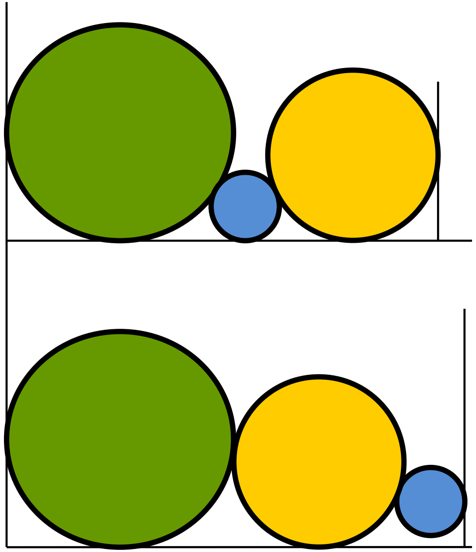
园排列问题

圆排列问题

给定n个圆的半径序列, 各圆与底线相切排列, 假定每个圆占大于1的长度, 求具有最小长度 l_n 的圆的排列顺序.



三个圆,半径给定

两种排列具有不同排列长度，第一种方式具有更小的排列长度

算法设计：分支限界

解：$<i_1,i_2,\ldots,i_n>$ 为 1,2,…,n 的排列

搜索空间为排列树

部分解向量 $<i_1,i_2,\ldots,i_k>$：表示前 k 个圆已排好. 令

$$B=\{i_1,i_2,\ldots,i_k\}$$

下一个园选择 i_{k+1}

约束条件：$i_{k+1}\in\{1, 2, \ldots, n\}-B$

界：当前得到的最小园排列长度

符号说明

k： 算法已经选择了第1— k 个圆

r_k： 第 k 个圆的半径

d_k： 第 k–1个圆到第 k个圆的圆心水平距离，$k>1$

x_k： 第 k 个圆的圆心坐标，规定 $x_1=0$,

l_k： 第 1— k 个圆的排列长度

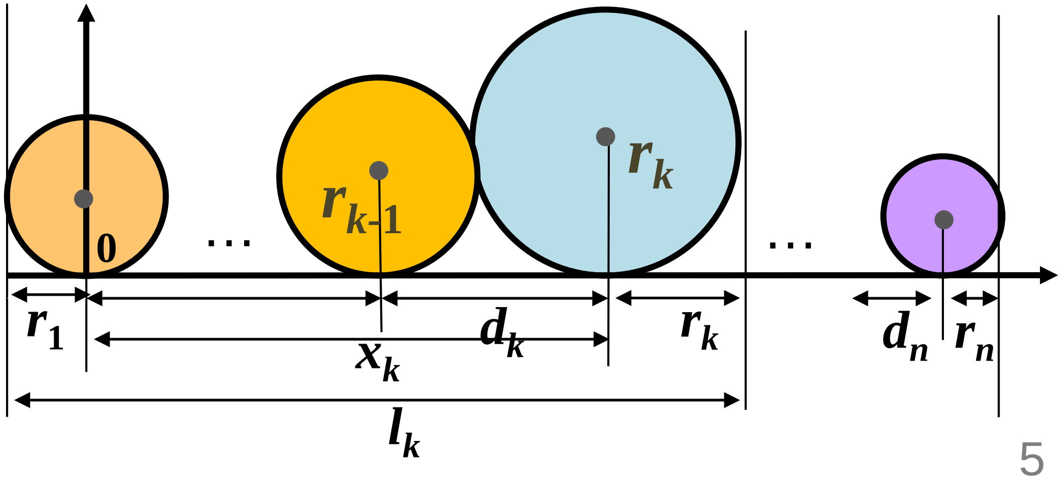
L_k： 放好 1—k 个圆以后，对应结点的代价函数值 $L_k \le l_n$

圆排列长度估计

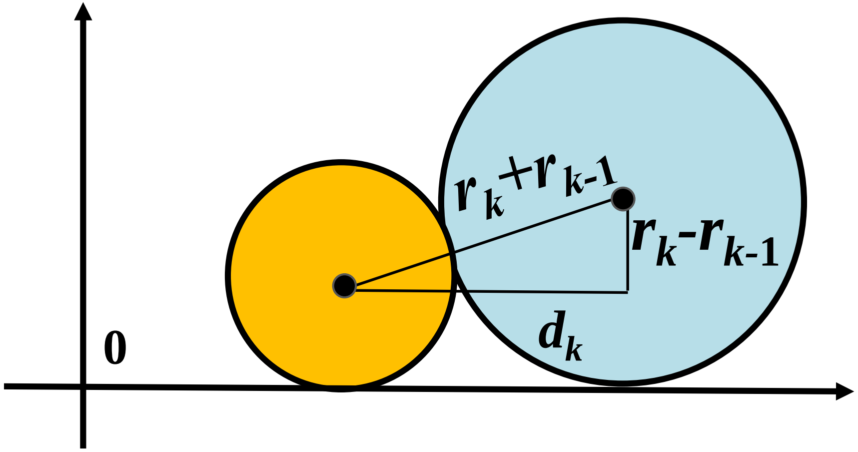
$x_k = x_{k-1} + d_k$

部分排列长度 $l_k = x_k + r_k + r_1$

排列长度 $\underline{l_n = x_k + d_{k+1} + d_{k+2} + \ldots + d_n + r_n} + r_1$



有关量的计算



$$d_k = \sqrt{(r_k + r_{k-1})^2 - (r_k - r_{k-1})^2}$$
$$= 2\sqrt{r_{k-1} r_k}$$

代价函数

$$d_k = 2\sqrt{r_{k-1} r_k}$$

排列长度

$$l_n = x_k + 2\sqrt{r_k r_{k+1}} + 2\sqrt{r_{k+1} r_{k+2}} + \ldots + 2\sqrt{r_{n-1} r_n} + r_n + r_1$$

$$\geq x_k + 2(n-k)r + r + r_1$$

下界

$$L_k = x_k + (2n-2k+1)r + r_1$$

$$r = \min(r_{i_j}, r_k)$$

$$i_j \in \{1, 2, \ldots, n\} - B$$

时间复杂度

- 搜索树树叶数为 $O(n!)$
- 每条路径代价函数的计算为 $O(n)$
- 最坏情况下算法时间复杂度为

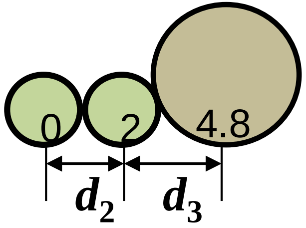
$$O(n\ n!) = O((n+1)!)$$

实例

输入： R={1,1,2,2,3,5}

排列 <1,2,3,4,5,6>

半径排列：1,1,2,2,3,5



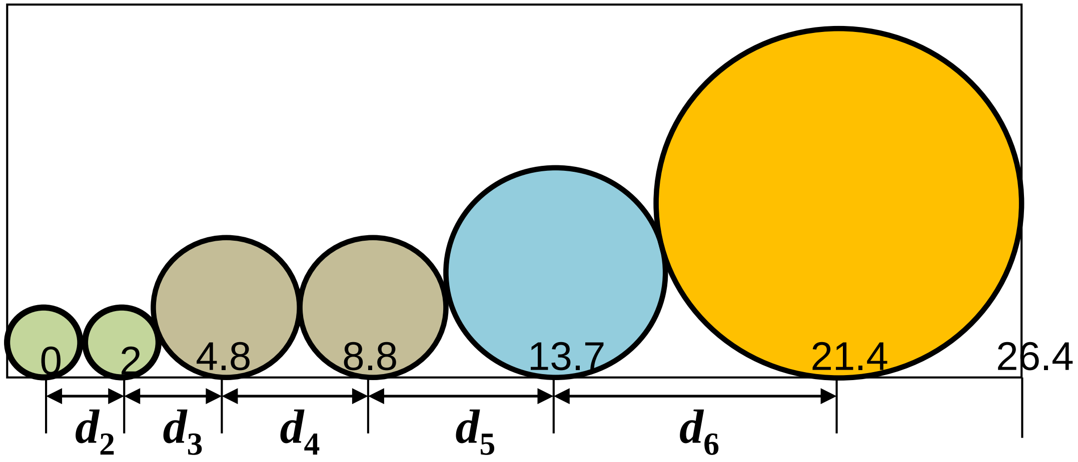
L_3=7.8+12=19.8

k	r_k	d_k	x_k	l_k	L_k
1	1	0	0	2	12
2	1	2	2	4	12
3	2	2.8	4.8	7.8	19.8
4	2	4	8.8	11.8	19.8
5	3	4.9	13.7	17.7	23.7
6	5	7.7	21.4	27.4	27.4

实例计算

半径排列：1,1,2,2,3,5，可行解l_6 =27.4

最优解：<1,3,5,6,4,2>, 最短长度26.5



小结

- 园排列问题的定义
- 如何设计代价函数？
  已有的排列长度+后续的圆排列
  长度的下界
- 时间复杂度估计：$O((n+1)!)$