Advances in Applied Mathematics 应用数学进展, 2025, 14(8), 68-75
Published Online August 2025 in Hans. https://www.hanspub.org/journal/aam
https://doi.org/10.12677/aam.2025.148371

Hans汉斯

基于随机算法求解多重线性PageRank问题

李贤艳

广东工业大学数学与统计学院，广东 广州

收稿日期：2025年7月5日；录用日期：2025年7月28日；发布日期：2025年8月6日

摘　要

本文针对多重线性PageRank问题，采用随机算法求解该模型，给出收敛性定理，并通过数值实验说明随机算法求解多重线性PageRank问题的有效性。

关键词

多重线性PageRank问题，随机算法

The Random Algorithm for Solving Multilinear PageRank Problem

Xianyan Li

School of Mathematics and Statistics, Guangdong University of Technology, Guangzhou Guangdong

Received: Jul. 5th, 2025; accepted: Jul. 28th, 2025; published: Aug. 6th, 2025

Abstract

This paper focuses on the multilinear PageRank problem, adopts the random algorithms to solve the problem, presents the convergence theorem, and demonstrates the effectiveness of the random algorithms in solving the multilinear PageRank problem through numerical experiments.

Keywords

Multilinear PageRank Problem, Random Algorithm

文章引用: 李贤艳. 基于随机算法求解多重线性 PageRank 问题[J]. 应用数学进展, 2025, 14(8): 68-75.
DOI: 10.12677/aam.2025.148371

1. 引言

Google 首次提出 PageRank 模型来决定一个代表网页的有向图的重要节点，即网页排序问题[1]。给定一个有向图的随机行走，修正后的 PageRank 模型建立一个存在唯一稳定分布的马尔科夫链。由于 PageRank 模型的广泛成功，再根据如今实际应用中高维数据的需要，Gleich 等[2]提出了高阶 PageRank 问题，它是一阶问题的推广。与此同时他们也发现了高阶 PageRank 的存储障碍[2]，之后，他们根据秩 1 对称近似估计，希望利用这种估计的存储量小的优势，进一步提出多重线性 PageRank 模型。

Gleich 等[2]基于秩 1 近似估计技术提出多重线性 PageRank 问题：令 P 是一个 m 阶随机张量，概率 $\alpha\in(0,1)$，v 是一个随机向量，则多重线性 PageRank 问题定义如下：

$$x=\alpha Px^{m-1}+(1-\alpha)v, \tag{1}$$

其中，x 是所求的随机向量(被称为多重线性 PageRank 向量)，并且

$$Px^{m-1}=\sum_{i_2,i_3,\cdots,i_m}^{n}p_{ii_2\cdots i_m}x_{i_2}\cdots x_{i_m}.$$

近年来，随着对多重线性 PageRank 问题研究的不断深入，诸多学者纷纷涌现出创新性的求解方法。这些方法旨在更高效地处理复杂网络中的多重线性关系、以更短的计算时间和更少的迭代次数求出多重线性 PageRank 值。例如，Bucci 和 Poloni [3]提出了一种计算多重线性 PageRank 的延拓方法：预测校正延拓算法。Meini 和 Poloni [4]利用二次向量方程理论，提出了基于 perron 的多重线性 PageRank 算法。Cipolla 等人[5]通过采用重新启动形式的简化拓扑－算法，引入了不动点多重线性 PageRank 计算的外推方法。Boubekraoui 等人[6]提出了一种利用向量 Aitken 外推技术计算多重线性 PageRank 向量的新方法。Lai 等[7]将 Anderson 加速技术集成到现有的松弛不动点迭代中，用于计算多重线性 PageRank 问题。Zhou 等[8]基于[1]中的不动点迭代和内外迭代，提出了一种两步迭代法(FPIO)来计算多重线性 PageRank 问题。然后考虑多步不动点迭代，并将其作为内外迭代的前提条件，给出了一种新方法：MFPIO 算法，该方法可以看作是 FPIO 方法的扩展。此外，将最小多项式外推(MPE)作为不动点迭代的加速技术以加快多重线性 PageRank 向量的计算速度，从而提出了一种新方法：FPMPE 算法。针对多重线性 PageRank 问题，结合松弛技术，提出了新的张量分裂算法。Bentbib 等人[9]提出了一些向量外推方法，如最小多项式外推(MPE)和降秩外推(RRE)可以用来加速定点多重线性 PageRank 的计算。

本文的主要贡献：1) 提出了采用随机算法求解多重线性 PageRank 问题；2) 给出数值实验说明随机算法求解多重线性 PageRank 问题的有效性。

2. 预备知识

首先介绍本文用到的符号和基本知识。令 $\|\cdot\|$ 表示欧氏范数，矩阵 A 的 F 范数为 $\|A\|_F=\sqrt{\sum_{i,j}A_{ij}^2}$，$A^+$ 表示矩阵 A 的广义逆，$\|A^+\|_2=\frac{1}{\delta_{\min}}$，$\delta_{\min}$ 为矩阵 A 的最小非零奇异值，$k_F(A)=\|A\|_F\|A^+\|_2$。令 $x\in\mathbb{R}^n$，当向量 x 满足 $x\geq 0$ 且 $\|x\|_1=1$ 时，称 x 为随机向量。P 是一个 m 阶 n 维张量，$P=\left(p_{i_1i_2\cdots i_m}\right)$，$\left(p_{i_1i_2\cdots i_m}\right)\in\mathbb{R}$，$i_1,i_2,\cdots,i_m\in n$。

对于多重线性 PageRank 问题(1.1)，与它等价的另一个形式为：

$$x=\alpha R\underbrace{(x\otimes\cdots\otimes x)}_{m-1}+(1-\alpha)v, \tag{2}$$

其中⊗为Kronecker积，R 为(1.1)中张量 P 的模1展开。令 P 为一个三阶 n 维随机张量，即 $P=\left(p_{ijk}\right)\in\mathbb{R}^{[3,n]}$，

其模 1 展开如下：

$$P_{(1)}=\begin{bmatrix} p_{111} & \cdots & p_{1n1} & p_{112} & \cdots & p_{1n2} & p_{11n} & \cdots & p_{1nn} \\ p_{211} & \cdots & p_{2n1} & p_{212} & \cdots & p_{2n2} & p_{21n} & \cdots & p_{2nn} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ p_{n11} & \cdots & p_{nn1} & p_{n12} & \cdots & p_{nn2} & p_{n1n} & \cdots & p_{nnn} \end{bmatrix}$$

给定一个向量 $x\in\mathbb{R}^n$，$P\in\mathbb{R}^{[m,n]}$，Px^{m-1} 的计算结果为一个向量，即 $Px^{m-1}\in\mathbb{R}^n$，该向量具体的项为：

$$\left(Px^{m-1}\right)_i=\sum_{i_2,\cdots,i_m=1}^{n}p_{ii_1\cdots i_m}x_{i_2}\cdots x_{i_m},\ i=1,\cdots,n.$$

$Px^{m-2}\in\mathbb{R}^{n\times n}$ 为矩阵，该矩阵定义如下；

$$\left(Px^{m-2}\right)_{i_1i_2}=\sum_{i_3,\cdots,i_m=1}^{n}p_{i_1i_2\cdots i_m}x_{i_3}\cdots x_{i_m},\ 1\le i_1,i_2\le n.$$

3. 收敛性定理

定义 3：[10]若对于每个 $i\in\{1,2,\cdots,m\}$ 且 $\forall x_1,x_2\in\mathbb{R}^n$，存在常数 $\eta_i\in[0,\eta)$，$(\eta=\max_i\eta_i<\frac{1}{2})$，使得以下式子成立：

$$\left|f_i(x_1)-f_i(x_2)-\nabla f_i(x_1)(x_1-x_2)\right|\le\eta_i\left|f_i(x_1)-f_i(x_2)\right|$$

则称函数 $f:\mathbb{R}^n\to\mathbb{R}^m$ 满足局部切向锥条件。

引理 3：如果函数 f 满足局部切向锥条件，对于每个 $1\le i\le m$ 以及 $\forall x_1,x_2\in\mathbb{R}^n$，有以下式子成立：

$$\frac{1}{1+\eta_i}\left|\nabla f_i(x_1)(x_1-x_2)\right|\le\left|f_i(x_1)-f_i(x_2)\right|$$

定理 3：对于非线性问题 $f(x)=0$，函数 $f:D\subseteq\mathbb{R}^n\to\mathbb{R}^m$ 的定义域 D 是有界闭集，并且有 $x_*\in\mathbb{R}^n$ 使得 $f(x_*)=0$。若函数 f 的偏导函数连续且对于 $\forall x\in D$，$f'(x)$ 均为列满秩的行下有界矩阵，对于每个 $i\in\{1,2,\cdots,m\}$，函数 $f_i(x)$ 满足局部切向锥条件，$\eta=\max_i\eta_i$，$(\eta<\frac{1}{2})$，则 NRK 算法依期望线性收敛

$$E\left\|x_k-x_*\right\|^2\le\frac{1-2\eta}{m(1+\eta)^2k_F^2\left(f'(x_{k-1})\right)}E\left\|x_{k-1}-x_*\right\|^2$$

4. 提出的算法

通过深入钻研求解大规模线性问题的 Kaczmarz 算法及其随机版本，王等[10]将 Kaczmarz 算法及其随机化版本的迭代思路以及随机选择概率引入到大规模非线性问题的求解中，根据所选择的随机选择投影行的概率方式的不同，提出的三种非线性 Kaczmarz 算法，分别是非线性 Kaczmarz 算法(NK)、非线性随机 Kaczmarz 算法(NRK)、非线性均匀随机 Kaczmarz 算法(NURK)。本文利用这三种非线性 Kaczmarz 算法求解多重线性 PageRank 问题，并通过数值实验验证求解的有效性。

令 $x\in\mathbb{R}^n$，且 $x_+=\max(x,0)$ 为非零向量，定义投影算子如下：

$$proj(x)=\frac{x_+}{\left\|x_+\right\|_1}.$$

易证 $proj(x)$ 是随机向量。

对于多重线性 PageRank 问题 $x=\alpha Px^{m-1}+(1-\alpha)v$，令 $f(x)=x-\alpha Px^{m-1}-(1-\alpha)v$，下面给出求解多重线性 PageRank 问题的三种算法框架(算法 1~3)：

算法 1. 非线性 Kaczmarz (NK)算法

1：给定 $f(x)$，初始值 x_0，停机误差 ε，最大迭代步数 $k_{\max}$，$r=f(x)$，令 $k=0$

2：while $\|r\|>\varepsilon$ & $k<k_{\max}$

3：取 $i=\mathrm{mod}(k,m)+1$

4：梯度计算 $g_i=\nabla f_i(x_k)$

5：迭代 $x_{k+1}=x_k-\dfrac{r(i)}{\|g_i\|}g_i$

6：$k=k+1$

7：计算残差 $r=f(x_k)$

8：end while

9：输出 x_k

算法 2. 非线性随机 Kaczmarz (NRK)算法

1：给定 $f(x)$，初始值 x_0，停机误差 ε，最大迭代步数 $k_{\max}$，$r=f(x)$，令 $k=0$

2：while $\|r\|>\varepsilon$ & $k<k_{\max}$

3：依概率 $p_i=\dfrac{|f_i(x)|^2}{\|f(x)\|^2}$ 选择行指标 $i\in\{1,2,\cdots,m\}$

4：梯度计算 $g_i=\nabla f_i(x_k)$

5：迭代 $x_{k+1}=x_k-\dfrac{r(i)}{\|g_i\|}g_i$

6：$k=k+1$

7：计算残差 $r=f(x_k)$

8：end while

9：输出 x_k

算法 3. 非线性均匀随机 Kaczmarz (NURK)算法

1：给定 $f(x)$，初始值 x_0，停机误差 ε，最大迭代步数 $k_{\max}$，$r=f(x)$，令 $k=0$

2：while $\|r\|>\varepsilon$ & $k<k_{\max}$

3：依概率 $p_i=unidrnd(m,1,1)$ 选择行指标 $i\in\{1,2,\cdots,m\}$

4：梯度计算 $g_i=\nabla f_i(x_k)$

5：迭代 $x_{k+1}=x_k-\dfrac{r(i)}{\|g_i\|}g_i$

6：$k=k+1$

7：计算残差 $r=f(x_k)$

8：end while

9：输出 x_k

记 $f(x)=(f_1(x),\cdots,f_m(x))^{\mathrm{T}}$，$f'(x)$ 为在点 x 的偏导数矩阵。$\nabla f_i(x)$ 是偏导数矩阵 $f'(x)$ 的第 i 行。

5. 数值实验

基于多重线性 PageRank 问题，我们将通过数值实验来验证所提出算法的有效性。主要从迭代步数

(IT)，以秒为单位的迭代时间(CPU)，误差(err)这三个方面来检验所提算法的效率。其中，误差 err 定义如下

$$err = \left\| x - \alpha P x^{m-1} - (1-\alpha) v \right\|,$$

停机误差为 $err = 10^{-10}$，最大迭代步数设置为 30,000，选取参数 α 为 $\alpha = 0.95$，初始值为 $x_0 = \left(\frac{1}{n}, \frac{1}{n}, \cdots, \frac{1}{n} \right)$，$n$ 为张量的维数。实验是在一台配备 1.80 GHz 4 核 Intel (R) Core (TM) i5-8250U CPU 以及 8 GB 2400 MHz DDR3 内存的计算机上，使用 MATLAB (版本 R2018a)进行的，并且我们还使用了 MATLAB 的 Tensor Toolbox 工具箱。

对于式多重线性 PageRank 问题(1.1)中的随机张量 P，我们通过以下几种方式获取。

例 1：运用 MATLAB 中的 *rand* 函数，生成若干个 3 阶 n 维随机张量 P。在本例中，我们给出了当 $n = 200, 400$ 两种情况，数值实验结果如下：

小结：从表 1 及图 1 可以看出，当维数 $n = 200$ 或 $n = 400$，$\alpha = 0.95$ 时，所用的三种算法能够有效求解重线性 PageRank 问题。

Table 1. Experimental results of Example 1
表 1. 例 1 实验结果

n	Method	CPU	IT	err
200	NK	12.6939	560	8.82e−11
	NRK	11.2142	500	7.41e−11
	NURK	11.8904	517	7.87e−11
400	NK	179.3317	1030	7.62e−11
	NRK	158.3353	909	4.34e−11
	NURK	143.4701	821	9.95e−11

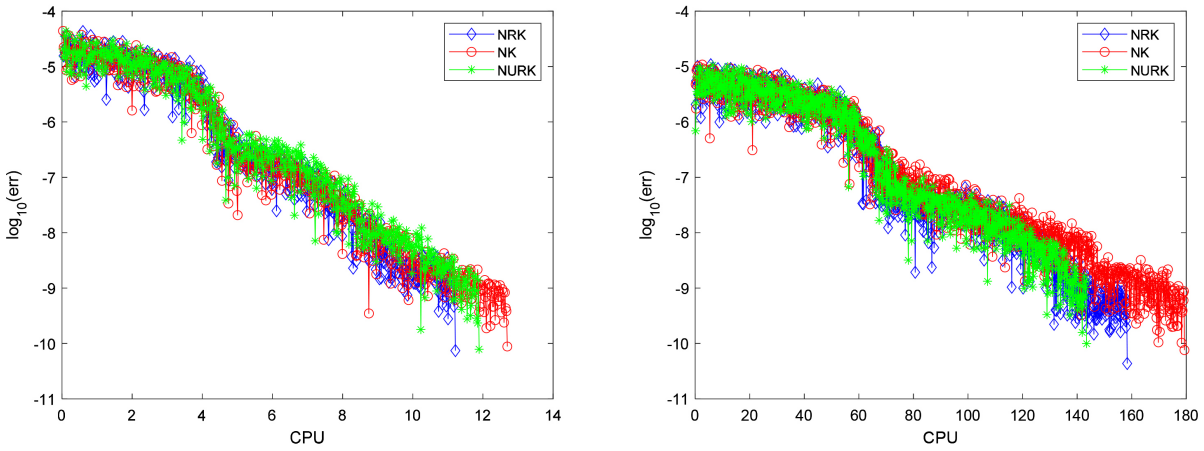

Figure 1. The relationship between CPU and err when n = 200 (left) and n = 400 (right)
图 1. n = 200 (左)及 n = 400 (右)时 CPU 与 err 的关系

例 2：设 t 表示张量零元素密度($t = 0$ 或 $t = 1$ 时，P 分别是正张量或零张量)，运用 MATLAB 中的 *randsample* 函数生成具有不同密度的稀疏张量 P。本例中取 $n = 300$，$t = 0.3$ 或 $t = 0.5$，实验结果如下：

小结：从表 2 及图 2 可以看出，当维数 $n = 300$，$t = 0.3$ 或 $t = 0.5$，$\alpha = 0.95$ 时，所用的三种算法能够有效求解重线性 PageRank 问题。

Table 2. Experimental results of Example 2
表 2. 例 2 实验结果

n	t	Method	CPU	IT	err
300	0.3	NK	66.3524	856	9.50e−11
		NRK	60.0733	775	4.99e−11
		NURK	64.1696	829	9.05e−11
	0.5	NK	56.0994	724	5.11e−11
		NRK	55.7386	722	6.82e−11
		NURK	62.3506	805	7.25e−11

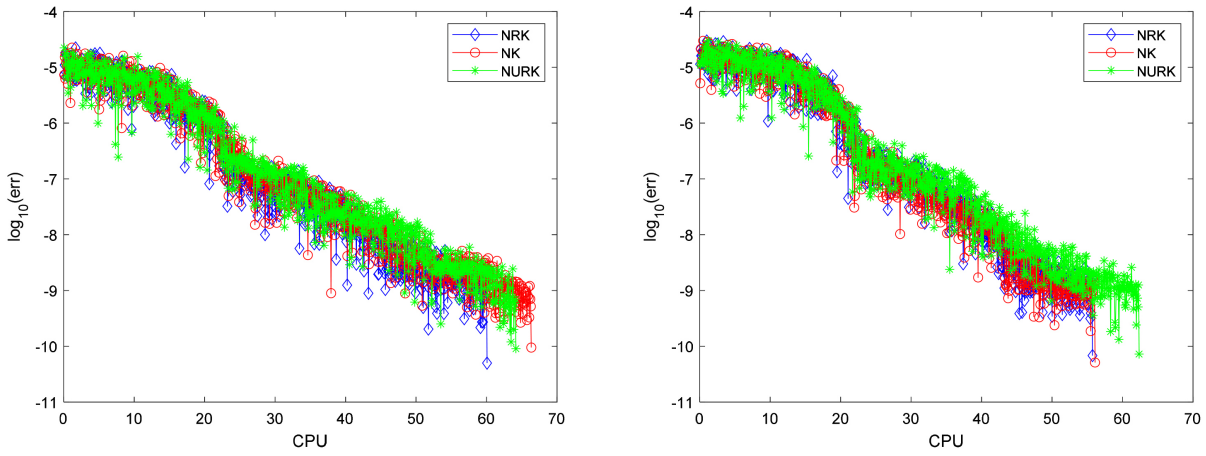

Figure 2. The relationship between CPU and err when t = 0.3 (left) and t = 0.5 (right)
图 2. t = 0.3 (左)及 t = 0.5 (右)时 CPU 与 err 的关系

例 3：[6]令 Γ 表示具有节点集 $V=\langle n\rangle=D\bigcup P$ 的有向图，其中 D 和 P 分别表示悬挂节点和两两连通节点的集合，n_p 表示子集 P 中的节点数，即 $P=\{1,2,\cdots,n_p\}$，那么我们可以构造非负张量 $\mathcal{A}$：

$$a_{i_1i_2\cdots i_l}=\begin{cases}a_{i_1i_2\cdots i_l}, & i_k\neq i_{k+1},i_1,i_k,i_l\in P,k=1,\cdots,l-1\\ 0, & i_1\neq i_2\,(i_1\in D),i_k\neq i_{k+1},i_k\in P,k=2,\cdots,l-1,\\ 1/n & \text{else}\end{cases}$$

其中 $a_{i_1i_2\cdots i_l}$ 从 $(0,1)$ 之间随机选取，规范化 $p_{i_1i_2\cdots i_l}=\dfrac{a_{i_1i_2\cdots i_l}}{\sum_{i_1=1}^{n}a_{i_1i_2\cdots i_l}}$ 得到张量 $P=p_{i_1i_2\cdots i_l}$。

在本例中，我们给出了当 $n=200$，$n_p=150$ 以及当 $n=400$，$n_p=300$ 的情况，数值实验结果如下表。

小结：从表 3 及图 3 可以看出，当 $n=200$，$n_p=150$ 或 $n=400$，$n_p=300$，$\alpha=0.95$ 时，所用的三种算法能够有效求解重线性 PageRank 问题。

Table 3. Experimental results of Example 3
表 3. 例 3 实验结果

n	n_p	Method	CPU	IT	err
200	150	NK	40.4712	1744	9.79e−11
		NRK	40.1985	1750	6.10e−11
		NURK	42.2552	1818	9.35e−11

续表

		NK	565.8186	3253	9.86e−11
400	300	NRK	611.5724	3393	6.44e−11
		NURK	560.7826	3313	9.60e−11

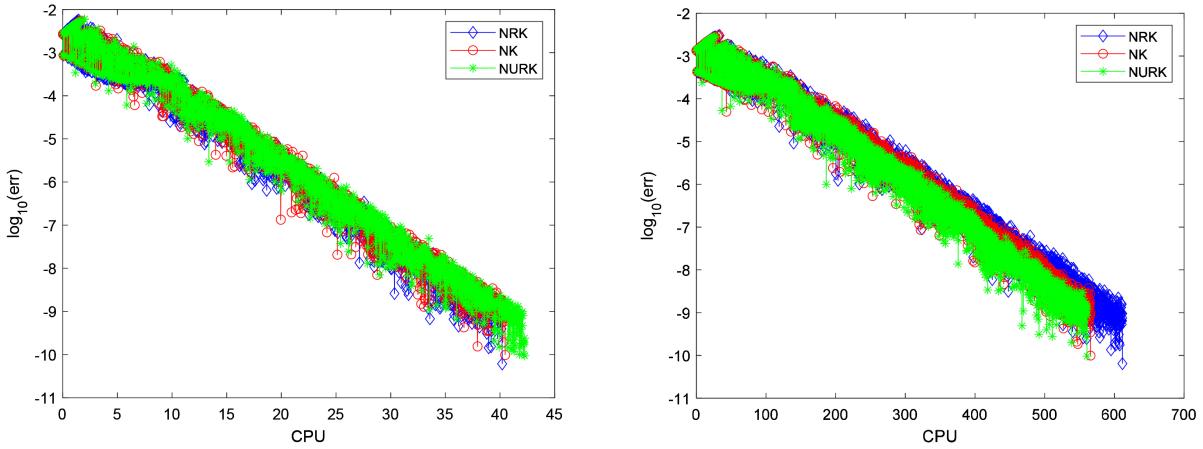

Figure 3. The relationship between CPU and err when n = 200 (left) and n = 400 (right)
图 3. n = 200 (左)及 n = 400 (右)时 CPU 与 err 的关系

例 4：[11]选用真实数据集 Edinburgh，阶数 $m=3$，维数 $n=1645$，该张量中非零元素个数为 4292。对于一个矩阵 $B\in\mathbb{R}^{n\times q}$ 以及向量 $e_k\in\mathbb{R}^k$，$dangling(B)$ 定义如下：

$$dangling(B)=e_q^{\mathrm{T}}-e_n^{\mathrm{T}}B\ .$$

我们通过对真实数据集的整理以获得三阶随机张量 P 的模 1 展开矩阵：

$$P_{(1)}=\omega\left[S+vdangling(S)\right]+(1-\omega)\left[M+vdangling(M)\right]\otimes e_n^{\mathrm{T}}.$$

取 $\omega=0.6$，通过验证，我们发现对于真实数据集，非线性随机 Kaczmarz 算法也能有效求解，如图 4。因此，无论是对于虚拟数据集还是真实数据集，非线性随机 Kaczmarz 算法可有效求解重线性 PageRank 问题。

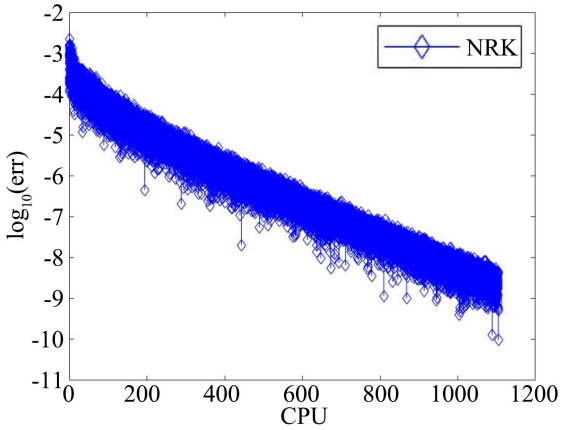

Figure 4. Edinburgh
图 4. Edinburgh

6. 结语

数值实验表明，用非线性 Kaczmarz 方法、非线性随机 Kaczmarz 方法以及非线性均匀随机 Kaczmarz 方法求解重线性 PageRank 问题，整体上非线性随机 Kaczmarz 方法的效果要好一些，达到停止准则时所需的时间较少。实验结果证实了利用随机算法求解算多重线性 PageRank 问题的有效性。

参考文献

[1] Page, L. (1999) The Page Rank Citation Ranking: Bringing Order to the Web. Technical Report.

[2] Gleich, D.F., Lim, L. and Yu, Y. (2015) Multilinear Pagerank. *SIAM Journal on Matrix Analysis and Applications*, **36**, 1507-1541. https://doi.org/10.1137/140985160

[3] Bucci, A. and Poloni, F. (2022) A Continuation Method for Computing the Multilinear PageRank. *Numerical Linear Algebra with Applications*, **29**, e2432. https://doi.org/10.1002/nla.2432

[4] Meini, B. and Poloni, F. (2018) Perron-Based Algorithms for the Multilinear PageRank. *Numerical Linear Algebra with Applications*, **25**, e2177. https://doi.org/10.1002/nla.2177

[5] Cipolla, S., Redivo-Zaglia, M. and Tudisco, F. (2020) Extrapolation Methods for Fixed-point Multilinear PageRank Computations. *Numerical Linear Algebra with Applications*, **27**, e2280. https://doi.org/10.1002/nla.2280

[6] Boubekraoui, M., Bentbib, A.H. and Jbilou, K. (2023) Vector Aitken Extrapolation Method for Multilinear PageRank Computations. *Journal of Applied Mathematics and Computing*, **69**, 1145-1172. https://doi.org/10.1007/s12190-022-01786-z

[7] Lai, F.Q., Li, W., Peng, X.F. and Chen, Y.N. (2023) Anderson Accelerated Fixed-Point Iteration for Multilinear PageRank. *Numerical Linear Algebra with Applications*, **30**, e2499. https://doi.org/10.1002/nla.2499

[8] Zhou, W.S., Wen, C., Shen, L.Z., *et al*. (2025) The MFPIO Iteration and the FPMPE Method for Multilinear PageRank Computations. *Journal of Computational and Applied Mathematics*, **454**, Article 116192. https://doi.org/10.1016/j.cam.2024.116192

[9] Bentbib, A.H., Boubekraoui, M. and Jbilou, K. (2024) Extrapolation Methods for Multilinear PageRank. *Numerical Algorithms*, **98**, 1013-1043.

[10] Wang, Q.F., Li, W.G., Bao, W,D., *et al*. (2022) Nonlinear Kaczmarz Algorithms and Their Convergence. *Journal of Computational and Applied Mathematics*, **399**, Article 113720. https://doi.org/10.1016/j.cam.2021.113720

[11] Grindrod, P. and Lee, T.E. (2016) Comparison of Social Structures within Cities of Very Different Sizes. *Royal Society Open Science*, **3**, Article 150526. https://doi.org/10.1098/rsos.150526