近似算法

1 近似算法

所有已知的解决 NP-难问题算法都有指数型运行时间。但是，如果我们要找一个“好”解而非最优解，有时候多项式算法是存在的。

给定一个最小化问题和一个近似算法，我们按照如下方法评价算法：首先给出最优解的一个下界，然后把算法的运行结果与这个下界进行比较。对于最大化问题，先给出一个上界然后把算法的运行结果与这个上界比较。

1.1 最小顶点覆盖

先来回忆一下顶点覆盖的定义，它是一个与图中所有边相关联的顶点集。最小顶点覆盖问题是要找一个顶点数最少的顶点覆盖。

最小顶点覆盖的下界可以由最大匹配给出。因为匹配中任两边不相邻，所以匹配中的每条边至少有一个顶点在顶点覆盖中。

而且，注意到在最大匹配中所有匹配顶点的集合就是一个顶点覆盖。这是因为，任何一条两端点均未被匹配的边可以添加到匹配中，与匹配的最大性相矛盾。显然，这个算法包含的顶点数是我们的下界，最大匹配的边数，的两倍。因此，算法得到的值不会超过最优值的两倍。

我们感兴趣的两个问题是：相对于最优解，我们的下界到底有多“好”，而最后的解又有多“好”？

首先来说明下界可能是最优值的两倍。例如 n 条边的完全图，最大匹配有 $\frac{n}{2}$ 条边，所以我们的下界是 $\frac{n}{2}$。但是，需要 n-1 个顶点来覆盖这个图。因为任取一个 n-2 个顶点的集合，此图是完全图，在被删掉的两个顶点之间肯定存在一条边与选中的这 n-2 个顶点不关联。N 足够大时，我们有 $\frac{\mathrm{OPT}}{\mathrm{LB}} = \frac{n-1}{\frac{1}{2}n} \to 2$。因此，比较算法与这个界，不可能有比最优值的 2 倍更好的下界了。

接下来比较算法的最后结果与最优解。算法输出被最大匹配匹配的所有顶点。考虑每部分有 n 个顶点的完全二分图，这个图存在完美匹配，因此算法输出每一个顶点，即 2n 个顶点。但是最优顶点覆盖仅包含来自一边的 n 个顶点。可以看出，算法的下界是紧的。

1.2 旅行售货员问题

旅行售货员问题如下：给定一个完全图 $G = (V, E)$和一个定义在每条边上的距离函数

$d(i,j)$，找一个长度最小的哈密顿圈。注意，最小支撑树(MST)是最优解的一个下界。因为如果有一条途径比 MST 更短，那么在此途径中删掉一条边，就可以得到更小的支撑树。

树算法：我们可以从 MST 出发构造一个近似解。找到 MST，把每条边变成两条，这样每个顶点的度数为偶数，我们可以在这个新图中找到一个欧拉环游。删掉在欧拉环游中重复出现的顶点就可以得到一个哈密顿圈。由于距离 $d(i,j)$满足三角不等式，所以，哈密顿圈的长度不大于欧拉环游的长度。而欧拉环游的长度，恰为 MST 的两倍，所以树算法给出的解小于最优解的两倍。

那么在最坏的情况下，算法的近似比为 2。这个比值能否达到呢？同样地，我们先来比较一下下界和最优值。在有些情况下，比值 2 是可以达到的。例如，包含一条过 n 个顶点的路的图，令这条路上的边费用为 1，不在这条路上的边费用为到这条路的最短路距离。那么最小支撑树的长度为 $n-1$。而最小途径的长度则为 $2(n-1)$。可以看到下界是最优值的一半。

下面观察一下算法相对于最优解是如何运行的。同样地，算法有可能给出一个两倍于最优解的解。考虑一个阶梯形的图，它的某些边权值为 1，其它边根据最短路距离来赋权。注意到所有从一个梯级到下一个梯级的对角边的权值都为 2。所有的梯阶和阶梯的一边构成一个最小生成树，形状像一把梳子。算法执行的一个方式就是从梳子形状的最小生成树出发，在梯阶之间以锯齿状方式游走。当到达梯子终端后，再跳回到出发点。如果有 n 个梯阶的话，这次环游的长度就是 4n-2。然而，最短的环游是沿梯子的边界活动，而不经过所有的梯阶。这种情况下，环游的长度为 2n。

Christofide 算法：最优值的另一个可行界可由一个最小权重完美匹配给出。因为这是两点间最小距离的集合，且含有 $\frac{1}{2}n$ 条边（因为我们的图是完全图），最小权重完美匹配小于最优途径的一半。事实上，任何一个偶数边的子图上的最小权重完美匹配也小于最优途径的一半，再加上三角不等式，这就给出了另一个算法。

首先，找一个 MST，然后在奇度数的顶点上找最小权重完美匹配。既然在每个奇度数的顶点上增加一条边，那么所有的顶点都变为偶度数，我们就可以找到一个欧拉环游。缩短这个环游得到一个哈密顿圈，则这个圈的长度小于 MST 和最小权重完美匹配的长度之和。也就是，小于最优值的 $\frac{3}{2}$倍，即我们得到一个 $\frac{3}{2}$近似算法。这个简单的算法提供了现今已知的所有欧氏距离旅行售货员问题的最好保证。

1.3 集合覆盖

考虑元素为 $\{e_1, e_2, e_3, ..., e_n\}$的集合 S和其子集 $S_1, S_2, S_3, ..., S_m \subseteq S$。最小集合覆盖是指找到子集的一个最小集合，使其并集为 S。

这个问题可以用写为矩阵形式。用行表示子集 S_i，列表示元素 e_j，令 $M_{i,j}$ 等于 1 如果 $e_j \in S_i$ 否则为 0。那么问题就转化为找行的最小基数集合，使其覆盖所有列。

一个近似算法是贪心算法。在每一步，选择能覆盖最多未被覆盖元素的行。下面举例说明这个算法能给出的最好近似比为 $O(\log n)$。令 $S=\{e_1,e_2,...e_{2n}\}$，$S_1=\{e_1,e_2,...e_n\}$，$S_2=\{e_{n+1},e_{n+2},...e_{2n}\}$，那么 S_1 和 S_2 显然是一个集合覆盖。现在，令 S_3 包含 S_1 和 S_2 的前一半和一个其它元素，因此它比 S_1、S_2 覆盖的元素更多。令 S_4 包含 S_1 和 S_2 中接下来的四分之一个元素，那么，选择 S_3 后，它能覆盖的 S 中其余的元素要比 S_1、S_2 多。用这种方式继续来定义子集 S_i。贪心算法将选择 S_3，S_4，……，直到 $\log_2(\frac{n}{2})$ 个集合结束。而最优解仅包含两个子集。

为了方便算法分析，给每一个元素赋一个基于贪心算法所选集合的费用。令 S_k 为贪心算法的第 k 步选择的集合，而 S_k' 为 S_k 中未被前面的 $\{S_1,S_2,\ldots,S_{k-1}\}$ 覆盖的元素的集合。令每个元素 e_i 的费用为 $\frac{1}{|S_k'|}$，这里，S_k 是覆盖 e_i 的第一个集合。可得，每个元素费用的加和就是集合覆盖的基数。

注意每个元素的极大似然平均费用为 $\frac{\mathrm{OPT}}{n}$。另外，因为贪心算法首先选择费用最小的元素，所以，$\mathrm{cost}(e_1)\le\frac{\mathrm{OPT}}{n}$。OPT 也是剩余元素费用和的一个上界，因此，$\mathrm{cost}(e_k)\le\frac{\mathrm{OPT}}{n-k+1}$，所以

$$\sum_{k=1}^{n}\mathrm{cost}(e_k)\le\sum_{k=1}^{n}\frac{\mathrm{OPT}}{n-k+1}=\mathrm{OPT}\sum_{k=1}^{n}\frac{1}{n-k+1}$$

右边的加和有整数界，且可以推断出它位于 ln(*n*+*1*) 和 ln*n*+1 之间。所以，贪心算法的上界是最优值的 ln*n*+1 倍。由前面的例子可以看出这个界是紧的。

已经证明寻找好于 $O(\log n)$ 的近似集合覆盖算法本身是一个 NP-难问题。

2 松弛和四舍五入

一个比较通用的近似技巧如下：首先把这个问题改写为一个整数规划，然后放松约束条件得到一个线性规划。求解这个线性规划（可能会用到一个分离谕示），然后把小数解四舍五入得到一个整数解。

2.1 最小拥塞

给定一个图 $G=(V,E)$ 和一组顶点对 $(s_1,t_1),(s_2,t_2),\ldots,(s_k,t_k)$，找出每对顶点 s_i

和t_i之间的一条路。一条边上的拥塞是指通过这条边的路的条数。问题是找一组路使最大拥塞最小。

这个问题是 NP 完全问题。

为了把这个问题归约到一个整数规划，令变量$x^i_{j,k} \in \{0,1\}$，如果边(j,k)在从s_i到t_i的路上，则变量取 1，否则取 0。

为了保证可以生成路，我们建立一个流问题。我们需要一个从每个s_i到t_i的流值为 1 的流，那么每个顶点的散度为 0，即$\sum_k x^i_{k,j} = \sum_l x^i_{j,l}$对每个顶点 j 成立，除了$j = s_i$或$j = t_i$，这两种情况下散度为 1。

把目标函数$\max_{e \in E} \sum_i x^i_e$看作另外一个满足所有的拥塞都小于某个整数$c$的约束。那么就可以利用二进制搜索中的可行性问题找到最优值。那么约束条件为对所有的边e有$\sum_i x^i_e \leq c$。

整数规划的最优解和其松弛问题的最优解不同，存在整数差距。比如一个盒子图，s_1在左上角，s_2在右上角，t_1在右下角，t_2在左下角。那么整数规划的最优值拥塞为 2，但是，我们可以给所有的边分配$1/2$的流值，从而得到最大拥塞为 1。

第一个把线性规划结果转化成一系列路的方法是以概率x^i_e设x^i_e为 1，而以概率$1 - x^i_e$设其为 0。那么数学期望值为$E(x^i_e) = 1 \cdot x^i_e + 0 \cdot (1 - x^i_e) = x^i_e$，而期望拥塞是所有期望边权的和，即$\sum_i x^i_e$。但是，这可能不是问题的一个解，因为算法中可能取到不在s_i到t_i的路上的边。

注意，线性规划的解给出了一系列流，而非路。我们可以把流分解到路上：找一条从s_i到t_i的路，设其权重为路上边的最小容量λ^i_1，然后从流中删去$\lambda^i_1 p_1$。重复，得到λ^i_j。

还应注意到，对所有的 i，$\sum_j \lambda^i_j = 1$成立。因此，要把线性规划的结果转化成整数解，可以为每个 i 以概率λ^i_j选择一条路。那么，既然给定边上λ^i_j的加和为x^i_e，通过这条边的从s_i到t_i的路的期望值为x^i_e。因此，每条边的期望拥塞仍然是$\sum_i x^i_e$。注意，这是是给定边的期望拥塞，我们仍然不知道最大拥塞是多少。

对某个特殊边，设其整数（松弛了的）拥塞为X，期望拥塞为μ，可以证明X大于常数因子$c\mu$的概率小于$\frac{1}{n^2}$，这里n为顶点个数。那么，每条边的拥塞小于$c\mu$的概率至少为$1/2$，因为线性规划的期望拥塞小于整数规划的拥塞，即$c\mu \leq c\mathrm{OPT}$。

由马尔可夫不等式可知，如果X是一个非负随机变量，那么$P(X > c\mu) < 1/c$对$c > 0$成立，我们不妨取$c = n^2$. 但是，没有什么比我们已经做的选取随机路更好的方法。因此，对某条特殊边，令X^j来标志事件“通过这条边的路 j 在取整过程中被选中”，令$X = \sum_j X^j$为这条边的整数拥塞，令$\mu = E(X)$。那么，

$$P\left(X > (1+\delta)\mu\right) = P\left(e^{tX} > e^{t(1+\delta)\mu} \cdot \frac{E(e^{tX})}{E(e^{tX})}\right)$$

我们知道，

$$E(e^{tX}) = E(e^{t\sum X^i}) = \prod_i E(e^{tX^i}) = \prod_i (p_i e^t + 1 - p_i) = \prod_i (1 + p_i(e^t - 1))$$

因为$1 + x \leq e^x$对任意实数x成立，且$\sum p_i = \mu$，所以

$$E(e^{tX}) \leq \prod_i e^{p_i(e^t-1)} = e^{e^t-1} \prod e^{p_i} = e^{\mu(e^t-1)}$$

由马尔可夫不等式可得

$$P\left(e^{tX} > e^{t(1+\delta)\mu} \cdot \frac{E(e^{tX})}{E(e^{tX})}\right) \leq \frac{E(e^{tX})}{e^{t(1+\delta)\mu}}$$

代入，得

$$\frac{E(e^{tX})}{e^{t(1+\delta)\mu}} \leq \frac{e^{\mu(e^t-1)}}{e^{t(1+\delta)\mu}}$$

令$t = \ln(1+\delta)$，对$1 + \delta \geq 2e$，我们有

$$P\left(X > (1+\delta)\mu\right) \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu \leq \left(\frac{e^{1+\delta}}{(2e)^{1+\delta}}\right)^\mu = \frac{1}{2^{(1+\delta)\mu}}$$

取δ满足$(1+\delta)\mu = \max(2e\mu, 2\log n)$. 那么

$$P(X < \max(2e\mu, 2\log n)) \leq \frac{1}{2^{2\log n}} = \frac{1}{n^2}$$

因此，没有一条边的拥塞大于$\max(2e\mu, 2\log n)$的概率为$\frac{1}{2}$。从而，重复取整过程，可

得近似比为$2e\mathrm{OPT}+2\log n$。那么最坏的情况下近似比为 $O(\log n)$ 。