高级算法设计与分析 Lecture 5

授课时间: 2020 年 4 月 12 日 授课教师: 孙晓明 记录人: 吴昊

1 代数化方法

代数化方法 (Algebraic Techniques) 是指将问题用代数的形式表示，将其转化为一个代数问题，从而可以借助代数中的定理与性质。前面课程中提到过“矩阵乘法验证”问题和“判定两个文件是否相同”问题都是代数化方法实际运用的例子，下面我们通过图论中的“完美匹配”问题进一步理解这一方法。

1.1 完美匹配问题

无向图 $G(V, E)$ 的一个匹配（matching）是边集 E 的一个子集，并要求该子集中的边没有公共顶点。如果一个匹配覆盖了顶点集 V，则称它是一个完美匹配（perfect matching）。完美匹配问题是要判断一张图中是否存在完美匹配。

在我们下面的讨论中，我们只讨论二部图的完美匹配问题。如果无向图 $G = (V, E)$ 的顶点集 V 可分割为两个互不相交的子集 A, B, 并且图中的每条边的两个顶点分别属于 A 和 B, 则称 G 是一个二部图，并记为 $G(A, B, E)$。注意到如果 A 与 B 集合的大小不同，那么显然不存在完美匹配，所以下面假设 A 与 B 包含的顶点个数相同，即 $|A| = |B|$。

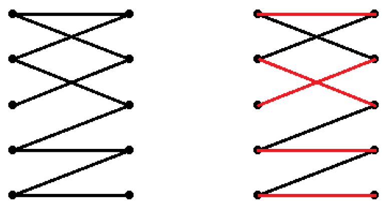

图 1: 一张二部图和它的一个完美匹配（红色边）

1.2 代数化过程

以上面的图 1为例，我们先写出该二部图的邻接矩阵 M，它可以用一个 5×5 的矩阵表达，如果第 i 行第 j 列是 1，说明从左边的第 i 个点到右边的第 j 个点有一条边[1]。

$$M = \begin{bmatrix} {\color{red}1} & 1 & 0 & 0 & 0 \\ 1 & 0 & {\color{red}1} & 0 & 0 \\ 0 & {\color{red}1} & 0 & 0 & 0 \\ 0 & 0 & 1 & {\color{red}1} & 0 \\ 0 & 0 & 0 & 1 & {\color{red}1} \end{bmatrix}.$$

如果我们把该图的完美匹配中的边在邻接矩阵中用红色标出，不难发现，这些边分别处于矩阵的不同行、不同列。这一性质很容易推广到一般的情况：对于一个二部图 $G(A, B, E)$，且 $|A| = |B| = n$,

[1]需要注意，此处二部图的邻接矩阵与一般图的邻接矩阵定义有区别

那么 G 存在一个完美匹配当且仅当其 $n\times n$ 的邻接矩阵中存在 n 个 “1”，它们分别处于不同行、不同列。而 “不同行、不同列” 这样的性质，很容易让我们联想到方阵行列式的计算公式：

$$\det(M)=\sum_{\sigma\in S_n}(-1)^{\mathbf{sgn}(\sigma)}\prod_{i=1}^{n}M_{i,\sigma(i)}.$$

观察上式，如果忽略掉求和式中的 $(-1)^{\mathbf{sgn}(\sigma)}$，那么求和式的每一项就对应于一个可能的完美匹配：如果 $\prod_{i=1}^{n}M_{i,\sigma(i)}$ 为 1，则由排列 σ 可以得到一个完美匹配。但由于前面的符号项，两个完美匹配对应 1 可能正负抵消。我们可以得到这样一个结论：如果 $\det(M)\neq 0$，则存在完美匹配；如果 $\det(M)=0$，则无法确定是否存在完美匹配。

事实上，之所以在计算 $\det(M)$ 时会出现正负抵消的问题，是因为 M 中的元素只有 0 和 1。如果我们不用 1 表示存在一条边，而是引入一个新的变量，那么 $\det(M)$ 就会变成一个多元多项式，因为变元不同，所以可以避免正负抵消。具体来看，在上面的例子中，将 M 中的非零元素替换成不同的变元，就得到下面的矩阵 M'：

$$M'(x_{11},\ldots,x_{55})=\begin{bmatrix} x_{11} & x_{12} & 0 & 0 & 0\\ x_{21} & 0 & x_{23} & 0 & 0\\ 0 & x_{32} & 0 & 0 & 0\\ 0 & 0 & x_{43} & x_{44} & 0\\ 0 & 0 & 0 & x_{54} & x_{55}\end{bmatrix}.$$

我们可以得到下面的定理：

定理 1. 二部图 G 存在完美匹配的**充分必要**条件是 $\det(M')\neq 0$。

证明 由行列式的定义，

$$\det(M')=\sum_{\sigma\in S_n}(-1)^{\mathbf{sgn}(\sigma)}\prod_{i=1}^{n}x_{i,\sigma(i)}. \tag{1.1}$$

上式对置换群 S_n 中的所有置换 σ 求和，其中每一项都是带符号的 M' 中不同行不同列的 n 个元素的乘积单项式；根据上述的观察，这一项不为 0 当且仅当 G 中存在这 n 个元素对应的完美匹配。所以，G 中存在完美匹配的充分必要条件是 M' 的行列式不恒为 0，即 $\det(M')\neq 0$。 □

而要判断多元多项式根与系数的关系，需要使用 Schwartz-Zippel 引理。

1.3 Schwartz-Zippel Lemma

定义 2 (多元多项式的次数)**.** 多元多项式 P 中非零单项式的最高次数称为多元多项式 P 的次数，记为 $\deg(P)$。

例 1 令 $P(x_1,x_2,x_3,x_4)=x_1^7x_2^3x_3+x_1x_2^8x_3^6$，则最高次单项式为 $x_1x_2^8x_3^6$，故 $\deg(P)=15$。

定理 3 (Schwartz-Zippel Lemma)**.** 设 $P(z_1,\ldots,z_n)$ 为域 $\mathcal{F}$ 上的非零多项式，即

$$P(z_1,\ldots,z_n)\in\mathcal{F}[z_1,\ldots,z_n]，且 P\neq 0$$

S 为域 $\mathcal{F}$ 的有限子集，即 $S\subseteq\mathcal{F}$，且 $|S|<+\infty$。若独立随机地从 S 中选取 $r_1,\ldots,r_n$，则有

$$\Pr(P(r_1,\ldots,r_n)=0)\leq\frac{\deg(P)}{|S|}$$

证明 对多项式 $P(z_1,\ldots,z_n)$ 的变量个数 n 进行归纳:
当 $n=1$ 时，由代数学基本定理，$P(z_1)=0$ 在 S 上至多有 $\deg(P)$ 个根。因此

$$\Pr(P(r_1)=0)\le\frac{\deg(P)}{|S|}$$

假设原命题对变量个数小于等于 n 的多项式都成立,现考虑变量个数为 $n+1$ 的多项式 $P(z_1,\ldots,z_{n+1})$。

$P(z_1,\ldots,z_{n+1})$ 可以写成关于变元 z_1 的多项式形式, 每一项的系数都是一个 n 元多项式, 假设 z_1 最大次数为 d，则有

$$\begin{aligned}P(z_1,z_2,\ldots,z_{n+1})=&z_1^d\cdot P_d(z_2,\ldots,z_{n+1})+z_1^{d-1}\cdot P_{d-1}(z_2,\ldots,z_{n+1})+\\&\cdots+z_1\cdot P_1(z_2,\ldots,z_{n+1})+P_0(z_2,\ldots,z_{n+1})\end{aligned}\tag{1.2}$$

当 $\Pr(P(r_1,r_2,\ldots,r_{n+1})=0)$ 时，我们对 $P_d(r_2,\ldots,r_{n+1})$ 是否为 0 分类讨论，得到下面的概率表达式

$$\begin{aligned}\Pr(P(r_1,r_2,\ldots,r_{n+1})=0)&=\Pr\Big(P(r_1,r_2,\ldots,r_{n+1})=0,P_d(r_2,\ldots,r_{n+1})=0\Big)\\&\quad+\Pr\Big(P(r_1,r_2,\ldots,r_{n+1})=0,P_d(r_2,\ldots,r_{n+1})\ne0\Big)\\&\le\Pr\Big(P_d(r_2,\ldots,r_{n+1})=0\Big)\\&\quad+\Pr\Big(P(r_1,r_2,\ldots,r_{n+1})=0,P_d(r_2,\ldots,r_{n+1})\ne0\Big)\end{aligned}\tag{1.3}$$

由归纳假设

$$\Pr(P_d(r_2,\ldots,r_{n+1})=0)\le\frac{\deg(P_d)}{|S|}=\frac{\deg(P)-d}{|S|}\tag{1.4}$$

又由式 (1.2), 当 $z_2,\ldots,z_{n+1}$ 的取值给定时, 即 $(z_2,\ldots,z_{n+1})=(b_2,\ldots,b_{n+1})$, 此时 $P(z_1,z_2,\ldots,z_{n+1})$ 可以视为关于变量 z_1 的一元 d 次多项式，记为 $Q_{b_2,\ldots,b_{n+1}}(z_1)$。则有

$$\begin{aligned}&\Pr\Big(P(r_1,r_2,\ldots,r_{n+1})=0,P_d(r_2,\ldots,r_{n+1})\ne0\Big)\\\le&\sum_{b_2,\ldots,b_{n+1}}\Pr(r_2=b_2,\ldots,r_{n+1}=b_{n+1})\cdot\Pr(Q_{b_2,\ldots,b_{n+1}}(r_1)=0)\\\le&\sum_{b_2,\ldots,b_{n+1}}\Pr(r_2=b_2,\ldots,r_{n+1}=b_{n+1})\cdot\frac{\deg(Q_{b_2,\ldots,b_{n+1}})}{|S|}\\\le&\frac{d}{|S|}\end{aligned}\tag{1.5}$$

由式 (1.3)，式 (1.4) 和式 (1.5) 可得，

$$\begin{aligned}\Pr(P(r_1,r_2,\ldots,r_{n+1})=0)&\le\frac{\deg(P)-d}{|S|}+\frac{d}{|S|}\\&=\frac{\deg(P)}{|S|}\end{aligned}$$

□

1.4 随机算法

给定一个二部图 $G=(A,B,E)$，其中 $|A|=|B|=n$，我们利用 1.1中的分析以及 Schwartz-Zippel 引理给出一个随机算法判定 G 中是否有完美匹配，具体描述如下：

1. 构造图 G 的代数化邻接矩阵 M'
2. 在 n^2 到 $2n^2$ 之间找一个素数 p，在有限域 $\mathcal{F}_p$ 上独立均匀随机地选取 $r_{11},\ldots,r_{nn}\in\mathcal{F}_p$，计算 $\det(M'(r_{11},\ldots,r_{nn}))$
3. 如果结果不为 0，则输出 “存在”；如果结果为 0，则输出 “不存在”。

注：由 *Bertrand–Chebyshev* 定理，对于任意大于 1 的整数 m，一定存在一个素数 p 满足 $m<p<2m$，因此一定可以找到这样的素数

由定理 1，如果图 G 不存在完美匹配，则算法一定输出 “不存在”；但当 G 中存在完美匹配时，虽然多项式 $\det(M')$ 不恒为 0，但由于 $r_{11},\ldots,r_{nn}$ 是随机选取的，它有可能恰好是这个多项式的根，从而算法有可能出错。下面分析算法出错的概率。

定理 4. *当 G 存在完美匹配时，$\Pr_{r_{11},\ldots,r_{nn}}(\det(r_{11}\ldots r_{nn})=0)\leq\frac{1}{n}$。*

证明　当 G 存在完美匹配时，多项式 $P=\det(M')$ 不恒为 0。令 $S=\mathcal{F}_p$，则 S 为有限集，由引理 3，对于独立均匀随机选取的 $r_{11},\ldots,r_{nn}$ 有

$$\Pr_{r_{11},\ldots,r_{nn}}(P(r_{11}\ldots r_{nn})=0)\leq\frac{\deg(P)}{|S|}=\frac{n}{p}\leq\frac{1}{n}. \tag{1.6}$$

□

算法的执行过程依赖于两个子过程：在 n^2 和 $2n^2$ 之间找一个素数，以及求一个数值矩阵的行列式。使用枚举法即可找到此素数，且素数判定存在多项式时间算法；求矩阵的行列式也存在多项式时间算法。所以整个算法是多项式时间的，从而是一个RP算法。

上述寻找素数的过程也可以进行一些改进：在 n^2 和 $2n^2$ 之间均匀随机选择一个整数 k，用素数判定算法判断 k 是否为素数；反复执行这个过程直到找到素数为止。事实上，如果设 $\pi(m)=\{p\leq m: p\text{是素数}\}$ 为不超过正整数 m 的素数的个数，则根据素数定理有

$$\pi(m)\sim\Theta(\frac{m}{\ln m}). \tag{1.7}$$

从而可以知道，在 n^2 和 $2n^2$ 之间有 $\Theta(\frac{1}{\ln n})$ 个素数。类比掷硬币的模型，在期望意义下，平均做 $\Theta(\ln n)$ 次这样的均匀随机选择就能得到一个素数。

Homework:（红-蓝匹配问题）为二部图的红蓝匹配问题设计一个多项式时间的随机算法。问题描述：二部图 G 的边被染成了红和蓝两种颜色，给定 $1\leq k\leq n$，判定 G 中是否存在一个完美匹配 M，使得 M 中恰有 k 条红边和 $n-k$ 条蓝边。

2 期望方法

2.1 最大割问题

定义 5 (割，割集)**.** *给定图 $G=(V,E)$，定义图 G 上的**割**为顶点集 A 和 B 组成的二元组 (A,B)，其中 $A\cap B=\varnothing$，$A\cup B=V$。由割 (A,B) 可定义**割集**$E(A,B)=\{(u,v)\in E|u\in A,v\in B\}$，*

最大割问题 (Max-Cut Problem)：给定图 $G=(V,E)$，求使得割集 $E(A,B)$ 元素个数最多的割 (A,B)。

最大割的判定问题：给定图 $G=(V,E)$ 以及正整数 k, 判定是否存在元素个数恰好为 k 的割集 $E(A,B)$。

最大割的判定问题为 NP-完全问题。最大割问题为 NP-难的问题，目前还没有已知的确定性多项式时间算法。下面介绍一个关于最大割问题的近似算法。

算法描述 (求解最大割问题的近似算法). 输入图 $G=(V,E)$，对于 V 中的任一顶点 v，以 $\frac{1}{2}$ 的概率将 v 放入顶点集 A 中，以 $\frac{1}{2}$ 的概率将 v 放入顶点集 B 中。

定理 6. 给定图 $G=(V,E)$，设随机变量 $E(A,B)$ 为上述算法输出的割集，设 opt 为图 G 的最大割集的大小。则有

$$\frac{\mathbb{E}(|E(A,B)|)}{opt} \geq \frac{1}{2}$$

证明 定义随机变量 X_i，$i=1,\dots,|V|$，

$$X_i = \begin{cases} 0, & \text{顶点} u_i \in A \\ 1, & \text{顶点} u_i \in B \end{cases}$$

则有 $\Pr(X_i=0)=\Pr(X_i=1)=\frac{1}{2}$。

对于边集 E 中的任一条边 (u_i,u_j)，有

$$\Pr((u_i,u_j)\in E(A,B)) = \Pr(X_i \neq X_j) = \frac{1}{2}$$

对于任意的 $(u_i,u_j)\in E$，定义随机变量 $Y_{i,j}$ 如下，

$$Y_{i,j} = \begin{cases} 0, & (u_i,u_j) \notin E(A,B) \\ 1, & (u_i,u_j) \in E(A,B) \end{cases}$$

则有 $\mathbb{E}(Y_{i,j})=\frac{1}{2}$

$$\begin{aligned} \mathbb{E}(|E(A,B)|) &= \mathbb{E}\left[\sum_{(u_i,u_j)\in E} Y_{i,j}\right] \\ &= \sum_{(u_i,u_j)\in E} \mathbb{E}[Y_{i,j}] \\ &= \frac{|E|}{2} \end{aligned}$$

又 $opt \leq |E|$，因此

$$\frac{\mathbb{E}(|E(A,B)|)}{opt} \geq \frac{1}{2}$$

□

2.2 期望方法去随机化

对于最大割问题，我们还可以通过去随机化，将随机算法转换成确定型算法，并且该算法的表现不劣于随机算法。我们已知随机算法中 $\mathbb{E}(|E(A,B)|)=\frac{|E|}{2}$。记 $|E(A,B)|=Y$，依据期望的全概率分解公式，有

$$\begin{aligned}\mathbb{E}(Y)&=\Pr(X_1=1)\mathbb{E}(Y|X_1=1)+\Pr(X_1=0)\mathbb{E}(Y|X_1=0)\\&=\frac{1}{2}\mathbb{E}(Y|X_1=1)+\frac{1}{2}\mathbb{E}(Y|X_1=0)\end{aligned}\tag{2.1}$$

去随机化的关键在于上式中 $\mathbb{E}(Y|X_1=0)$ 和 $\mathbb{E}(Y|X_1=1)$ 是可以**直接计算得到的**。以 $\mathbb{E}(Y|X_1=0)$ 为例，由于 $X_1=0$ 表示顶点 u_1 在 A 中，对于任意一条边 (u_i,u_j)，若该边与 u_1 相连，则它最终是否为割边取决于另一个端点是否在 B 中，所以随机变量 $Y_{i,j}$ 对条件概率贡献 $\frac{1}{2}$；若边 (u_i,u_j) 与 u_1 不相连，则它对应的随机变量 $Y_{i,j}$ 对条件概率贡献 $\frac{1}{2}$。

事实上，当任何 k 个顶点 $u_{i_1},u_{i_2},\ldots,u_{i_k}$ 已被确定所属（即去随机化）的情况下，对应的条件期望 $\mathbb{E}(Y|X_{i_1}=x_{i_1},\ldots,X_{i_k}=x_{i_k})$ 都可以结合图的具体结构直接计算——考虑图中所有边 (u_i,u_j)，若两个顶点都已被去随机化，则该边是否为割便已清楚，对应随机变量 $Y_{i,j}$ 贡献 0 或 1；若一个或零个顶点被去随机化，对应随机变量 $Y_{i,j}$ 贡献 $\frac{1}{2}$。

所以式 2.1中，两个条件期望计算后可知其中较大者。不妨设 $\mathbb{E}(Y|X_1=0)\leqslant\mathbb{E}(Y|X_1=1)$，于是有

$$\begin{aligned}\mathbb{E}(Y)&\leqslant\mathbb{E}(Y|X_1=0)\\&=\mathbb{E}(Y|X_1=0,X_2=0)\Pr(X_2=0|X_1=0)+\mathbb{E}(Y|X_1=0,X_2=1)\Pr(X_2=1|X_1=0)\\&=\frac{1}{2}\mathbb{E}(Y|X_1=0,X_2=0)+\frac{1}{2}\mathbb{E}(Y|X_1=0,X_2=1)\end{aligned}$$

以同样方式计算两个条件期望的值并取较大者，不断重复下去，最终将所有 X_i 去随机化，便得到了一个确定的割，且割集大小 $|E(A,B)|\geqslant\mathbb{E}(E(A,B))=\frac{m}{2}$。

3 LP 中的 Rounding 方法

3.1 Max-SAT 问题

一个包含 n 个布尔变量的合取范式 $\varphi(x_1,\ldots,x_n)=c_1\wedge c_2\wedge\cdots\wedge c_m$ 是由 m 个子句 $c_1,\ldots,c_m$ 合取得到的。每个子句是若干个文字（变量 x_i 或变量的非 $\overline{x_i}$）的析取，形如 $c_i=x_1\vee\overline{x_3}\vee x_4\vee x_5$。Max-SAT 问题关心在各种可能的真值赋值下，最多能有多少个子句为真。

我们先考虑 Max-SAT 问题的一个最简单的近似算法：对 n 个变量随机赋值为 T 或 F，输出满足的子句数。

对于给定的公式 φ，定义其 Max-SAT 问题的最优解为 opt，算法结果为随机变量 Y。设子句 c_i 的真假值对应随机变量 Y_i

$$Y_i=\begin{cases}0, & c_i=F\\1, & c_i=T\end{cases}$$

于是有 $Y=\sum_{i=1}^m Y_i$。考虑 Y_i，假设 c_i 中有 t_i 个文字，在变量的随机赋值下，$\mathbb{E}(Y_i)=\Pr(c_i=T)=1-\frac{1}{2^{t_i}}=p_i$，故 $\mathbb{E}(Y)=\sum_{i=1}^m p_i\geqslant mp$，其中 p 是 p_i 中的最小值。而 $opt\leqslant m$，所以

$$\frac{\mathbb{E}(Y)}{opt}\geqslant p$$

可以看到，p 取决于各个子句中最短者，且该子句越短 p 越小。此外，该随机算法也可以去随机化得到一个确定性算法，有兴趣的同学可以课下研究。

3.2 LP + rounding

事实上，Max-SAT 问题可以编码成整数线性规划 (ILP) 问题进行求解。但我们知道，线性规划（LP）存在多项式时间算法，但 ILP 是一个 NP 完全问题，没有多项式时间算法。下面我们通过随机算法利用 LP 和 rounding 的方法找 ILP 问题的解。

算法描述 以下面的 Max-SAT 具体问题为例，其中每一个变量记为 x_i，且 $x_1, \ldots, x_n \in \{0,1\}$，记每一个子式为 y_j $(j = 1, \ldots, m)$，有 $y_1, \ldots, y_m \in \{0,1\}$.

$$\begin{aligned}\varphi &= (x_1 \vee \bar{x}_2 \vee x_3) \wedge (x_1 \vee x_4) \wedge (\bar{x}_2 \vee x_4 \vee x_6 \vee x_7) \wedge x_5 \\ &= y_1 \wedge y_2 \wedge y_3 \wedge y_4\end{aligned}$$

那么原 Max-SAT 问题可以写成 ILP 的形式：

$$\begin{aligned}\max \quad & y_1 + y_2 + y_3 + y_4 \\ s.t. & \begin{cases} y_1 \leq x_1 + (1 - x_2) + x_3 \\ y_2 \leq x_1 + x_4 \\ y_3 \leq (1 - x_2) + x_4 + x_6 + x_7 \\ y_4 \leq x_5 \\ x_1, \cdots, x_7 \in \{0,1\} \\ y_1, \ldots, y_4 \in \{0,1\} \end{cases}\end{aligned}$$

注意到上面 ILP 与 LP 的区别只在于 x_i, y_j 都是 $0,1$ 变量，我们可以把这个条件放松，让 $x_i, y_j \in [0,1]$，从而得到了一个 LP 问题，但注意，这个松弛的过程会扩大解空间，有可能得到的解不是原 ILP 问题的解。

把 ILP 松弛成 LP 形式:

$$\begin{aligned}\max \quad & y_1 + y_2 + y_3 + y_4 \\ s.t. & \begin{cases} y_1 \leq x_1 + (1 - x_2) + x_3 \\ y_2 \leq x_1 + x_4 \\ y_3 \leq (1 - x_2) + x_4 + x_6 + x_7 \\ y_4 \leq x_5 \\ x_1, \cdots, x_7 \in [0,1] \\ y_1, \cdots, y_4 \in [0,1] \end{cases}\end{aligned}$$

先求解这一 LP 问题，记最优值为 $y_{LP}^{(opt)}$，相应的最优解记为：$x_1^*, x_2^*, \ldots, x_7^*, y_1^*, \ldots, y_4^* \in [0,1]$. 则：$y_{LP}^{(opt)} = y_1^* + y_2^* + y_3^* + y_4^*$. 相应的，将 ILP 问题的最优值记为 $y_{ILP}^{(opt)}$. 由于搜索范围扩大，所以显然有 $y_{LP}^{(opt)} \geqslant y_{ILP}^{(opt)}$.

接下来，通过随机舍入的方法，将 LP 的最优解**整数化**，得到一组 ILP 的解。LP 问题的最优解 $x_1^*, \ldots, x_7^* \in [0,1]$，可以看作是一组 0-1 随机变量 X_i 取值为 1 的概率：

$$\Pr(X_i = 1) = x_i^*, \quad \Pr(X_i = 0) = 1 - x_i^*, \quad i = 1, \ldots, 7$$

定义 0-1 随机变量 $Y_j, j=1,\ldots,4$，表示取定 X_i 后，第 j 个子句是否被满足。

算法近似比分析 下面分析由随机算法得到的解对应的目标函数值和 $y_{ILP}^{(opt)}$ 之间的关系。考虑 Max-SAT 中每个子句可满足的概率，以第一个子句为例：$c_1 = x_1 \vee \bar{x}_2 \vee x_3$，

$$\begin{aligned}
\Pr(Y_1=1) &= 1-\Pr(Y_1=0) \\
&= 1-\Pr(X_1=0, X_2=1, X_3=0) \\
&= 1-(1-x_1^*)\,x_2^*\,(1-x_3^*) \\
&\geq 1-\left[\frac{(1-x_1^*)+x_2^*+(1-x_3^*)}{3}\right]^3 \text{（根据均值不等式）} \\
&= 1-\left[\frac{3-(x_1^*+(1-x_2^*)+x_3^*)}{3}\right]^3 \\
&= 1-(1-\frac{y_1^*}{3})^3
\end{aligned}$$

更一般的，对于有 k 个文字的子句 c_i：

$$\Pr(Y_i=1) = 1-\Pr(Y_i=0) \geq 1-\left(1-\frac{y_i^*}{k}\right)^k \geq (1-\frac{1}{e})y_i^*.$$

因此随机算法得到的一组整数解，对应的目标函数值的期望为，

$$\mathbb{E}(Y_1+\cdots+Y_m) = \sum_{i=1}^{m}\Pr(Y_i=1) \geq \left(1-\frac{1}{e}\right)\sum_{i=1}^{m} y_i^* = \left(1-\frac{1}{e}\right)y_{LP}^{(opt)} \geq \left(1-\frac{1}{e}\right)y_{ILP}^{(opt)}$$

故近似比为 $(1-\frac{1}{e})$。