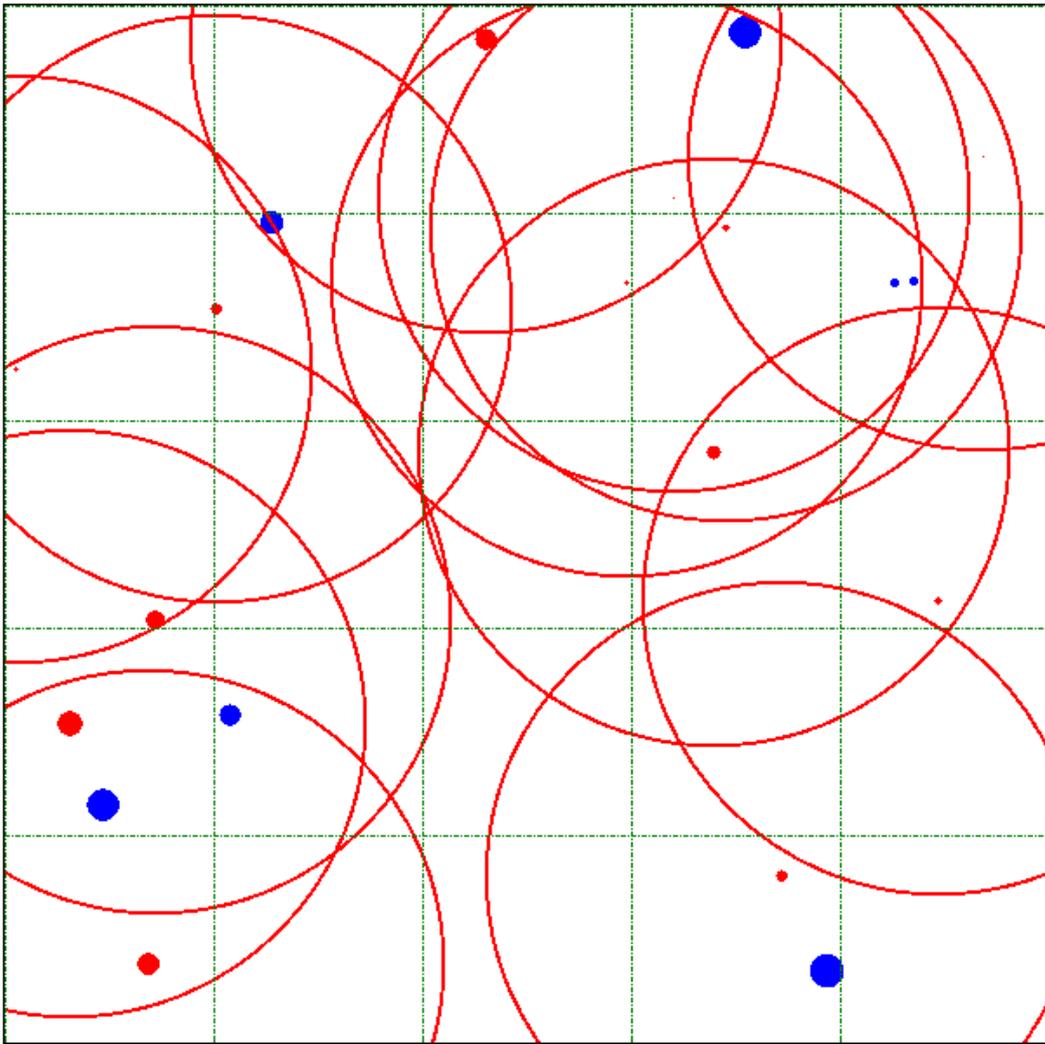
## ——单位圆盘图上带权最小支配集的近似算法

计研四 宋浩 2008310484

计研四 陈世腾 2008310488

### 问题描述:

在平面上有一个图，图上有一些顶点，每一个顶点有一个权重。要求从这些顶点中选出若干个来，用这些顶点做为圆心做单位圆，可以覆盖其他所有顶点，且选中顶点的权重总合尽可能小。如下图，就是一个符合条件的覆盖：



图中点为要求覆盖的点，红点为被选中的圆心。

### 问题的应用:

说一个比较简单的应用得例子：一个地区有一些信号发射站，每个发射站的发射都只能确保

将信号发射到一个相同大小的圆内。而且由于地理位置的不同，每个发射站的发射功率不同，因此发射的成本也不一样。现在有一个消息，从这些发射站中选出几个来广播，要求所有发射站收到，问如何能使发射成本最低？

## 相关工作：

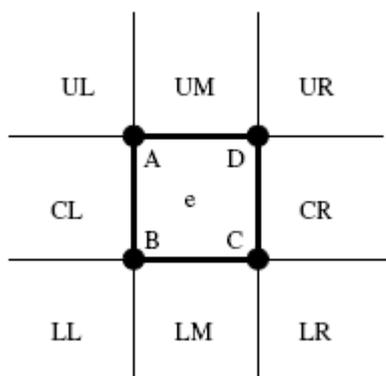
- 1990 年，Clark, Colbourn 和 Johnson 等人在名为 Unit Disk Graphs 的论文中证明了此问题是 NP 完全的 [1]
- 2006 年，Ambuhl 等人在一篇论文中给出了此问题的一个时间复杂度为多项式级的近似算法，近似比为 72 [2]
- 2008 年，高晓峰（音译）等人给出了此问题的近似比为  $(6+\epsilon)$  的多项式级近似算法 [4]
- 2008 年，戴德承和余昌远在一篇论文中给出了此问题的近似比为  $(5+\epsilon)$  的多项式级近似算法 [3]

由于在理论研究中，大家一般只重视近似比，对于时间复杂性只要求是多项式级别即可，对于具体的幂指数量级并不关心，因此一些近似比较好的算法（如上面提到的后两个）复杂性一般太高，不适合实际使用。

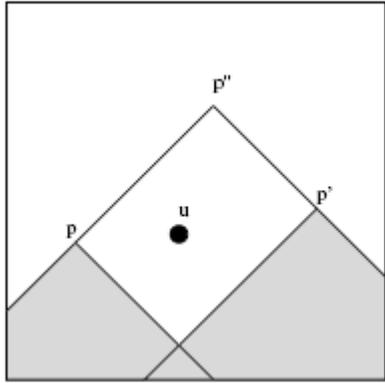
我们参考了以上一些算法设计中的一些技巧，在权衡时间复杂性和近似比的条件下，提出了如下算法。

## 算法描述：

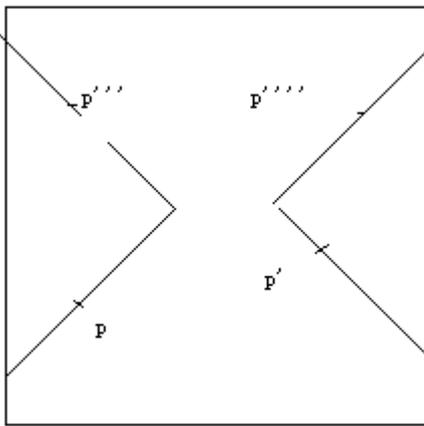
1. 将平面划分成边长为  $\sqrt{2}/2$  的正方形，确定每个哪些点在哪个正方形中。这一步需要  $O(n)$  时间。
2. 对于每个正方形，运行以下的覆盖子算法，计算覆盖这个正方形内的点的最优解，这些正方形最多有  $n$  个：
  - 2.1 计算选中正方形内点覆盖方法的最优值。由于正方形内任何一个点都能覆盖整个正方形，因此取遍其中所有点然后比较即可。
  - 2.2 对于正方形外的覆盖，方法比较复杂。首先，如下图，把正方形  $e$  外的点分成 8 块。



然后，猜测选出正方形  $e$  中最左上的一个被下方（LL, LM, LR）覆盖而没有被 CL 覆盖的点  $p$ ，最右上的一个被下方覆盖而没有被 CR 覆盖的点  $p'$ 。  $p$ ,  $p'$  各做一条 45 度的斜线交于  $p''$ 。易于证明， $p''$  下方的三角形没有被 CL, CR 覆盖。这要猜测  $n^2$  次。



同理在上方猜出  $p'''$  和  $p''''$ ，猜  $n^2$  次。这四点下方的三角形连接成一个沙漏状。



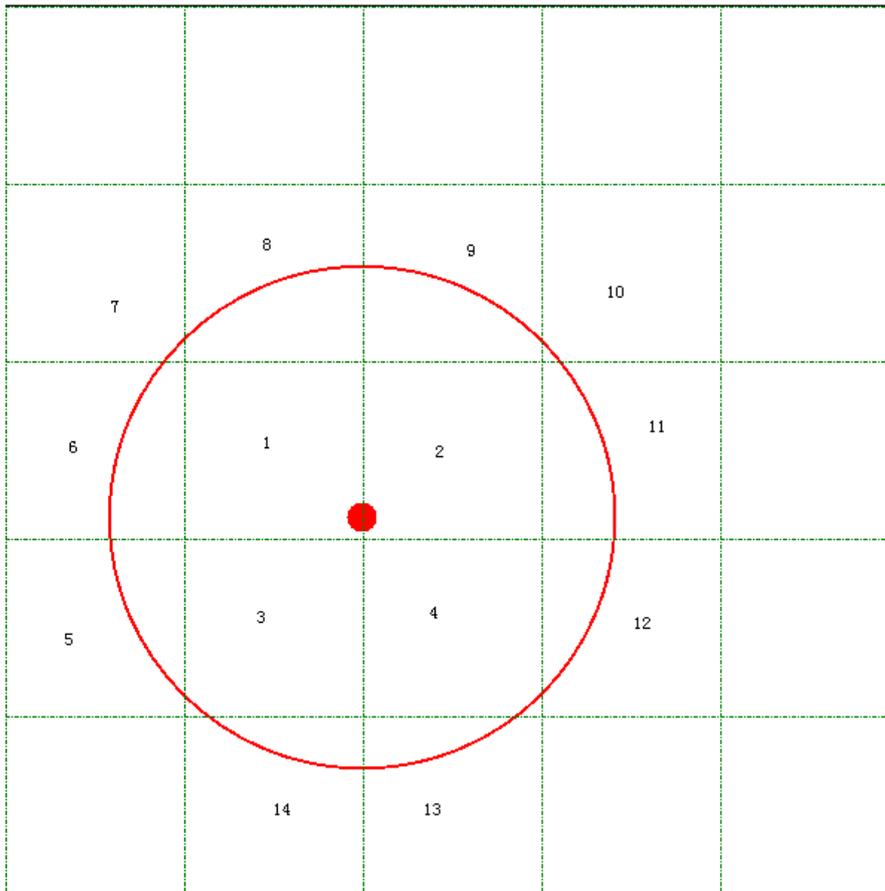
这个沙漏只由上下方的圆覆盖，用动态规划可以算出沙漏的最佳覆盖。然后同样用动态规划算出沙漏外左边和右边的两个三角形的最优覆盖。将这两者合并。动态规划的运算时间大约是  $O(n^5)$  级别。

由于 UR, UL, LL, LR 可能会被计算两次，因此这个解的代价最多是最优解的 2 倍。

这一步的总时间不大于  $O(n^9)$ 。

2.3 比较这两者，取较好的一个。

3. 合并所有正方形的结果。由于每个圆最多覆盖 14 个这样的正方形，因此这里最多产生 14 倍的偏差。

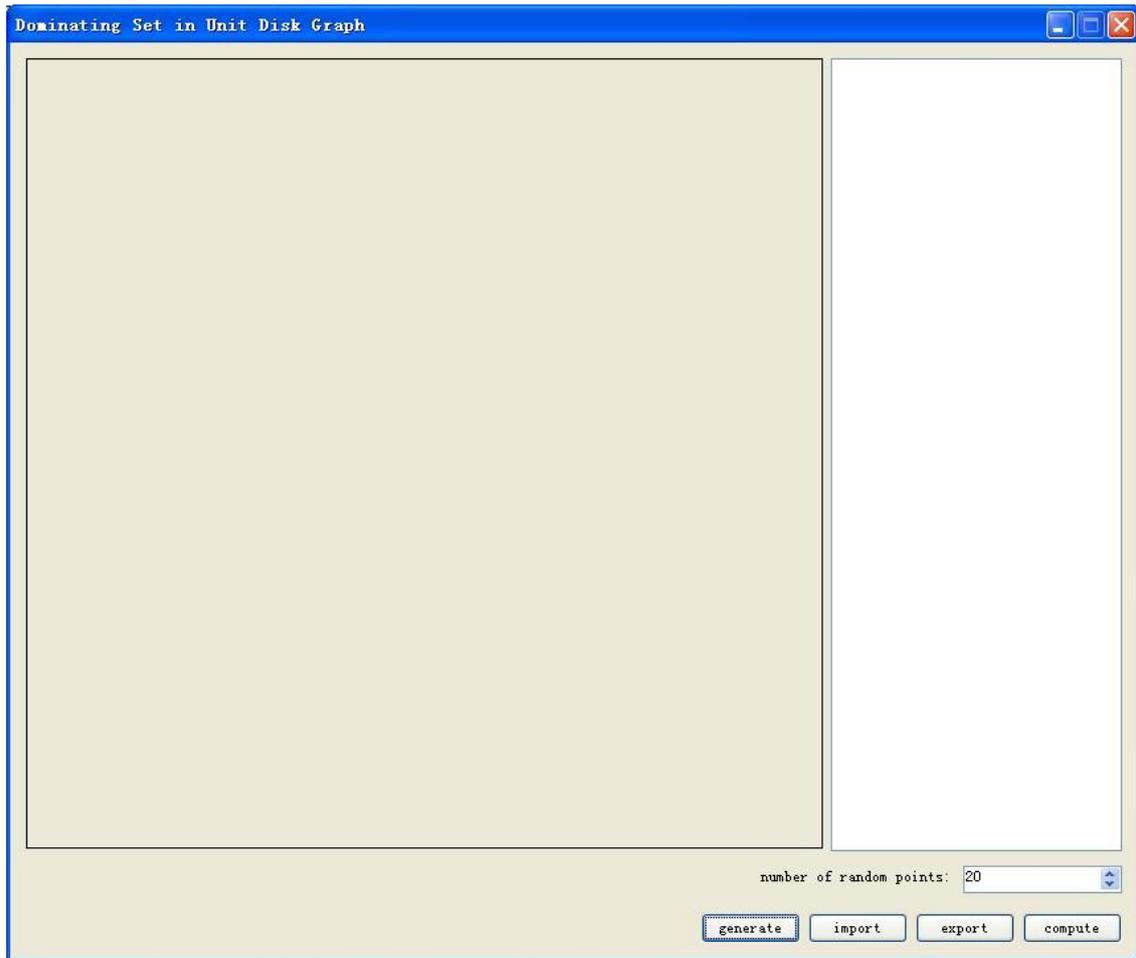


因此，综上，这个算法的时间复杂度不超过  $O(n^{10})$ ，其结果最多是最优解的 28 倍。

### 程序实现：

我们用 C# 语言实现了上述算法，在 Microsoft Visual C# 2008 Express Edition 开发环境下开发了程序，并使用 .NET 框架类库中的 Windows Forms 组件开发了图形界面。

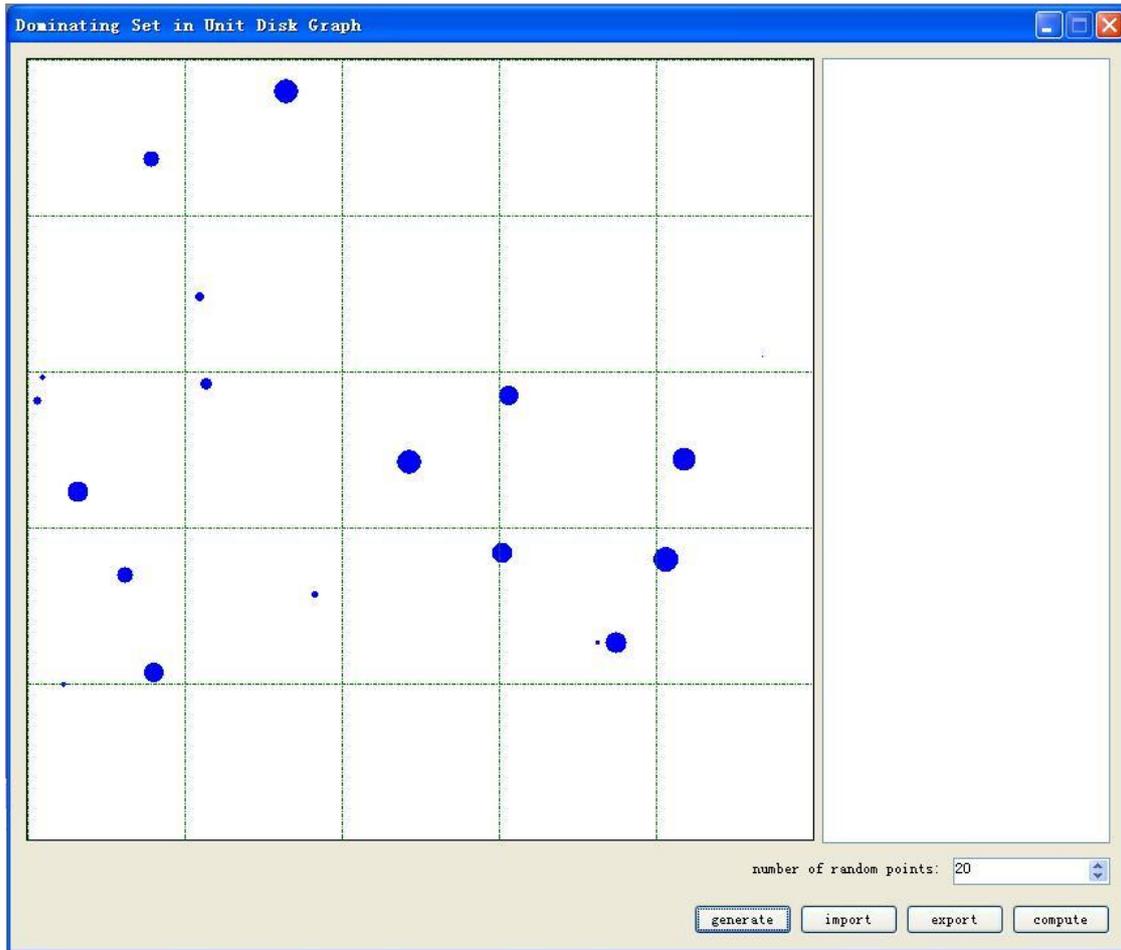
程序启动后，将会出现如下界面：



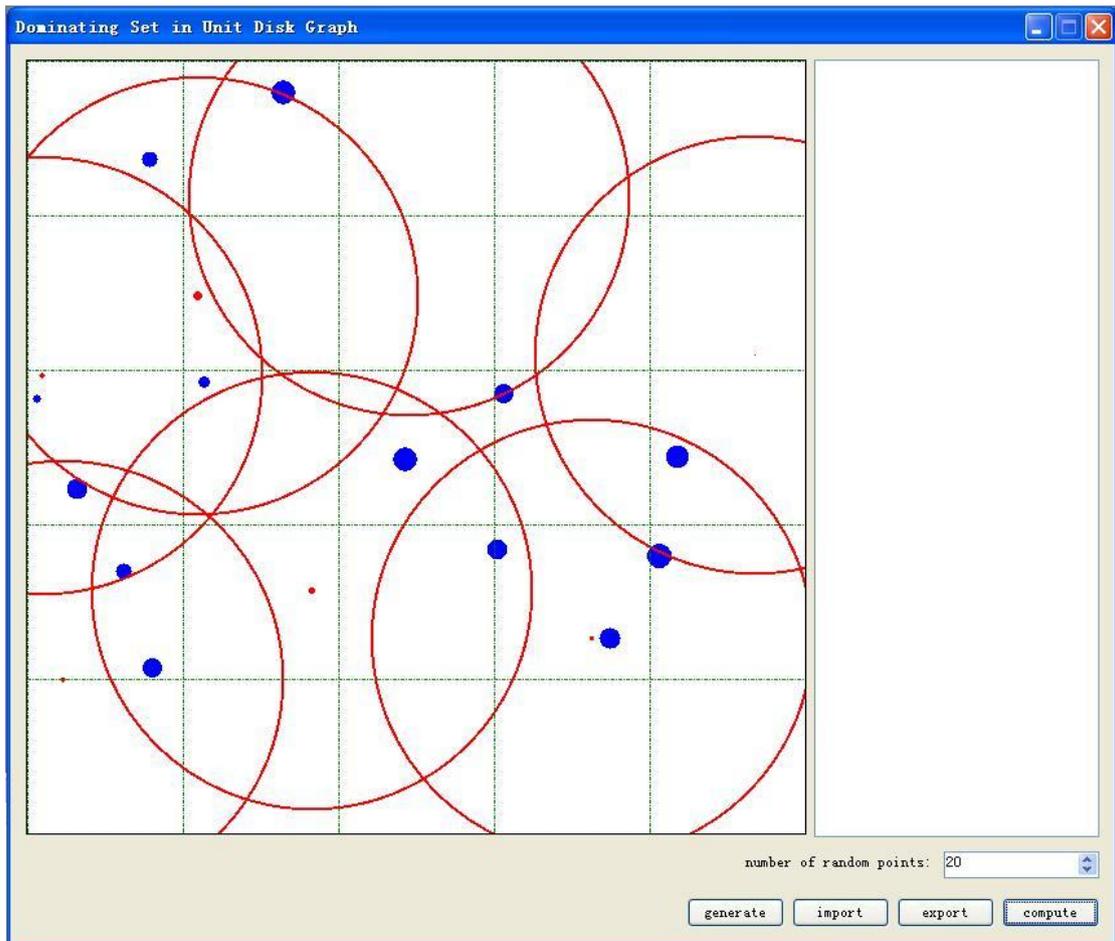
这时用户可以

- 1、调整随机产生样点的数目，然后点击 **generate** 按钮产生一组随机样点，程序将提示用户保存产生的样点；
- 2、点击 **import** 按钮导入过去保存的样点数据

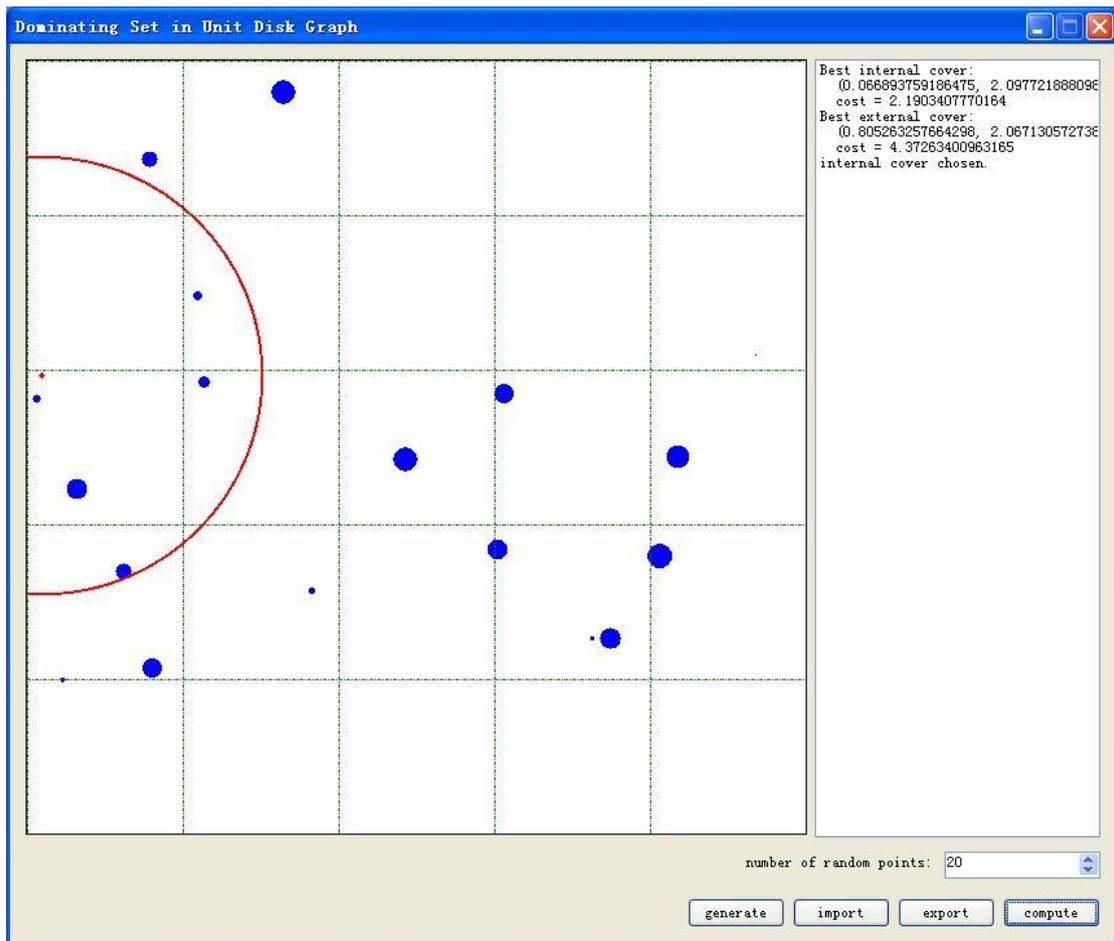
在有了一组样点数据之后，程序将会把样点显示在左上部分的绘图区域中，点的权值将用点的大小形象地表示，点的半径和它的权值成正比，如下图：



然后，用户可以点击 **compute** 按钮进行支配集的计算，计算结果将立刻显示出来，如下图：



这时用户可以点击某个方格察看该方格的支配集选择方案,并在右侧的文本框中看到详细的计算情况,如下图(察看程序选择的左起第一列上起第三格的支配集):



在察看完毕之后，用户可以点击 **export** 按钮保存计算出的（整体）支配集的方案。

## 参考文献：

1. Clark, B.N., Colbourn, C.J., Johnson, D.S.: *Unit Disk Graphs*. Discrete Mathematics, 86, 165–177 (1990)
2. Christoph Ambuhl, Thomas Erlebach, Matus Mihalak, Marc Nunkesser: *Constant-Factor Approximation for Minimum-Weight (Connected) Dominating Sets in Unit Disk Graphs*. 2006, In Proc. APPROX-RANDOM, pp. 3-14.
3. Decheng Dai, Changyuan Yu: *A  $5+\epsilon$  -Approximation Algorithm for Minimum Weighted Dominating Set in Unit Disk Graph*.
4. Yaochun Huang, Xiaofeng Gao, Zhao Zhang, Weili Wu: *A better constant-factor approximation for weighted dominating set in unit disk graph*. 1 March 2008, Journal of Combinatorial Optimization. Springer Netherlands. ISSN 1382-6905 (Print), 1573-2886 (Online)