基于线性代数、数论和随机化的对边有限制的完美匹配判定算法

程泽瑞

(交叉信息研究院，学号：2019012355，手机号：17855329418)

摘　要: 二分图的匹配算法一直是理论计算机领域的热门课题之一，在工程和生活中也有着颇多应用，但是，为大家所熟知的一般二分图匹配算法只能用于对边没有限制的情形，而在对于边集有所限制时，常见的匈牙利算法、网络流算法都很难有用武之地。本文中，基于 Edmonds 矩阵行列式与完美匹配之间的一一对应关系（Lovasz, 1979），Schwartz-Zippel 引理以及 Cygan, Gabow, Sankowski 于 2015 年提出的利用矩阵乘法求解二分图最大匹配方法，同时引入原根和 k 次单位根等数论知识，将线性代数、数论和随机化相结合，得到了一种可以高效解决对边有限制的二分图完美匹配判定算法。本文将对这一算法进行描述、正确性分析和证明，并利用 C++进行编程实验得到该算法的实际表现情况。该工作是数学、计算机中几大重要方法的结合，同时利用矩阵行列式定义简化了指数级的枚举计算，通过高斯消去法在较小时间复杂度内求解出原问题，体现出数学和计算机交叉融合之美。

关键词: 完美匹配；Edmonds 矩阵；行列式；随机化；原根；k 次单位根；

1. 引言与问题描述

图匹配问题（Graph Matching）在算法和理论计算机领域一直是一个热门课题，除此之外，其在运筹学、系统等工程领域中也有着很多应用，与我们的日常生活息息相关。近些年来，随着理论计算机和组合数学的迅速发展，匹配问题的相关理论也逐渐完备化和系统化——匹配问题一般可以根据图的性质（是否为二部图）分为二分图匹配（Bipartite Graph Matching）和一般图匹配（General Graph Matching）两种。

对于一般图匹配，现有带花树（Blossom Algorithm）和最小费用最大流（Min-cost Max-flow）等方法能够在多项式时间复杂度内找到其最大匹配，这些算法能够解决所有的图匹配问题，普适性很强，但时间复杂度较劣，且没有利用图的相关性质，在日常生活和科研实际中应用并不广泛。

在实际问题中，k 类不同的物品“配对”问题，也就是说，选择尽量多的集合，满足每个集合由 k 类物品中的各一个组成，且每个物品最多被选入一个集合，是一种更为常见的假设。但是，对于 k>2 的 k 类物品配对的问题，我们可以将其规约到 d-dimensional matching 问题，而这个问题被证明是 NP-Complete 的。因此，对于 k=2 的情况，也就是二分图匹配问题，是相关研究的热门领域之一。

在本文中，我们关注二分图的完美匹配问题，其形式化定义如下——

> 给定无向二分图$G(V,E)$，V为点集，E为边集，其中V可以划分为 $V=A\cup B$，满足$A\cap B=\emptyset$ 且 $|A|=|B|$，E中所有边形如(u,v)，$u\in A$，$v\in B$，询问是否存在集合$S\subseteq E$，$|S|=|A|=|B|=\frac{|V|}{2}$，满足任意V中的点中在S中恰好出现一次。

二分图完美匹配问题起源于 Hall 于 1935 年提出的“婚姻问题”，同时，Hall 提出了著名的 Hall 定理，给出了二分图存在完美匹配问题的充分必要条件，但在实际中，Hall 定理只能在理论上将问题转化为指数级别规模的子问题，因此只适用于小规模问题，并不能高效判定完美匹配的存在性，实际中，常常通过求最大匹配判断其是否是完美匹配——1955 年，Harold Kuhn 在匈牙利数学家 Dénes Kőnig 和 Jenő Egerváry 的研究基础上，提出了匈牙利算法（Hungarian Algorithm），能够在$\Theta(|V||E|)$的时间复杂度内找到二分图的最大匹配；进一步地，二分图的最大匹配可以通过建图转化为网络流问题，具体而言，建立源点 s 和汇点 t，将源点 s 与 A 部分的点连接权值为 1 的边，将 B 部分的点与汇点 t 连接权值为 1 的边，对于边集 E 中的边(Ai,Bj)，将 Ai 与 Bj 之间连接权值为 1 的边，易证明构造出网络的 s-t 最大流即对应了二分图的一个最大匹配，而解决网络流问题，现有 Ford-Fulkerson，Edmonds-Karp 和 Dinic 等多种经典算法，利用 Dinic 算法，单位流量的最大流可以在 $\Theta(|E|sqrt(|V|))$ 时间复杂度内求出，这也是当前求解二分图匹配问题、判定二分图完美匹配存在性的通用算法。而在最新的研究中，Alex Madry 提出了时间复杂度为$\Theta(|E|^{\frac{10}{7}})$的内点法，但是，这种方法局限性很强，只能够求解没有任何限制的二分图最大匹配。对于实际情况，对于最终选进匹配 S 中的边，往往会有所限制，例如，我们考虑如下一种对边集有所限制的二分图完美匹配问题——

给定无向二分图$G(V,E)$，V为点集，E为边集，其中V可以划分为 $V = A \cup B$, 满足$A \cap B = \emptyset$ 且 $|A| = |B|$, E可以划分为$E = E_1 \cup E_2$, 满足$E_1 \cap E_2 = \emptyset$, E中所有边形如(u, v)，$u \in A$, $v \in B$, 询问是否存在集合$S \subseteq E$，$|S| = |A| = |B| = \frac{|V|}{2}$, 满足任意V中的点中在S中恰好出现一次, 且对于给定的数字k, $|S \cap E_1| \equiv 0 \pmod p$ (即E_1中选出的边数为k的倍数)。

对于以上问题，我们不仅对点有所限制，对边也有所限制，此时，若想利用前文提到的匈牙利算法和网络流算法，我们只能对于E_1集合中被选到的边集进行枚举，若数字 k 比较大（与 n 同级别），枚举的数量级将达到指数级，以上两种算法很难再有用武之地。因此，在本文中，基于 Lovasz 于 1979 年提出的用线性代数手段判断二分图是否存在完美匹配的定理，以及 Cygan, Gabow, Sankowski 的利用矩阵乘法求解二分图最大匹配的方法，沿着这种进行扩展，将线性代数和随机化相结合，可以得到一种可以高效以上问题的算法，下文中，我将对于这个算法进行描述和分析，同时通过实际编程实验分析其运行效果。

2. 相关记号、约定与定理证明

2.1 Edmonds矩阵和二分图存在完美匹配的充分必要条件

定义1 （Edmonds矩阵）. 对于二分图$G(U, V, E)$, 其中两个部分的点分别为$U = \{u_1, u_2, ...u_n\}$, $V = \{v_1, v_2, ...v_n\}$, 定义图G的Edmonds矩阵为A_{n*n},

$$A_{ij} = \begin{cases} x_{ij} & (u_i, v_j) \in E \\ 0 & (u_i, v_j) \notin E \end{cases}$$

在上述表示中，x_{ij} 对于不同的数对(i,j)是不同的变量，此时，矩阵的行列式是由|E|个变量组成的多项式。在 Edmonds 矩阵的基础上，我们有如下定理判断二分图 G 是否存在完美匹配——

定理1 (L.Lovasz 1979). 二分图$G(U, V, E)$存在完美匹配，当且仅当$det(A) \neq 0$（这里0指的是0多项式）。

利用行列式的定义，我们可以得到定理 1 的证明，如下所示——

证明. 考虑行列式的定义，设$|\sigma|$为排列σ的逆序对数量，枚举所有排列σ，我们可以得到

$$det(A) = \sum_{\sigma} (-1)^{|\sigma|} \Pi_{i=1}^{n} A_{i,\sigma_i}$$

一方面，若存在完美匹配，假设对于任意$1 \leq i \leq n$，该完美匹配中u_i与v_{p_i}两两配对，则p必然是一个排列，且$(-1)^{|p|}\Pi_{i=1}^{n} A_{i,p_i} \neq 0$ (因为(u_i, v_{p_i})之间必然有连边)，又因为不同排列对应不同的选边方式，即不同变量组合，无法两两抵消，因此，$det(A) \neq 0$。

另一方面，若$det(A) \neq 0$，则一定存在排列p满足$(-1)^{|p|}\Pi_{i=1}^{n} A_{i,p_i} \neq 0$ （否则由上式可知$det(A) = 0$)，此时，对于任意$1 \leq i \leq n$，(u_i, v_{p_i})之间必然有连边，将u_i与v_{p_i}两两配对，即可得到一组完美匹配。

综上所述，$det(A) \neq 0$是二分图$G(U, V, E)$存在完美匹配的充分必要条件。

2.2 Schwartz-Zippel引理及其证明

但是，注意到上面的矩阵 A 中元素是|E|个不同的变量，而不是常数，因此，A 的行列式是一个由|E|个变量构成的多项式，且多项式中的不同项数是|E|的指数级别（最多 2^|E|项）。因此，实际运算中，我们无法在 poly(|V|,|E|)的时间复杂度内求出矩阵 A 的行列式，我们引入 Schwartz-Zippel 引理——

引理1（Schwartz-Zippel）. 对于域$\mathcal{F}$上不恒为0的n元d度多项式$P(x_1, x_2, ...x_n)$, 设$r_1, r_2, ...r_n$为域$\mathcal{F}$等概率独立随机的n个随机数，则

$$Pr[P(r_1, r_2, ...r_n) = 0] \leq \frac{d}{|\mathcal{F}|}$$

Schwartz-Zippel 引理的证明过程如下——

证明. 我们考虑通过对n的数学归纳法证明此定理。

对于$n = 1$的情况，根据代数基本定理，d度多项式$P(x)$在域$\mathcal{F}$上最多有d个根，因此随机在$\mathcal{F}$内选择一个数，其恰好为多项式$P(x)$的根之一的概率不超过$\frac{d}{|\mathcal{F}|}$；

对于$n = k > 1$的情况，由归纳假设，假设题设条件对于任意d和任意$(k-1)$元d度多项式均成立。

将多项式$P(x_1, x_2, ...x_n)$按照x_1的系数分类，设

$$P(x_1, x_2, ...x_n) = \sum_{i=0}^{n} x_1^i P_i(x_2, x_3, ...x_n)$$

其中，由$P(x_1, x_2, ...x_n)$为d度多项式，则$P_i(x_2, x_3, ...x_n)$的度数最多为$d-i$的$(k-1)$元多项式

同时，由于$P \neq 0$，一定存在i满足$P_i(x_2, x_3, ...x_n) \neq 0$，取所有满足条件的$i$中最大的，对于随机选择的$r_2, r_3, ...r_n$，则$P(x_1, r_2, ...r_n)$是关于$x_1$的1元多项式。

同时，对于随机选择的$r_2, r_3, ...r_n$，由对$(k-1)$元多项式的归纳假设，可以得到

$$Pr[P_i(r_2, r_3, ...r_n) = 0] \leq \frac{d-i}{|\mathcal{F}|}$$

若$P_i(r_2, r_3...r_n) \neq 0$, 此时$P(x_1, r_2, ...r_n) \neq 0$，且为关于$x_1$的1元$d$度非0多项式，由归纳奠基，对于随机选择且与$r_2, ...r_n$相独立的$r_1$，可得

$$Pr[P(r_1, r_2, r_3, ...r_n) = 0 | P_i(r_2, r_3...r_n) \neq 0] \leq \frac{i}{|\mathcal{F}|}$$

由上所述，可以得到

$$\begin{aligned}
&Pr[P(r_1, r_2, r_3, ...r_n) = 0] \\
=&Pr[P(r_1, r_2, r_3, ...r_n) = 0 \cap P_i(r_2, r_3...r_n) \neq 0] + Pr[P(r_1, r_2, r_3, ...r_n) = 0 \cap P_i(r_2, r_3...r_n) = 0] \\
=&Pr[P(r_1, r_2, r_3, ...r_n) = 0 | P_i(r_2, r_3...r_n) \neq 0] Pr[P_i(r_2, r_3...r_n) \neq 0] \\
&+ Pr[P(r_1, r_2, r_3, ...r_n) = 0 | P_i(r_2, r_3...r_n) = 0] Pr[P_i(r_2, r_3...r_n) = 0] \\
\leq&\frac{i}{|\mathcal{F}|} * 1 + 1 * \frac{d-i}{|\mathcal{F}|} \\
=&\frac{d}{|\mathcal{F}|}
\end{aligned}$$

由归纳法原理，Schwartz-Zippel引理得证。

因此，我们选定质数p，令数域F_p为$[0, p-1]$形成的域（大小为p），对于$|E|$个不同的变量随机赋值$[0, p-1]$之间的数字，计算矩阵的行列式。由于矩阵A是$n * n$的，所以行列式最多为n度多项式，因此，若取p为n^2级别的质数，我们的误判率不会超过$\frac{n}{p} = \frac{1}{n}$，非常可观。

3. 算法描述与正确性分析

3.1 一些预备数论知识

考虑如何利用前文所述的 Lovasz 定理和 Edmonds 矩阵判断权值模 k 为 0 完美匹配的存在性，我们利用相关数论知识可以实现这一点，在描述具体算法之前，我们先介绍阶、原根和 k 次单位根的概念以及相关算法。

定义2 （阶）. 由欧拉定理可知，对$a \in \mathbb{Z}$，$m \in \mathbb{N}^*$，若 $gcd(a,m)=1$ (gcd指最大公约数)，则$a^{\phi(m)} \equiv 1 \pmod m$($\phi$为欧拉函数)。因此满足同余式 $a^n \equiv 1 \pmod m$ 的最小正整数n 存在，称作a模m的阶，记作$\delta_m(a)$。

定义3 （原根）. 对于$a \in \mathbb{Z}$，$m \in \mathbb{N}^*$，若 $gcd(a,m)=1$ (gcd指最大公约数)，且$\delta_m(a)=\phi(m)$，则a称为模m的原根。

定义4 （k次单位根）. 对于$a \in \mathbb{Z}$，$m \in \mathbb{N}^*$，若 $gcd(a,m)=1$ (gcd指最大公约数)，且$\delta_m(a)=k$，则a称为模m的k次单位根。

下面，我们有如下定理，可以用来检验原根的存在性，并高效判定给定的数字 g 是否是质数 p 的原根。

定理2 （原根存在原理）. 一个数m存在原根，当且仅当$m=2,4,p^\alpha,2p^\alpha$，其中p为奇质数，α为正整数。

定理3 （原根判定原理）. 若一个数g是模p的原根，则有对于$p-1$任何大于1且不为自身的因数d，都有$g^{(p-1)/d} \not\equiv 1 \pmod p$。

定理 2、3 的证明利用到分类讨论、拉格朗日定理、裴蜀定理和反证法，由于其与主题关系不大，在此略去，同时，我国著名数学家王元于 1959 年证明了最小原根的数量级，如定理 4 所示——

定理4 （最小原根的数量级，王元 1959）. 对于任意有原根的数$m \in \mathbb{N}^*$，其最小原根不超过$O(m^{0.25})$数量级。

由**定理 2、3、4**，我们可以得到如下算法，可以高效地在亚线性时间内求出质数 p 的一个原根。

算法1. (求质数p的某一个原根)： 首先，根据定理2，质数p一定存在原根。对$(p-1)$作质因数分解，利用最朴素的枚举法可以在$O(p^{0.5})$时间内求出$(p-1)$的所有因子，因子数量同样不会超过$O(p^{0.5})$数量级。从小到大枚举原根g, 利用定理3，对于每个满足g，我们遍历$(p-1)$的所有因子d，计算$g^{(p-1)/d}$ mod p，将结果与1比较，如果均不为1，则g就是原根，退出循环；否则，遍历下一个g重复上述过程。由于每次检验需要遍历不超过$O(p^{0.5})$个因子，对于每个因子，用快速幂计算$g^{(p-1)/d}$ mod p的复杂度是$O(\log p)$的。同时，定理4保证了，最多遍历$O(p^{0.25})$轮即可得到结果，因此，算法1的总时间复杂度是$O(p^{0.5}+p^{0.5}\log p * p^{0.25}) = O(p^{0.75}\log p)$，能够在亚线性时间内找到质数p的某一个原根$g$。

接下来，我们有以下定理证明 k 次单位根的存在性（以下公式中，除法默认为向下取整）。

定理5 （质数p存在k次单位根的充要条件）. 质数p存在k次单位根的充要条件是，$p \equiv 1 \pmod k$。

证明. 先用反证法证明必要性，假设g是p的k次单位根且$p \not\equiv 1 \pmod k$，则$g^{p-1} \equiv g^{(p-1) mod k} * (g^k)^{(p-1)/k} \equiv g^{(p-1) mod k} * 1^{(p-1)/k} \equiv g^{(p-1) mod k} \pmod k$, 由$(p-1) \not\equiv 0 \pmod k$且$g$是$p$的k次单位根，则$g^{(p-1) mod k} \not\equiv 1 \pmod k$，即$g^{p-1} \not\equiv 1 \pmod k$，与费马小定理矛盾，因此$p \equiv 1 \pmod k$是必要条件。

充分性，我们可以通过构造法证明。对于质数p,设g是其原根之一，则我们认为，$p_k = g^{(p-1)/k}$ mod p是质数p的一个k次单位根。首先${p_k}^k \equiv (g^{(p-1)/k})^k \equiv g^{p-1} \equiv 1 \pmod p$. 其次，反证法，假设存在$k' < k$满足${p_k}^{k'} \equiv 1 \pmod p$, 令$t = (p-1)/k * k'$, 则$g^t = (g^{(p-1)/k})^{k'} \equiv {p_k}^{k'} \equiv 1 \pmod p$，又由$k' < k$，则$t < p-1$，与$g$为原根矛盾。综上所述，则$p_k$是质数$p$的一个$k$次单位根。

在**定理 5** 的基础上，我们可以得到以下算法，在亚线性时间内求出质数 p 的一个 k 次单位根。

算法2. (求质数p的某一个k次单位根)： 利用定理5中的充分性证明，对于满足$p \equiv 1 \pmod k$的质数p，我们首先用算法1在$O(p^{0.75}\log p)$内求出p的一个原根g。接下来，我们直接用一次快速幂求出$p_k = g^{(p-1)/k} \mod p$，即可以$O(p^{0.75}\log p)$时间复杂度内求出p的一个k次单位根p_k。

实际运行结果表明，算法 2 的实际复杂度远达不到上界，具体分析可见第 4 部分。

3.2 算法描述与证明

在以上基础上，我们考虑对 Edmonds 矩阵进行如下扩展，以解决带有限制的完美匹配问题。

首先，对于给定的 k，我们随机找一个大质数 p 满足 p 存在一个 k 次单位根（即随机数字 x，计算出 kx+1，判断其是否为质数，如果是，由**定理 5**，该质数 p=kx+1 一定存在 k 次单位根），通过**算法 2** 求出其 k 次单位根之一，设其为 g，则对于二分图 G(U,V,E) （G 中各部分定义和**定义 1** 中相同），我们定义一组矩阵如下——

对于所有$0 \le R \le k-1$，定义矩阵A^R满足

$$A_{ij}^R = \begin{cases} g^R * x_{i,j} & (u_i, v_j) \in E_1 \\ x_{i,j} & (u_i, v_j) \in E_2 \\ 0 & (u_i, v_j) \notin E \end{cases}$$

此时，通过对 Lovasz 定理的扩展，我们可以证明如下定理——

定理6. 二分图$G(U,V,E)$存在E_1中边数为k的倍数完美匹配，当且仅当$\sum_{R=0}^{k-1} det(A^R) \not\equiv 0 \pmod p$（这里0指的是0多项式）。

证明. 由定理1的证明和行列式的定义，可得，对于所有完美匹配$\{(u_i, v_{p_i})\}$(p为完美匹配对应排列)，设排列p对应的完美匹配中有w条边在集合E_1中，则

$$det(A^R) \equiv \sum_p (-1)^{|p|} (g^R)^w \Pi_{i=1}^n x_{i,p_i} \pmod p$$

因此，可以得到

$$\sum_{R=0}^{k-1} det(A^R) \equiv \sum_p (-1)^{|p|} (\sum_{R=0}^{k-1} g^{Rw}) \Pi_{i=1}^n x_{i,p_i} \pmod p$$

其中，若$w = tk, t \in \mathbb{Z}$,

$$\begin{aligned} \sum_{R=0}^{k-1} g^{Rw} &\equiv \sum_{R=0}^{k-1} (g^k)^{tR} \\ &\equiv \sum_{R=0}^{k-1} 1^{tR} \equiv \sum_{R=0}^{k-1} 1 \equiv k \pmod p \end{aligned}$$

其中，若w不是k的整数倍，则$g^w \not\equiv 1 \pmod p$(由k次单位根定义)，

$$\begin{aligned}\sum_{R=0}^{k-1} g^{Rw} &\equiv \sum_{R=0}^{k-1}(g^w)^R \\ &\equiv \frac{1-(g^w)^k}{1-g^w} \\ &\equiv (1-(g^k)^w) * (1-g^w)^{-1} \\ &\equiv (1-1) * (1-g^w)^{-1} \equiv 0 \pmod p\end{aligned}$$

由上所述，可得

$$\begin{aligned}&\sum_{R=0}^{k-1} det(A^R) \\ &\equiv \sum_p (-1)^{|p|}(\sum_{R=0}^{k-1} g^{Rw})\Pi_{i=1}^n x_{i,p_i} \pmod p \\ &\equiv k * \sum_{p,k|w}(-1)^{|p|}\Pi_{i=1}^n x_{i,p_i} \pmod p\end{aligned}$$

对于所有满足$k|w$，即所选E_1中边数为k倍数的完美匹配，因为所有$\{x_{i,p_i}\}$对应的变量集合不完全相同，所以不可能抵消，因此，$\sum_{R=0}^{k-1} det(A^R) \not\equiv 0 \pmod p$是二分图$G(U,V,E)$存在$E_1$中边数为$k$的倍数完美匹配的充要条件。

因为有限的求和操作不会增加多项式的度数，所以这k个行列式之和仍然为最高为n度的$|E|$元多项式，其定义域F_p大小为p，因此，利用Schwartz-Zippel引理，我们可以在$[0,p-1]$区间中对$|E|$个变量随机赋值，分别在模p意义下计算出k个行列式值，再求和判断是否为0。若为0，则认为不存在符合条件的完美匹配；否则，认为存在符合条件的完美匹配。该算法执行一次的误判概率不超过$\frac{n}{p}$, 如果取p为n^2级别的质数，我们的误判率不会超过$\frac{1}{n}$. 如果重复执行t次，我们的错误概率将会被缩小至$\frac{1}{n^t}$.

接下来，我们分析上述算法的最坏情况时间复杂度。

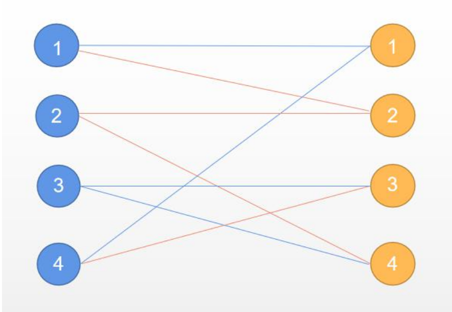
首先，假设所取的质数p的数量级是$O(p)$，随机找到一个满足$p = kx+1$的质数期望$O(\log p)$轮可以找到（由质数的分布定律），每一轮判断用最朴素的方式，复杂度为$O(p^{0.5})$，因此，找到一个符合条件的质数p期望时间复杂度是$O(p^{0.5}\log p)$，接下来，利用前述**算法2**，我们可以在$O(p^{0.75}\log p)$时间复杂度内找到p的一个k次单位根g。接下来，我们可以根据上述算法，构造出k个矩阵A^R，其中$0 \leq R \leq k-1$, 同时，对于$|E|$个变量随机$|E|$个$[0,p-1]$范围内的随机数，将矩阵元素变为纯数字，最后，对这k个矩阵分别用高斯消元法求行列式，时间复杂度为$O(kn^3)$。综上所述，执行1次该算法的时间复杂度上界为$O(p^{0.75}\log p + kn^3)$，误判率不超过$\frac{n}{p}$。特别地，如果取$p = O(n^4)$，时间复杂度为$O(n^3\log n + kn^3)$，误判率不超过$O(\frac{1}{n^3})$。

进一步地，我们可以发现，这个做法拥有良好的数值稳定性。因为我们把所有运算限制在了一个大小为 p 的有限域内，所以实际编写 C/C++代码时，如果 p 不超过 int 范围，我们只需要在 int 范围内进行每一步运算，既不需要时间复杂度高的高精度大整数类，也不需要用 double 等浮点数类型导致精度丢失。因此，这种在有限域内随机的算法拥有良好的数值稳定性和实际表现，不需要担心由于误差存在而影响最终结果的情况发生。

3.3 举例分析

为了更好阐述 3.2 中所述算法，在这一部分，我们将通过举例，剖析以上算法的流程。

考虑对如下图所示的二分图，其中 n=4，取 k=3 时，边集被划分为 E1 和 E2 集合，其中橙色边代表 E1 中的边，蓝色边代表 E2 中的边，问题是——判断是否存在 E1 中选择边数为 k=3 的倍数的完美匹配。



执行上述算法，我们取$x = 6$，得到质数$p = kx + 1 = 19$，同时，通过**算法1**，我们可以得到2是$p = 19$的一个原根，再执行**算法2**，可以得出$g = 2^{(19-1)/3} \mod p = 7$是$p = 19$的一个$k = 3$次单位根。

由$g^0 \equiv 1 \pmod{19}, g^1 \equiv 7 \pmod{19}, g^2 \equiv 11 \pmod{19}$, 因此，我们可以得到3个矩阵$A^0$, A^1和A^2为:

$$A^0 = \begin{bmatrix} x_{1,1} & x_{1,2} & 0 & 0 \\ 0 & x_{2,2} & 0 & x_{2,4} \\ 0 & 0 & x_{3,3} & x_{3,4} \\ x_{4,1} & 0 & x_{4,3} & 0 \end{bmatrix}$$

$$A^1 = \begin{bmatrix} x_{1,1} & 7x_{1,2} & 0 & 0 \\ 0 & 7x_{2,2} & 0 & 7x_{2,4} \\ 0 & 0 & x_{3,3} & x_{3,4} \\ x_{4,1} & 0 & 7x_{4,3} & 0 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} x_{1,1} & 11x_{1,2} & 0 & 0 \\ 0 & 11x_{2,2} & 0 & 11x_{2,4} \\ 0 & 0 & x_{3,3} & x_{3,4} \\ x_{4,1} & 0 & 11x_{4,3} & 0 \end{bmatrix}$$

将所有8个变量替换为$[0, p-1] = [0, 18]$范围内的随机数，令$x_{1,1} = 16, x_{1,2} = 3, x_{2,2} = 2, x_{2,4} = 1, x_{3,3} = 17, x_{3,4} = 12, x_{4,1} = 10, x_{4,3} = 6$, 在数域$F_p$内计算三个矩阵的行列式（即模p意义下），可以得到

$$det(A^0) = det \begin{bmatrix} 16 & 3 & 0 & 0 \\ 0 & 2 & 0 & 1 \\ 0 & 0 & 17 & 12 \\ 10 & 0 & 6 & 0 \end{bmatrix} \equiv 11 \pmod{19}$$

$$det(A^1) = det \begin{bmatrix} 16 & 2 & 0 & 0 \\ 0 & 14 & 0 & 7 \\ 0 & 0 & 17 & 12 \\ 10 & 0 & 4 & 0 \end{bmatrix} \equiv 7 \pmod{19}$$

$$det(A^2) = det \begin{bmatrix} 16 & 14 & 0 & 0 \\ 0 & 3 & 0 & 11 \\ 0 & 0 & 17 & 12 \\ 10 & 0 & 9 & 0 \end{bmatrix} \equiv 1 \pmod{19}$$

因此，$\sum_{R=0}^{k-1} det(A^R) \equiv 11 + 7 + 1 \equiv 0 \pmod{19}$，原二分图不存在$E_1$中选择边数为3的倍数的完美匹配。而代入原图中验证，可以发现确实不存在符合条件的完美匹配，算法得到的结果正确。

4. 代码实现与实验结果

基于第 3 部分提到的算法，我编写了 C++程序对该算法的实际效率进行了测试，其中，随机数用到的随机源是梅森旋转算法系中的 mt19937，质数 p 选取为 10^9 级别的大质数，行列式的运算用高斯消去法在模 p 意义下实现，由于 p 比较大，在 n 不超过 1000 时出错概率不超过百万分之一，因此算法执行次数为 1。

为了测试代码的实际运行情况，测试用例根据给定的参数 n,d,s 随机生成（其中，n 表示二分图一个部分的点数（即图中总点数为 2n），d 表示图的密度（即对于给定的 d，在 2n 个点的完全二分图中随机 dn 条边作为这组测例的边集）；s 为[0,10]范围内的整数，表示 E1 中边占总边数的占比，即对于全部 dn 条边，每条边独立地以 s/10 的概率划分进 E1 集合，以(10-s)/10 的概率划分进 E2 集合；同时，k 的取值在[1,n]范围内等概率随机。对于不同的参数组合(n,d,s)，均测试了 100 组数据（对于 n=500，取的是 10 组）取平均，作为最终的结果。

该算法的效率如下表所示（电脑配置为 Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz）：

运行时间	d=2,s=2	d=2,s=5	d=2,s=8	d=10,s=2	d=10,s=5	d=10,s=8	d=n/2,s=2	d=n/2,s=5	d=n/2,s=8
n=50	11.2ms	14.0ms	17.1ms	14.3ms	14.5ms	15.9ms	15.8ms	16.1ms	16.2ms
n=100	178.8ms	172.6ms	187.3ms	169.9ms	170.9ms	176.4ms	181.1ms	179.6ms	180.5ms
n=200	1.956s	2.367s	2.414s	2.232s	2.275s	2.364s	2.372s	2.423s	2.510s
n=500	60.581s	61.255s	61.549s	59.978s	60.322s	60.178s	63.652s	63.157s	69.062s

由上表可得，对于 n 不超过 500 的数据，该算法在一般的家用电脑上能够在 1 分钟内得出结果，效率比较客观。同时，该算法对于不同的数据，运行时间非常稳定，因此该算法的鲁棒性很强，适用于各种场景。

同时，通过减小 p 的取值，我们可以得到算法的正确率，如下表所示（p 指的是实际选择质数 p 的数量级，下表数据对于每一组不同的(n,p)，参数(d,s)随机，基于 1000 组独立实验得到）：

正确率	p=3n	p=6n	p=10n	p=15n	p=n^2	p=n^3	p=10^3	p=10^5
n=10	96.7%	97.7%	99.0%	99.4%	98.9%	99.8%	99.8%	100%
n=20	97.0%	98.8%	99.1%	99.6%	99.7%	100%	99.8%	100%
n=50	98.0%	99.1%	99.6%	99.7%	100%	100%	99.9%	100%

由上表可得，该算法的实际正确率远远高于前文中 Schwartz-Zippel 引理给出的正确率下界，实际表现中，其正确率更接近于 1/p，因此，一般来说，取 p 为 n^3 级别即可保证算法的高正确率。

综上所述，该算法的实际表现与理论证明相符，可以高效地判定此类对边有限制的二分图完美匹配问题。

5. 总结与展望

本文中，通过线性代数、数论和随机化方式相结合，在 Edmonds 矩阵、Lovasz 定理的基础上，通过引入原根和 k 次单位根的数论知识，我们得到了一种可以在 O(n^4)时间复杂度以内实现对边有限制的二分图完美匹配的判定算法，该算法的亮点有以下两点——首先，通过行列式的定义，利用行列式运算代替了复杂度为指数级的枚举过程，而行列式可以利用高斯消去法在 O(n^3)时间复杂度内高效求出，大大降低了计算复杂度；其次，对于边上的限制，利用数论中原根相关的知识，将其与行列式的计算融为一体，从而通过设置 Edmonds 矩阵变量前的参数实现了对边的限制，同时，在这里的转化中，在整数域上的运算被转化为了有限数域 Fp 内的运算，因此不会出现运算中间结果过大导致丢失精度等问题，使得该算法具有良好的数值稳定性。通过编程实验，我们也验证了此算法具有高效、鲁棒等特性，可以应用于生产生活实际中。我认为，这个算法是数学领域的数论、线性代数和计算机领域的数值计算、随机化等知识的完美融合，体现出数学和计算机学科的交叉之美，同时，这种利用数学知识简化计算复杂度的思路，也有着非常广阔的应用前景，是非常值得探索的研究方向。

当然，以上算法也有着其局限性，并有待其进一步优化。关于在此主题上的延伸以及将来研究方向的展望，我认为可以分为以下几个部分——

1. 如何将此算法扩展到一般图

上述算法仅仅适用于二分图的情形，如何将此算法扩展到一般图，是一个值得讨论的有趣问题。

以上算法基于 Lovasz 定理和对 Edmonds 矩阵基于数论知识的修改，幸运的是，Lovasz 定理在一般图上有着其扩展，具体如下所述——

定义6 （Tutte矩阵）. 对于图$G(V,E)$, 其中$|V|=n$，定义图G的Tutte矩阵为A_{n*n}, 满足

$$A_{ij}=\begin{cases} x_{ij} & \text{if } (i,j)\in E \text{ and } i<j \\ -x_{ji} & \text{if } (i,j)\in E \text{ and } i>j \\ 0 & \text{otherwise} \end{cases}$$

定理7 （Tutte定理）. 一般图$G(V,E)$存在完美匹配，当且仅当$det(A)\neq 0$（这里0指的是0多项式）。

利用 Tutte 定理和 Tutte 矩阵，可能能够在一般图上得到与前文所述算法相似的结果，关于该算法在一般图上的扩展与表现，仍然有待进一步探索和深入研究。

2. 如何进一步优化该算法的计算复杂度

在第 4 部分的编程实验中，在进行矩阵行列式的求解时，采用的是时间复杂度为 O(n^3)的高斯消去法。

但是，我们知道，行列式的求解可以转化为矩阵乘法，而矩阵乘法的解决有着时间复杂度更加优秀的算法，例如《算法导论》中介绍的广为人知的 Strassen 矩阵乘法（其时间复杂度约为 O(n^2.81)），以及近些年来的一些最新论文，已经将最优复杂度提升至约 O(n^2.373)。然而，我在第 4 部分的编程实验中，也对 Strassen 算法进行了尝试，但由于其常数比较大，在数据规模较小时（n 不超过 500），实际运行时间劣于 O(n^3)的高斯消去法。因此，任何矩阵乘法复杂度的改进或者常数的减小，都将提升上述算法的实际表现，让该算法能够适用于更大的数据规模，矩阵乘法的优化，也是在此主题上的延伸与扩展方向之一。

3. 是否存在多项式时间复杂度的确定性算法解决此问题

最后，是否存在多项式时间复杂度的确定性算法，也是值得探讨的问题之一。如上所述，本文中的算法基于随机化，其正确率并没有达到 100%，在一些对于正确率要求很高的场景中难以应用。因此，该问题是否存在多项式时间复杂度的确定性算法，同样是有关此主题可能的探究方向之一。

综上所述，对边有限制的二分图完美匹配和一般图完美匹配判定问题，是一个充满着很多可能性和未知的方向，它结合了数值计算、数论、线性代数和随机化等多门学科，无论在科研领域，还是在生产生活实际中，都有着很多应用场景，也非常值得我们进行进一步的研究探索。

参考文献

[1] Lovász L. On determinants, matchings, and random algorithms[C]//FCT. 1979, 79: 565-574.
[2] R. Motwani, P. Raghavan . Randomized Algorithms.[C]. 1995, Cambridge University Press. p. 167. ISBN 9780521474658.
[3] Mucha M, Sankowski P. Maximum matchings via Gaussian elimination[C]//45th Annual IEEE Symposium on Foundations of Computer Science. IEEE, 2004: 248-255.
[4] Cygan M , G Ab Ow H N , Sankowski P . Algorithmic Applications of Baur-Strassen's Theorem: Shortest Cycles, Diameter and Matchings[J]. Journal of the ACM, 2015, 62(4):1-30.
[5] Tutte W T. The factorization of linear graphs[J]. Journal of the London Mathematical Society, 1947, 1(2): 107-111.
[6] Strassen V. Gaussian elimination is not optimal[J]. Numerische mathematik, 1969, 13(4): 354-356.
[7] Coppersmith D, Winograd S. Matrix multiplication via arithmetic progressions[C]//Proceedings of the nineteenth annual ACM symposium on Theory of computing. 1987: 1-6.

[8] Gall F L . Powers of Tensors and Fast Matrix Multiplication[J]. ACM, 2014.
[9] P Erdős, Shapiro H N . On the least primitive root of a prime[J]. Pacific Journal of Mathematics, 1957, 7(1).
[10] 王元. 论素数的最小正原根[C]. 数学学报. 1959;4.