近似算法

当我们遇到一个 NP 完全问题怎么办?

- 或者，多项式时间的解决方案非常慢呢?
- 假定输入是随机的，算法能达到“期望的性能”例如：所有图中的哈密尔顿路径。问题：假定均匀分布。
- 运行一个非多项式时间的算法（希望程度比较轻）如例如分支界限法。通常不能求证运行时间的界限，但有时也能。
- 用启发式搜索算法来解决，但该算法被证明并非十分有效。

定义：

- 最优化问题，问题实例 I，解决方案 $S(I)$，映射函数 $f: S(I) \rightarrow R$
- 最大值化/最小值化：在 $S(I)$ 中找到解决方案使 f 最大或最小
- 叫做该问题的最优解，记为 $OPT(I)$
- 例如：装箱问题。问题实例是一个集合 S，元素 $S_i \in 0,1$，将 S 没有一个子集中元素超过 1

通常做如下技术假定：

- 假设：所有的输入和函数 f 的变化范围为整数/有理数（不能代表实数，但允许，如线性规划，二叉查找）
- 假定 $f(\sigma)$ 是一个多项式大小（比特数）的数（否则输出时间会很长）
- 找到在比特复杂度下多项式时间的解

NP 难度：

- 一个问题为最优化 NP 难问题如果它可由判定性 NP 难问题转化而来
- （例如,像“当问题实例≤ k 时是否为最优解？”这样的问题）
- 更多用到还原性图灵机（Turing-reducibility）（GJ）的概念

近似算法：

- 任何给出一个可行解的算法
- 例如，每个物品都在自己的箱子
- 当然，希望一个好的算法。怎么衡量?

绝对近似算法

定义：对于任何的输入 I，有 $|A(I) - OPT(I)| \leq k$，称为 k-绝对近似。

EG：平面图染色问题

- 判断一个图是否是 3-colorable 是个 NP 难问题
- 已知 4 种可以
- 5 种更容易
- 重复对每条边着色：根据 Vizing’s 理论最优解为 Δ 或（有建设性的）$\Delta + 1$

我们所知的最优解有常数边界的很少。
通常，我们通过把问题“缩减”的方法来证明该问题没有绝对近似算法。

- 例如 背包问题
  - 规定价值为 p_i，大小为 s_i，背包大小为 B
  - 假定 p_i 为整数
  - 假定有 k-绝对近似
  - 把所有 p_i 乘以 $k+1$，求解，再还原
- 例如 独立子集（团）
  - G 的 $k+1$ 个副本

相对近似算法

定义：

- 一个 α-优化解决方案，在算法求最小值时最多为最优解 α 的倍，在算法求最大值时至少为最优解的 $1/\alpha$ 倍
- 如果一个近似算法对任何的输入都能产生 α-近似的可行解，我们就称该算法的近似比为 α
- 称为 α-近似算法

怎样证明一个算法是相对近似的？

- 无法描述最优解，所以也无法比较
- 不过，可以同能计算出来的下界进行比较

贪心算法

每一步的做法显而易见

- 难点在于证明每一步是有效的
- 通常，密切关注正确的上界和下界

最大割集

- 容易找到最到上界
- 最大分割的贪心算法

最大直径聚类

- Gonzales’s 算法
- 到当前中心的距离不断下降
- 假定 k 次选择之后，最远的距离为 d
- 那么 OPT $\geq d$
  - $k+1$ 个相互距离都为 d 的点
  - 必定有某些点在一个聚类中
- 现在把每个点分给距其最近的中心的那个分组中
- 距中心的最大距离（半径）为 d
- 所以最大直径为 $2d$
- 2-近似

集覆盖

- n 个物品

- OPT = k
- 每一步，仍可以用 k 个集合覆盖剩余的物品
- 所以就覆盖了 $1/k$ 个剩余物品

顶点覆盖：

- 定义问题
- 假定重复挑选没有覆盖的边然后覆盖：没有近似比
- 假定挑选没有覆盖的边然后覆盖其两个顶点：2-近似
- 有时，需要用到另外一种更好的贪心策略
- 明确下界是怎么得来的——下界的求法决定了算法

Graham’s 规则对多机调度问题是近似比为 $2-\frac{1}{m}$ 的近似算法

- 阐述问题：m 台机器，n 个工作每个工作的处理时间为 p_j，最小的处理时间
- 也可以考虑成为最小化最大装载的连续运行时间
- 运用贪心算法解决
- 通过与下界的比值来证明
- 第一个下界：平均装载量：OPT $\geq \frac{1}{m}\sum p_j$
- 第二个下界：OPT ≥max p_j
- 假定 M_1 最后有最大的运行时间 L
- 假定 j 是最后一个加到 M_1 上的工作
- 那么 j 记载之前，M_1 上已经装载的工作运行时间为 $L-p_j$，是当前最小的
- 所以 $\sum p_j \geq m(L-p_j)+p_j$
- 所以 $OPT \geq L+(1-\frac{1}{m})p_j$
- 所以 $L \leq OPT+(1-\frac{1}{m})p_j \leq (2-\frac{1}{m})OPT$

注意：

-
- 该算法是在线算法，近似比为 $2-\frac{1}{m}$
- 我们并不知道最优的调度方式，只是用到了其下界
- 我们采用的贪心策略
- 边界情况：考虑 $m(m-1)$ 个大小为 1 的工作，1 个大小为 m 的工作
- 问题在于：如果最后一个工作运行时间很长
- LPT 可以达到 4/3，但为离线算法
- 新的在线算法可以达到 1.8 的近似比

近似方案

目前，我们已经见过各种近似因子为常数的近似算法。

- 能达到的最好的常数因子是多少？
- 下界：APX-hardness/Max-SNP

一个近似方案是算法 A_ε 的集合,该类算法满足：

- 每个算法有多项式的运行时间
- A_ε 近似比能达到 $1+\varepsilon$

但注意：运行时间可能随着 ε 的变化变的非常糟糕

FPAS，伪多项式算法

背包问题

- 对价值的边界的确定用动态规划法
- $B(j,s)$ = 工作 1，2，…，j 总大小 $\le s$ 的最佳子集
- 近似
  - 假设最优解为 P
  - 把权值扩大为 $\lfloor (n/\varepsilon P) p_i \rfloor$
  - 新的最优解至少为 $n/\varepsilon - n = (1-\varepsilon)n/\varepsilon$
  - 所以找到的解决方案在原始最优解的 $1-\varepsilon$ 的范围内
  - 表的大小是多项式
- 这个能证明 P=NP？不能
- 回到伪多项式算法

伪多项式可提供 FPAS，相反的情况基本上也都是真实的

- 背包问题是个弱 NP 难问题
- 强 NP 难（定义）问题意味着没有伪多项式

枚举

更有效的想法：k-枚举

- 回到多机调度问题
- 最优地调度 k 个最大的工作
- 对剩余工作的调度采用贪心算法
- 分析：注意 $A(I) \le OPT(I) + p_{k+1}$
  - 考虑一个工作的最大完成时间为 c_j
  - 如果该工作是 k 中的一个，那么算法以求得最优解结束
  - 否则，被分配到最小装载的那台机器上，所以 $c_j \le OPT + p_j \le OPT + p_{k+1}$

- 如果 p_{k+1} 最小，算法结束
- 注意 $OPT(I) \geq (k/m) p_{k+1}$
- 推出 $A(I) \leq (1+m/k) OPT(I)$
- 所以，对确定的 m，可以得到任意想要的近似比

调度任意台机器

- 把枚举和近似法相结合
- 假定只有 k 中工作
  - 对于一给定的机器其上每种类型的工作个数的向量表示—叫做“机器类型”
  - 只有 n^k 个不同的向量/机器类型
  - 所以需要找到每种机器类型有多少个
  - 用动态规划法：
    * 枚举所有可以在 j 台机器上 T 时间内完成的工作类型
    * 如果是 $j-1$ 种机器类型和 1 种机器类型的和就在集合中
  - 因为只有多项式数量的工作类型，所以该算法对其有效
- 用近似来确定少数重要的工作类型
  - 在 ε 条件下猜测最优解 T （用二叉查找）
  - 所有完成时间超过 εT 的工作都是“大”工作
  - 对其中的每一个完成时间近似到 $(1+\varepsilon)$ 的下一个幂次
  - 只有 $O(1/\varepsilon \ln 1/\varepsilon)$ 种大类型
  - 最优解决
  - 对剩余工作的调度采用贪心算法
    * 如果最后一个工作很大，能在问题被近似为最优解的 ε 范围内求得其最优解
    * 如果最后一个工作很小，贪心算法的分析证明了解在最优解的 ε 范围内

松弛算法

旅行商问题

- 需要遍历：没有近似算法（找到哈密尔顿回路是个 NP 难问题）
- 无向图：MST（最小生成树）松弛算法近似比为 2， Christonfides’ heuristic
- 有向图：环路覆盖松弛算法

LP 松弛算法

三步

- 写整型线性程序
- 松弛
- 近似

顶点覆盖

MAX SAT

定义

- 变量
- 子句
- NP-完全问题

随机设置

- 达到$1-2^{-k}$
- 对值大的k很有效，但当$k=1$是只有 1/2

LP

$$\text{Max} \quad \sum z_j$$

$$\sum_{i \in C_j^+} y_i + \sum_{i \in C_j^-} (1-y_1) \geq z_j$$

分析

- $\beta_k = 1-(1-1/k)^k$. 值 1，3/4，.704，…
- 随机近似y_i
- 引理：k-变量的子句的合取范式 w/pr 至少为$\beta_k \hat{z}_j$
- 证明：
  - 假定所有变元为正
  - 探测$1-\prod(1-y_i)$
  - 当所有的$y_i = \hat{z}_j/k$ 时为最大值 。
  - 导出$1-(1-\hat{z}/k)^k \geq \beta_k \hat{z}_k$
  - 在$z=0,1$时检查
- 结果：近似比$(1-1/e)$ (收敛为$(1-1/k)^k$
- 对于小k的情况好的多：即对$k=1$时近似比为 1

小子句线性规划法较好，大子句随机法较好

- 更好：两种方法都尝试
- 在两种方法中都有用到n_1, n_2
- 导出$(n_1+n_2)/2 \geq (3/4)\sum \hat{z}_j$
- $n_1 \geq \sum_{C_j \in S^k} (1-2^{-k})\hat{z}_j$
- $n_2 \geq \sum \beta_k \hat{z}_j$
- $n_1 + n_2 \geq \sum (1-2^{-k}+\beta_k)\hat{z}_j \geq \sum \frac{3}{2}\hat{z}_j$

0.1 切尔诺夫边界近似

集覆盖

定理：

- 假设 X_i 个待检测位置（即 0/1 独立），$E[\sum X_i]=\mu$

$$Pr[X>(1+\varepsilon)\mu]<[\frac{e^{\varepsilon}}{(1+\varepsilon)^{(1+\varepsilon)}}]^{\mu}$$

- 注意 n 个值相互独立，指数为 μ

证明

- 对于任何 $t>0$，

$$Pr[X>(1+\varepsilon)\mu]=\Pr[\exp(tX)>\exp(t(1+\varepsilon)\mu)]$$

$$<\frac{E[\exp(tX)]}{\exp(t(1+\varepsilon)\mu)}$$

- 考虑到其独立性，

$$E(\exp(tX)]=\prod E(\exp(tX_i)]$$

$$E(\exp(tX_i)]=p_ie^t+(1-p_i)$$

$$=1+p_i(e^t-1)$$

$$\leq\exp(p_i(e^t-1))$$

$$\prod\exp(p_i(e^t-1))=\exp(\mu(e^t-1))$$

- 所以总的边界为

$$\frac{\exp(\mu(e^t-1))}{\exp(t(1+\varepsilon)\mu)}$$

对任何 t 都成立的，为达到最小值，另 $t=\ln(1+\varepsilon)$

- 更简单的边界：

－ 对 $\varepsilon<1$，小于 $e^{-\mu\varepsilon^2/3}$

－ 对 $\varepsilon<2e-1$，小于 $e^{-\mu\varepsilon^2/4}$

－ 对较大 ε，小于 $2^{-(1+\varepsilon)\mu}$

- 在 $\exp(-tX)$ 上做同样推导

$$\Pr[X < (1-\varepsilon)\mu] < [\frac{e^{-\varepsilon}}{(1-\varepsilon)^{(1-\varepsilon)}}]^{\mu}$$

以 $e^{-\varepsilon^2/2}$ 为界

基本应用：

- 在 c 个箱子中放 $cn\log n$ 个球
- 最大均值匹配
- a fortiori for n balss in n bins

大致观察如下：

- 当 $\varepsilon \approx 1/\sqrt{\mu}$ 时边界减小，即绝对误差为 $\sqrt{\mu}$
- 不要奇怪，因为标准误差在 μ 左右
- 如果 $\mu = \Omega(\log n)$，可能随着常数 ε 的变化，误差的变化为 $O(1/n)$，对于多项式的事件数来说是很有效的。
- 注意与高斯分布的相似性
- 总结：可运用于任何在[0,1]范围内变化的分布中

诸多 Chernoff 应用

- 多种商品流松弛
- Chernoff 边界
- 联合边界