

Article

Not peer-reviewed version

A Parameterized Multisplitting Iterative Method for Solving the PageRank Problem

[Yajun Xie](#) , Lihua Hu , [Changfeng Ma](#) *

Posted Date: 27 June 2023

doi: 10.20944/preprints202306.1917.v1

Keywords: PageRank; inner-outer iterations; multi-parameter iteration; inner sub-systems



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

A Parameterized Multisplitting Iterative Method for Solving the PageRank Problem

Yajun Xie ^{1,†}, Lihua Hu ^{1,†}, Changfeng Ma ^{2,*}

¹ Department of Mathematics and Physics, Nanchang Normal College of Applied Technology, Nanchang 330108, China; xyj@fzfu.edu.cn.

² School of Big Data, Fuzhou University of International Studies and Trade, Fuzhou 350202, China; xyj@fzfu.edu.cn

* Correspondence: mcf@fzfu.edu.cn

† These authors contributed equally to this work.

Abstract: In this paper, a new multi parameter iterative algorithm is proposed to address the PageRank problem based on the multi-splitting iteration method described by Gu et al [8]. The proposed method in each iteration needs to solve two linear subsystems by splitting the coefficient matrix, therefore, we consider inner and outer iteration to find the approximate solutions of these linear subsystems. It can be shown that the iterative sequence generated by the multi parameter iterative algorithm finally converges to the PageRank vector when the parameters satisfy the certain conditions. Numerical experiments show that the proposed algorithm has better convergence and numerical stability than the existing algorithms.

Keywords: PageRank; inner-outer iterations; multi-parameter iteration; inner sub-systems

MSC: 65F10

1. Introduction

Consider the following linear equation systems:

$$\mathcal{A}x = x, \quad (1.1)$$

with \mathcal{A} is a convex combination Google matrix composed of matrix P and matrix E , and $\mathcal{A} = \alpha P + (1 - \alpha)E$, where $\alpha \in (0, 1)$ denotes the damping factor that determines the weight given to the web link graph, and $E = ve^T$, $e = (1, 1, \dots, 1)^T \in R^n$, and $v = \frac{e}{n}$ is a personalization vector or a teleportation vector. n is the dimension of P , and x is our desired eigenvector.

The system of linear equations in (1.1) above is what we refer to as the PageRank problem. Google's PageRank algorithm has grown to be one of the most well-known algorithms in online search engines thanks to the rapid development of the internet, link analysis method called PageRank is used to rank online pages and assess their significance in relation to the link structure of Web, calculating the primary eigenvectors of the Google matrix, forms the basis of the PageRank algorithm. Although Google's exact ranking technology and calculation techniques have gradually improved, the PageRank problem is still a major concern and has recently gained a lot of attention in the world of scientific and engineering computation.

To solve the PageRank problem, the power method is easy to calculate and the most classical algorithm, while all other eigenvalues of matrix \mathcal{A} aside from the principal eigenvalues are simply scalar times the corresponding eigenvalues of matrix P . As a result, the power approach converges very slowly when the primary eigenvalue of matrix \mathcal{A} is closely related to other eigenvalues, or when the damping factor is close to 1. The power method is not the ideal way to solve this problem, but a quicker and more logical way to solve the principal eigenvectors of the Google matrix is required to speed up the calculation of PageRank. The network graph is extremely large, with 1 billion or even 10 billion web page nodes. Additionally, a good search algorithm should minimize the lag time,

which is the time from the search target proposed to the search result feedback to the web browser. In recent years, numerous researchers have proposed various methods to speed up the calculation of PageRank, among them, the design of power function method and its variant method for accelerating the solution of PageRank problem are favored by many researchers. For instance, Gleich et al. [4] proposed an inner outer iteration method combined with Richardson iteration, in which each iteration needs to solve a linear system whose algebraic structure is similar to the original system; Gu and Xie [5] proposed the PIO iteration algorithm, which combines the power method and the inner-outer iteration method, after that, Ma et al. [11] suggested a relaxed two-step splitting iteration strategy to address the PageRank problem based on [4] and [5], adding a new relaxation parameter; Gu et al. [8] introduced a two parameter iteration approach based on multiplicative splitting iteration in order to increase the possibility of optimizing the iterative process; based on the iteration framework [7] and relaxed two-step splitting (RTSS) iteration method [11], Two relaxed iteration techniques are presented by Tian et al. [12] for resolving the PageRank issue. Additionally, the PageRank problem can be solved by using Krylov subspace methods, which is a problem of solving linear equations. For instance, Wu and Wei propose a hybrid algorithm, power-Arnoldi algorithm [14], which combines its power technique and thick restart Arnoldi algorithm; as well as the Arnoldi-extrapolation method [26] and speeding the Arnoldi-type algorithm [23]. We cite [7–29] for a more in-depth theoretical study.

The structure of this essay is as follows: we briefly introduce the inner-outer iterative PageRank problem techniques in Section 2. In Section 3, we first examine the theoretical foundations of the multiplicative splitting iterative method before introducing our brand-new approach, the parameterized MSI iteration method. Section 4 reports on numerical testing and comparisons. Finally, Section 5 provides a few succinct closing notes.

2. The inner-outer method

First, we provide a brief summary of the methodological inside-out iteration procedure proposed by Gleich et al. [4], for computing the inside-out iteration of PageRank. It is clear that linear systems can be used to rewrite the eigenvector problem (1.1).

$$(I - \alpha P)x = (1 - \alpha)v, \quad (2.1)$$

since $e^T x = 1$.

We observe that when the damping vector is small, it is simpler to solve the PageRank problem, Gleich et al. defined the outer iteration with a smaller damping factor β ($0 < \beta < \alpha$), rather than immediately resolving the equation (1.1). Therefore, the equations below are used to rewrite the linear system (1.1).

$$(I - \beta P)x = (\alpha - \beta)Px + (1 - \alpha)v. \quad (2.2)$$

So the stationary outer iteration scheme

$$(I - \beta P)x^{(k+1)} = (\alpha - \beta)Px^{(k)} + (1 - \alpha)v, \quad k = 0, 1, 2, \dots, \quad (2.3)$$

For computing $x_{(k+1)}$, define the inner linear system as

$$(I - \beta P)y = f, \quad (2.4)$$

where $f = (\alpha - \beta)Px^{(k)} + (1 - \alpha)v$, and compute $x^{(k+1)}$ via the Richardson inner iteration

$$y^{(j+1)} = \beta Py^{(j)} + (\alpha - \beta)Px^{(k)} + (1 - \alpha)v, \quad j = 0, 1, 2, \dots, l - 1, \quad (2.5)$$

where $y^{(0)} = x^{(k)}$, The l -th step inner solution $y^{(l)}$ is assigned to be the next new $x^{(k+1)}$. The stopping criteria are given as follows, The outer iteration (2.1) terminates if

$$\|(1 - \alpha)v - (I - \alpha P)x^{(k+1)}\|_2 < \tau, \quad (2.6)$$

while the inner iteration (2.5) terminates if

$$\|f - (I - \beta P)y^{(j+1)}\| < \eta, \quad j = 0, 1, 2, \dots, l - 1, \quad (2.7)$$

where η and τ are the inner and outer tolerances respectively.

The MSI iteration method

Gu et al. suggested the MSI approach in [8] to expedite the PageRank vector calculation. Here is a quick overview of the MSI approach, the MSI approach entails writing $I - \alpha P$ as

$$I - \alpha P = (I - \beta_1 P) - (\alpha - \beta_1)P = (I - \beta_2 P) - (\alpha - \beta_2)P,$$

here $0 < \beta_1 < \alpha$, $0 < \beta_2 < \alpha$, given an initial vector $x^{(0)}$, for $k = 0, 1, 2, \dots$, perform the following two-step iteration

$$\begin{cases} (I - \beta_1 P)u^{(k+1)} = (\alpha - \beta_1)Px^{(k)} + (1 - \alpha)v, \\ (I - \beta_2 P)x^{(k+1)} = (\alpha - \beta_2)Pu^{(k+1)} + (1 - \alpha)v. \end{cases} \quad (2.8)$$

until the sequence $\{x^{(k)}\}$ converges to the exact solution x^* .

Theorem 2.1 ([8]). Let α be the damping factor in the PageRank linear system, and let $M_i = I - \beta_i P$, $N_i = (\alpha - \beta_i)P$, ($i = 1, 2$) are the two splittings of the matrix $I - \alpha P$. Then the iterative matrix $\tilde{H}_{MSI}(\beta_1, \beta_2)$ of the MSI method for PageRank computation is given by

$$\tilde{H}_{MSI}(\beta_1, \beta_2) = (I - \beta_2 P)^{-1}(\alpha - \beta_2)P(I - \beta_1 P)^{-1}(\alpha - \beta_1)P, \quad (2.9)$$

and its spectral radius $\rho(\tilde{H}_{MSI}(\beta_1, \beta_2))$ is bounded by

$$\sigma(\beta_1, \beta_2) \equiv \frac{(\alpha - \beta_2)(\alpha - \beta_1)}{(1 - \beta_2)(1 - \beta_1)}, \quad (2.10)$$

therefore, it holds that

$$\rho(\tilde{H}_{MSI}(\beta_1, \beta_2)) \leq \sigma(\beta_1, \beta_2) < 1, \quad \forall 0 \leq \beta_1 < \alpha, 0 \leq \beta_2 < \alpha. \quad (2.11)$$

the multiplicative splitting iteration method for PageRank computation converges to the unique solution $x^* \in \mathbb{C}^n$ of the linear system of equations.

3. The parameterized MSI iteration method

The PageRank problem model is presented in this section, and it illustrates how the problem can be solved more simply by choosing a smaller damping factor α . We introduce a parameter ω based on the MSI method in order to further control the range of α , reduce the spectral radius, and speed up convergence, this results in a new iterative algorithm, denoted as the PMSI method below, that is described as follows.

The PMSI iteration method

$$\begin{cases} (I - \beta_1 P)u^{(k+1)} = (\omega\alpha - \beta_1)Px^{(k)} + (1 - \omega)x^{(k)} + \omega(1 - \alpha)v, \\ (I - \beta_2 P)x^{(k+1)} = (\omega\alpha - \beta_2)Pu^{(k+1)} + (1 - \omega)u^{(k+1)} + \omega(1 - \alpha)v. \end{cases} \quad (3.1)$$

with $\omega > 0$ and $0 < \beta_1 < \alpha < 1, 0 < \beta_2 < \alpha < 1$. If $\omega = 1$, then the PMSI iteration method becomes the MSI iteration method.

Algorithm 1 PMSI method

Input: Parameters $P, q, v, \alpha, \tau, \eta$;
Output: x

```

1:  $x \leftarrow v, y \leftarrow Px$ 
2: if  $\|(1 - \alpha)v + \alpha y - x\|_1 \geq \tau$  then
3:    $f_1 = (\omega\alpha - \beta_1)y + (1 - \omega)x + \omega(1 - \alpha)v$ ;
4:   repeat
5:      $x = \beta_1 y + f_1$ ;
6:      $y = Px$ ;
7:   until  $\|f_1 + \alpha y - x\|_1 < \eta$ 
8:    $f_2 = (\omega\alpha - \beta_2)y + (1 - \omega)x + \omega(1 - \alpha)v$ ;
9:   repeat
10:     $x = \beta_2 y + f_2$ ;
11:     $y = Px$ ;
12:   until  $\|f_2 + \alpha y - x\|_1 < \eta$ 
13:    $x = \beta_2 y + f_2$ ;
14:    $y = Px$ ;
15: end if
16:  $x = \alpha y + (1 - \alpha)v$ .
```

Remark 3.1. The computational cost of the PMSI iteration approach is somewhat higher than that of (2.8) since it simply requires an additional saxpy operation, $(1 - \omega)u^{(k+1)}$ Vector addition and the price $(\omega\alpha - \beta_1)Pu^{(k+1)} + (1 - \omega)u^{(k+1)}$ each iteration of with $o(n)$ flops.

In the sequel, we will analyze the convergence property of the parameterized MSI iteration method.

Lemma 3.1 ([8]). Let $\mathcal{A} \in C^{n \times n}$, $\mathcal{A} = M_i - N_i$ ($i = 1, 2$) be two splittings of the matrix \mathcal{A} , and let $x^{(0)} \in C^n$ be a given initial vector. If $x^{(k)}$ is a two-step iteration sequence

$$\begin{cases} M_1 x^{(k+\frac{1}{2})} = N_1 x^{(k)} + b, \\ M_2 x^{(k+1)} = N_2 x^{(k+\frac{1}{2})} + b, \end{cases} \quad (3.2)$$

then

$$x_{k+1} = M_2^{-1} N_2 M_1^{-1} N_1 x_k + M_2^{-1} (I + N_2 M_1^{-1}) b, k = 0, 1, 2, \dots \quad (3.3)$$

Moreover, if the spectral radius $\rho(M_2^{-1} N_2 M_1^{-1} N_1)$ is less than 1, then the iteration sequence $x^{(k)}$ converges to the unique solution $x^* \in C^{n \times n}$ of the system of linear equation (2.1) for all initial vectors $x^{(0)} \in C^n$.

The multiplicative splitting iteration method for (2.1) is obviously related with the splitting of the coefficient matrix $I - \alpha P$, and we will subsequently demonstrate that there exists a plausible convergent domain of two-parameters for the parameterized method.

$$\begin{aligned} I - \alpha P &= M_i - N_i \quad (i = 1, 2), \\ M_1 &= I - \beta_1 P, N_1 = (\omega\alpha - \beta_1)P + (1 - \omega)I, \\ M_2 &= I - \beta_2 P, N_2 = (\omega\alpha - \beta_2)P + (1 - \omega)I. \end{aligned}$$

according to (3.1), The two-step iterative matrix corresponding to the multiplication split iterative method is as follows

$$\tilde{G}_{PMSI}(\beta_1, \beta_2) = M_2^{-1}N_2M_1^{-1}N_1 = (I - \beta_2P)^{-1}((\omega\alpha - \beta_2)P + (1 - \omega)I) \\ (I - \beta_1P)^{-1}((\omega\alpha - \beta_1)P + (1 - \omega)I), \quad (3.4)$$

Now, we examine the convergence property of the multiplicative-splitting iterative method. By applying Lemma 3.1, we can obtain the following main theorem..

Theorem 3.1. Let α be the damping factor in the PageRank linear system, and let $M_i = I - \beta_iP$, $N_i = (\alpha - \beta_i)P + (1 - \omega)$ ($i = 1, 2$) are the two splittings of the matrix $I - \alpha P$. Then the iterative matrix $\tilde{G}_{PMSI}(\beta_1, \beta_2)$ of the PMSI method for PageRank computation is given by

$$\tilde{G}_{PMSI}(\beta_1, \beta_2) = (I - \beta_2P)^{-1}((\omega\alpha - \beta_2)P + (1 - \omega)I) \\ (I - \beta_1P)^{-1}((\omega\alpha - \beta_1)P + (1 - \omega)I), \quad (3.5)$$

and its spectral radius $\rho(\tilde{G}_{PMSI}(\beta_1, \beta_2))$ is bounded by

$$\psi(\beta_1, \beta_2) \equiv 1 - \frac{(1 - \alpha)\omega[2 + \omega(1 - \alpha) - \beta_2 - \beta_1]}{(1 - \beta_1)(1 - \beta_2)}, \quad (3.6)$$

therefore, it holds that

$$\rho(\tilde{G}_{PMSI}(\beta_1, \beta_2)) \leq \psi(\beta_1, \beta_2) < 1, \forall 0 \leq \beta_1 < \alpha, 0 \leq \beta_2 < \alpha. \quad (3.7)$$

the multiplicative splitting iteration method for PageRank computation converges to the unique solution $x^* \in C^n$ of the linear system of equations.

Proof. From Lemma 3.1 we can obtain the iterative matrix of the PMSI method for PageRank computation Eq. (3.5).

Let $\beta = \min\{\beta_1, \beta_2\}$, since $e^TP = e^T$, $\frac{\beta}{\alpha} \leq \omega \leq 1$, then the matrix $(\omega\alpha - \beta_i)P + (1 - \omega)I$ ($i = 1, 2$) is a nonnegative matrix and the matrix \tilde{G}_k is also nonnegative.

In addition, from (3.5) it turns out that

$$e^T\tilde{G}_{PMSI}(\beta_1, \beta_2) = e^T[(I - \beta_2P)^{-1}((\omega\alpha - \beta_2)P + (1 - \omega)I) \\ (I - \beta_1P)^{-1}((\omega\alpha - \beta_1)P + (1 - \omega)I)], \quad (3.8)$$

if λ_i is an eigenvalue of P , The spectral radius of $\tilde{G}_{PMSI}(\beta_1, \beta_2)$ are

$$\rho(\tilde{G}_{PMSI}(\beta_1, \beta_2)) = \frac{((\omega\alpha - \beta_2) + (1 - \omega))((\omega\alpha - \beta_1) + (1 - \omega))}{(1 - \beta_1)(1 - \beta_2)}, \quad (3.9)$$

for

$$((\omega\alpha - \beta_2) + (1 - \omega))((\omega\alpha - \beta_1) + (1 - \omega)) \\ = (\omega(\alpha - 1) + 1 - \beta_2)(\omega(\alpha - 1) + 1 - \beta_1) \\ = [(\omega(\alpha - 1))^2 + \omega(\alpha - 1)(2 - \beta_2 - \beta_1)] + (1 - \beta_2)(1 - \beta_1) \\ = (1 - \beta_2)(1 - \beta_1) - (1 - \alpha)\omega[2 + \omega(1 - \alpha) - \beta_2 - \beta_1]. \quad (3.10)$$

Since $(2 + \omega(1 - \alpha) - \beta_2 - \beta_1) > 0$, combining the above relations (3.9) and (3.10), we can get

$$\rho(\tilde{G}_{PMSI}(\beta_1, \beta_2)) < 1 - \frac{(1 - \alpha)\omega[2 + \omega(1 - \alpha) - \beta_2 - \beta_1]}{(1 - \beta_1)(1 - \beta_2)} < 1. \quad (3.11)$$

So for any given constants β_1 and β_2 , $0 \leq \beta_1 < \alpha$, $0 \leq \beta_2 < \alpha$. The PMSI method converges to a unique solution to the linear system (2.1), the PageRank vector. \square

Since $0 < \beta_i < \alpha < 1$ ($i = 1, 2$), then $\frac{\beta_i}{\alpha} < \omega < 1$. Immediately, a comparison result is obtained for the parameterized MSI iteration method compared with the MSI iteration method.

Theorem 3.2. Let $0 < \beta_i < \alpha < 1$ ($i = 1, 2$), $\beta = \min\{\beta_1, \beta_2\}$, If $\frac{\beta}{\alpha} < \omega < 1$, then the parameterized MSI iteration method converges faster than the MSI iteration method.

Proof. From Eq (3.9), it follows that the spectral radius of the parameterized MSI iteration method is

$$\rho(\tilde{G}_{PMSI}(\beta_1, \beta_2)) = \frac{((\omega\alpha - \beta_2) + (1 - \omega))((\omega\alpha - \beta_1) + (1 - \omega))}{(1 - \beta_1)(1 - \beta_2)}. \quad (3.12)$$

Let $\omega = 1$ in (3.12), then we obtain the spectral radius of the MSI iteration method as follows:

$$\rho(\tilde{H}_{MSI}(\beta_1, \beta_2)) = \frac{(\alpha - \beta_2)(\alpha - \beta_1)}{(1 - \beta_1)(1 - \beta_2)}. \quad (3.13)$$

For $0 < \beta_i < \alpha < 1$, $\frac{\beta_i}{\alpha} < \omega < 1$ ($i = 1, 2$), from (3.12)-(3.13), it is clear that

$$\begin{aligned} \rho(\tilde{G}_{PMSI}(\beta_1, \beta_2)) &= \frac{((\omega\alpha - \beta_2) + (1 - \omega))((\omega\alpha - \beta_1) + (1 - \omega))}{(1 - \beta_1)(1 - \beta_2)} \\ &= \frac{(\omega\alpha - \beta_2)(\omega\alpha - \beta_1)(1 - \omega)^2}{(1 - \beta_1)(1 - \beta_2)} \\ &< \frac{(\omega\alpha - \beta_2)(\omega\alpha - \beta_1)}{(1 - \beta_1)(1 - \beta_2)} \\ &< \frac{(\alpha - \beta_2)(\alpha - \beta_1)}{(1 - \beta_1)(1 - \beta_2)} \\ &= \rho(\tilde{H}_{MSI}(\beta_1, \beta_2)). \end{aligned} \quad (3.14)$$

It is obvious that $\rho(\tilde{G}_{PMSI}(\beta_1, \beta_2)) < \rho(\tilde{H}_{MSI}(\beta_1, \beta_2))$, and the proof is completed. \square

Corollary 3.1. In the range of $\frac{\beta}{\alpha} < \omega < 1$, $\beta = \min\{\beta_1, \beta_2\}$, $0 < \beta_i < 1$ ($i = 1, 2$), when ω increases gradually within the value range, the smaller the iterative spectral radius of the PMSI algorithm, the faster the convergence speed.

Proof. According to Equation (3.9), we know that

$$\rho(\tilde{G}_{PMSI}(\beta_1, \beta_2)) = \frac{((\omega\alpha - \beta_2) + (1 - \omega))((\omega\alpha - \beta_1) + (1 - \omega))}{(1 - \beta_1)(1 - \beta_2)}.$$

Let

$$\begin{aligned} \tilde{f}(\omega) &= \frac{((\omega\alpha - \beta_2) + (1 - \omega))((\omega\alpha - \beta_1) + (1 - \omega))}{(1 - \beta_1)(1 - \beta_2)} \\ &= \frac{(\omega(\alpha - 1) + 1 - \beta_2)(\omega(\alpha - 1) + 1 - \beta_1)}{(1 - \beta_1)(1 - \beta_2)} \\ &= \frac{(\omega(\alpha - 1))^2 + \omega(\alpha - 1)(2 - \beta_1 - \beta_2) + (1 - \beta_1)(1 - \beta_2)}{(1 - \beta_1)(1 - \beta_2)}. \end{aligned}$$

It is easy one can obtain from above relation

$$\begin{aligned}\tilde{f}'(\omega) &= \frac{(2\omega(\alpha-1) + (\alpha-1)(2-\beta_1-\beta_2))((1-\beta_1)(1-\beta_2))}{((1-\beta_1)(1-\beta_2))^2} \\ &= \frac{2\omega(\alpha-1) + (\alpha-1)(2-\beta_1-\beta_2)}{(1-\beta_1)(1-\beta_2)} \\ &= \frac{(\alpha-1)(2\omega+2-\beta_1-\beta_2)}{(1-\beta_1)(1-\beta_2)}.\end{aligned}$$

since $\frac{\beta}{\alpha} < \omega < 1$, $0 < \beta_i < \alpha < 1$ ($i = 1, 2$), so we have $(2\omega+2-\beta_1-\beta_2) > 0$. we can get $\tilde{f}'(\omega) < 0$, we can see the PMSI iterative method may be more efficient when ω is large, the conclusion is proved. \square

4. Numerical results

In this section, we compare the performance of the parameterized multisplitting (PMSI) iteration method to that of the inner-outer (IO) and multi-splitting (MSI) iteration methods, respectively. On dual-core processing, numerical experiments are carried out in Matlab R2018a (2.30 GHz, 8GB RAM). Four iteration parameters, the number of matrix vectors (denoted as MV), the iteration step size (denoted as IT), the calculation time in seconds (denoted as CPU), and the relative residual (denoted as res(k)) are used to test these iterative approaches, defined

$$res(k) = \frac{\|r_k\|_2}{\|(1-\alpha)v\|_2}, \quad k = 0, 1, \dots$$

where $r_k = \frac{(1-\alpha)v}{(I-\alpha P)x_k}$.

Table 1 lists the properties of the test matrices P , where average non-zero refers to each row of non-zero elements, and

$$\rho_{den} = \frac{nnz}{n \times n} \times 100.$$

All test matrices can be downloaded from https://www.cise.ufl.edu/research/sparse/matrices/list_by_id.html. For the interest of fairness, we assume that the transfer vector $x^{(0)} = v^{(0)} = \frac{e}{n}$ ($e = (1, 1, \dots, 1)^T$) is the initial guess for each test matrix. In all numerical tests, the damping factors are assumed to be $\alpha = 0.98, 0.99, 0.995, 0.997, 0.998$. The residual specification $\eta = 0.01$, $\tau < 10^{-8}$ determines when all algorithms end.

Table 1. properties of test matrices

	Size	n_{nnz}	ρ_{den}
wb-cs-stanford	9914×9914	2 312 497	0.291×10^{-2}
amazon0312	400,727×400,727	3200,440	1.993×10^{-3}

Example 4.1. In this example, we compare the PMSI iteration method with the MSI iteration method. The test matrices are the wb-cs-stanford, and amazon0312 matrices, respectively. In order to verify the efficiency of the PMSI iteration method, we use

$$S_{PMSI} = \frac{CPU_{MSI} - CPU_{PMSI}}{CPU_{MSI}}$$

to describe the speedups of the PMSI iteration compared with the MSI iteration associated with CPU time.

The numerical outcomes of the MSI and PMSI iterative procedures, where $\omega = 0.9$ and $\beta_1 = 0.9$ and $\beta_2 = 0.8$, are displayed in Tables 2 and 3. Tables 2 and 3 shows that the PMSI iterative technique performs better

than the MSI iterative method in terms of IT, MV, and CPU time, especially for bigger α , such the $\alpha = 0.998$ in Tables 2 and 3. As can be observed, most S_{PMSI} values are more than 20%, sometimes even reaching 50%.

Table 2. Test results for the wb-cs-stanford matrix

α		IO	MSI	PMSI	S_{pt}
0.98	IT(MV)	536	270(541)	228(457)	14.90%
	CPU	0.1261	0.1268	0.1079	
0.99	IT(MV)	1096	537(1075)	417(835)	27.52%
	CPU	0.2151	0.2274	0.1648	
0.995	IT(MV)	2168	1095(2191)	962(1525)	22.96%
	CPU	0.7280	0.3819	0.2942	
0.997	IT(MV)	3577	1806(3613)	1213(2427)	26.93%
	CPU	0.6380	0.5954	0.4350	
0.998	IT(MV)	5450	2698(5397)	1663(3327)	32.37%
	CPU	0.9354	0.8669	0.5862	

Table 3. Test results for the amazon0312 matrix

α		IO	MSI	PMSI	S_{pt}
0.98	IT(MV)	367	178(357)	170(341)	3.54%
	CPU	7.3867	6.8073	6.5658	
0.99	IT(MV)	733	363(727)	292(585)	14.44%
	CPU	15.7108	14.1045	12.0670	
0.995	IT(MV)	1436	723(1447)	5110(1021)	27.69%
	CPU	30.1137	29.4107	21.2650	
0.997	IT(MV)	2507	1164(2329)	717(1435)	42.79%
	CPU	30.1137	48.6659	27.8410	
0.998	IT(MV)	3630	1863(3727)	911(1823)	50.00%
	CPU	90.7846	75.3888	37.6927	

Example 4.2. With the test matrices being the wb-cs-stanford and amazon0312 matrix with various ω parameters, we will further examine the convergence performance of the PMSI iterative method in this example. We have set the ω value range to 0.4 – 0.9. The numerical outcomes are shown in Figure 3, where $\beta_1 = 0.9$ and $\beta_2 = 0.8$. According to the findings, the number of repetitions constantly lowers as ω rises. Because of this, we used $\omega = 0.9$ in our studies, which is consistent with the finding in Corollary 3.1.

Example 4.3. Theorem 3.1 states that the PMSI method converges for any value of β_1 and β_2 , satisfying the conditions of $0 \leq \beta_1 < \alpha$ and $0 \leq \beta_2 < \alpha$. This is what we take into consideration in this example. For two matrices, wb-cs-stanford and amazon0312, Tables 4 and 5 display the number of iterations of the PMSI approach. The values of β_1 and β_2 change from 0.1 to 0.9 and 0.1 to 0.9, respectively, when $\alpha = 0.99$. From Tables 4 and 5, it can be inferred that, once one of the parameters β_1 and β_2 is determined, the number of iteration steps typically decreases first before increasing as more parameters are added. For instance, in Table 3, β_2 increased from β_2 of 0.8 to β_2 of 0.9. The number of iteration steps first declines, and then β_1 takes 0.1 to 0.5, and the number of iteration steps continues to rise. Finding an explicit link between β_1 and β_2 , or the ideal β_1 and β_2 , for the universal PageRank matrix, is quite difficult. Our considerable experience has shown that selecting $\beta_1=0.9$ and $\beta_2=0.8$ usually results in good performance. For this reason, in our studies, we used $\beta_1=0.9$ and $\beta_2=0.8$ in the PMSI approach.

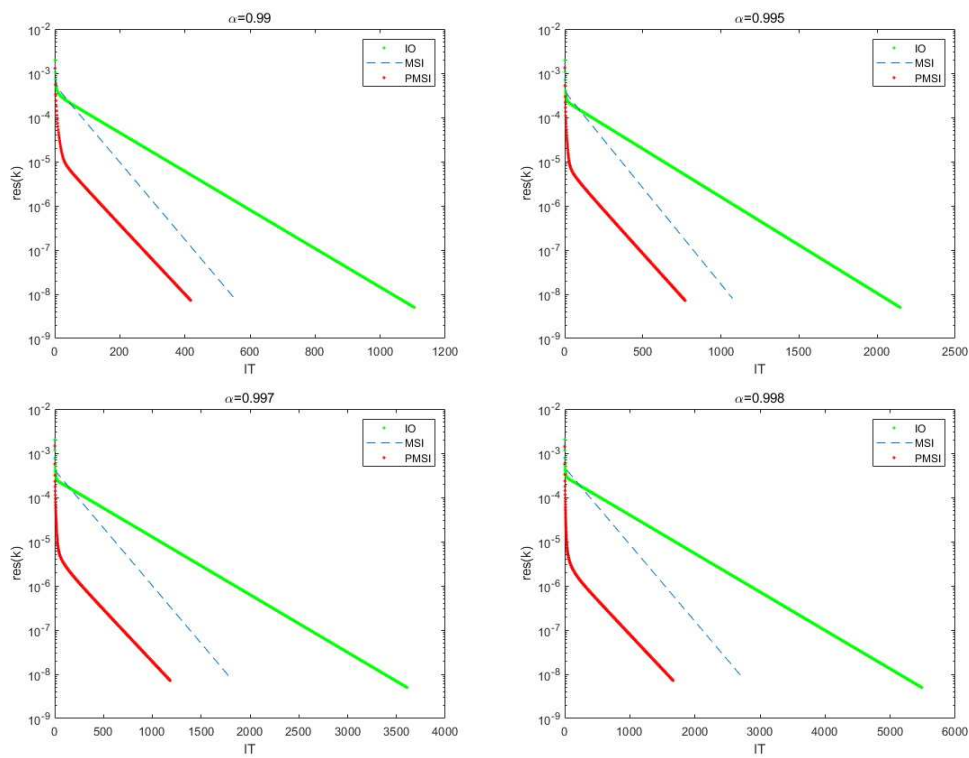


Figure 1. Convergence effect of three algorithms for wb-cs-stanford matrix, $\tau = 10^{-8}$

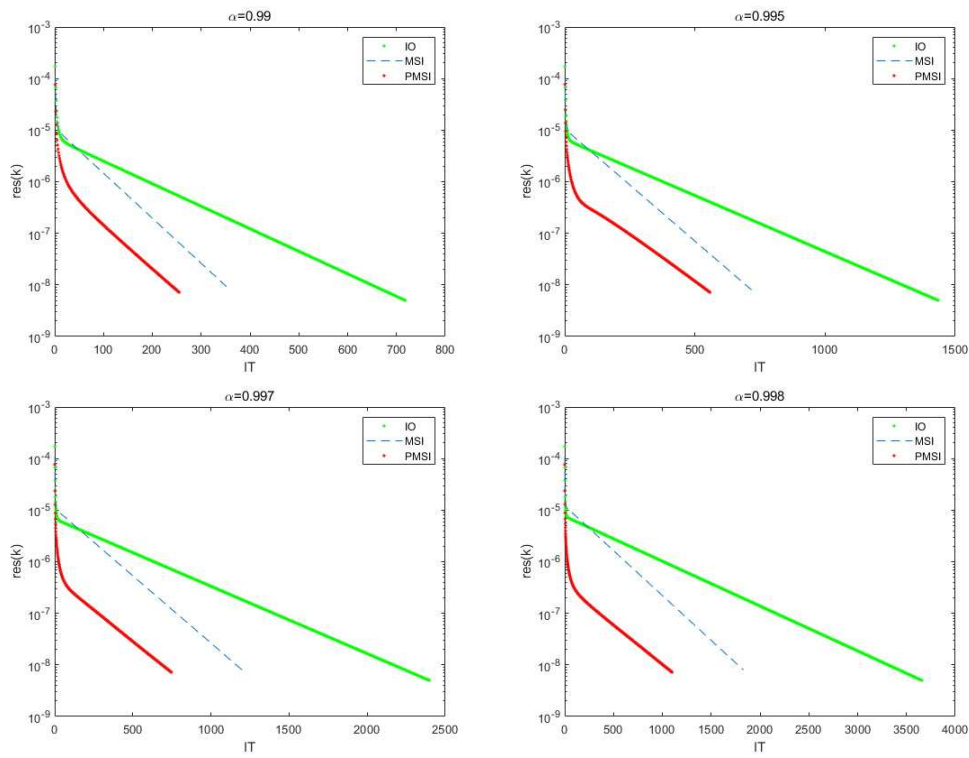


Figure 2. Convergence effect of three algorithms for amazon0312 matrix, $\tau = 10^{-8}$

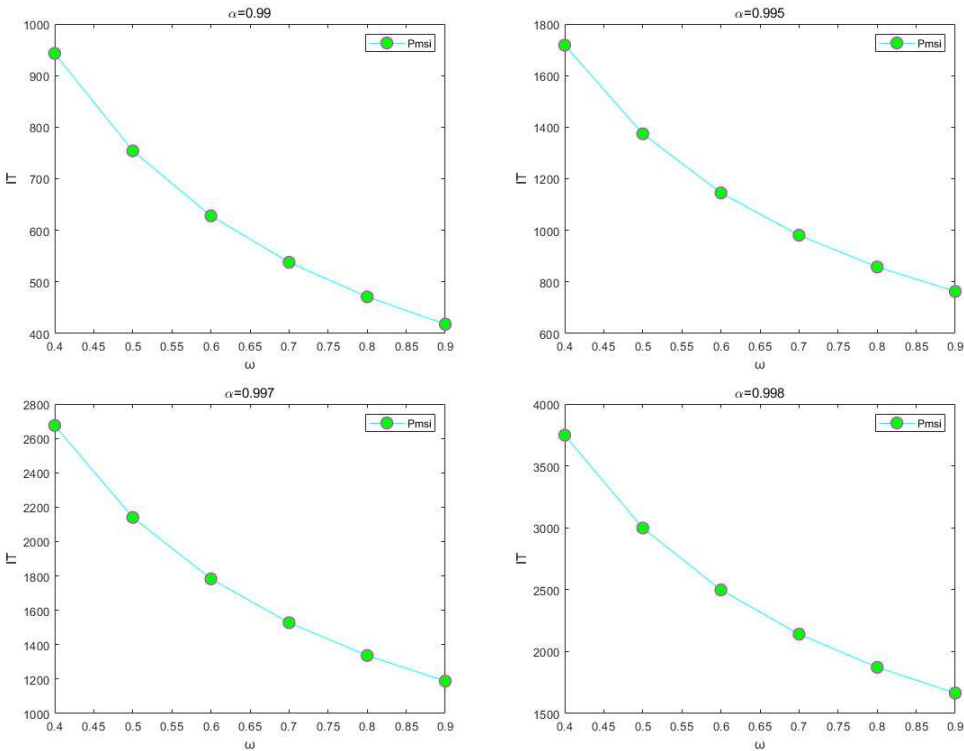


Figure 3. Numerical results for the wb-cs-stanford matrix in Example 4.2

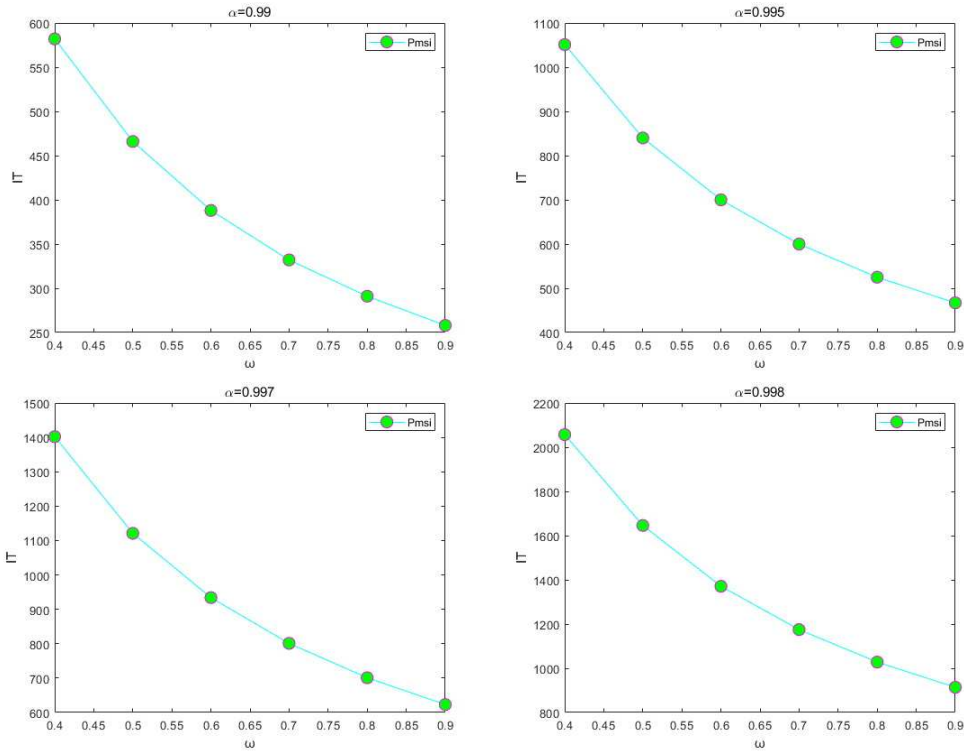


Figure 4. Numerical results for the amazon0312 matrix in Example 4.2

Table 4. Numerical results for the wb-cs-stanford matrix in Example 4.3

$\beta_2 \backslash \beta_1$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	418(837)	410(821)	419(839)	419(839)	414(829)	422(845)	415(831)	417(835)	415(831)
0.3	416(833)	415(831)	418(837)	417(835)	420(841)	415(831)	426(853)	422(845)	417(835)
0.5	412(825)	420(841)	415(831)	416(833)	425(851)	411(823)	426(853)	424(569)	418(837)
0.7	418(837)	414(829)	417(837)	414(829)	419(839)	416(833)	412(825)	419(839)	417(835)
0.9	420(841)	422(845)	428(857)	420(841)	424(849)	422(845)	416(833)	417(835)	412(825)

Table 5. Numerical results for the amazon0312 matrix in Example 4.3

$\beta_2 \backslash \beta_1$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.1	293(587)	275(551)	262(525)	302(605)	284(569)	269(539)	275(551)	277(555)	262(525)
0.3	262(525)	265(531)	294(589)	262(525)	263(527)	303(607)	271(543)	269(539)	258(517)
0.5	281(563)	257(535)	269(539)	308(617)	256(513)	276(553)	303(607)	284(569)	278(557)
0.7	322(665)	275(551)	303(607)	255(511)	301(603)	272(545)	269(539)	314(629)	264(529)
0.9	259(517)	276(553)	274(549)	270(541)	251(503)	274(549)	272(545)	272(545)	255(511)

5. Conclusions

In this paper, in order to further improve the two-step splitting iterative method, we propose a parameterized multisplitting iterative method to solve the PageRank problem by introducing a relaxation parameter ω , when $\omega = 1$, the PMSI method reduces to the MSI method. Numerical experiments show that the sequence of iterations generated by the PMSI method converges to the PageRank vector when the parameters ω , β_1 , and β_2 satisfy specific requirements. The proposed method also has better convergence performance than IO and MSI methods. Since the new algorithms are parameter-dependent, how to obtain the optimal parameters in the general case remains to be studied.

References

1. N.-A. Langville, C.-D. Meyer, Google’s PageRank and beyond: the science of search engine rankings. Princeton University Press, Princeton, NJ, 2006. 195-224.
2. B. Philippe, Y. Saad, W.-J. Stewart, Numerical methods in Markov chain modeling, Oper. Res. 40 (1992), 1156-1179.
3. Golub, H. Gene, L. Van, F. Charles, Matrix computations, Johns Hopkins University Press. (1989). 569-784.
4. D.-F. Gleich, A.-P. Gray, C. Greif, T. Lau, An inner-outer iteration for computing PageRank, SIAM J. Sci. Comput. 32 (2010), 349-371.
5. C.-Q. Gu, F. Xie, K. Zhang, A two-step matrix splitting iteration for computing PageRank, J. Comput. Appl. Math. 278(2015) 19-28.
6. C. Wen, T.-Z. Huang, Z.-L. Shen, A note on the two-step matrix splitting iteration for computing PageRank, J. Comput. Appl. Math. 315 (2017), 87-97.
7. Z.-L. Tian, X.-J. Li, Z.-Y. Liu, A general multi-step matrix splitting iteration method for computing PageRank, Filomat 35 (2021), 679-706.
8. C.-Q. Gu, L. Wang, On the multi-splitting iteration method for computing PageRank, J. Comput. Appl. Math. 42 (2013), 479-490.
9. , M.-Y. Tian, Y. Zhang, Y.-D Wang, Z.-L. Tian, A general multi-splitting iteration method for computing PageRank, Comput. Appl. Math. 38 (2019), 60.
10. , X.-D. Chen, S.-Y. Li, A generalized two-step splitting iterative method modified with the multi-step power method for computing PageRank, (Chinese) J. Numer. Methods Comput. Appl. 39 (2018), 243-252.
11. Y.-J. Xie, C.-F. Ma, A relaxed two-step splitting iteration method for computing PageRank, J. Comput. Appl. Math. 37 (2018), 221-233.
12. Z.-L. Tian, Y. Zhang, J.-X. Wang, C.-Q. Gu, Several relaxed iteration methods for computing PageRank, J. Comput. Appl. Math. 388 (2021), 21.

13. Z-L Tian, X.-Y. Liu, Y.-D. Wang, P.-H. Wen, The modified matrix splitting iteration method for computing PageRank problem, *Filomat* 33 (2019), 725-740.
14. G. Wu, Y.-M. Wei, A power-Arnoldi algorithm for computing PageRank, *Numer. Linear Algebra Appl.* 14 (2007), 521-546.
15. C.-Q. Gu, Y. Nie, J.-B. Wang, Arnoldi-PIO algorithm for PageRank, (Chinese) *J. Shanghai Univ. Nat. Sci.* 23 (2017), 555-562.
16. Z-H. Qiu, C.-Q. Gu, A GMRES-RPIO algorithm for computing PageRank problem, *J. Numerical Mathematics a Journal of Chinese Universities*, 40 (2018), 331-345.
17. C.-Q. Gu, W.-W. Wang, An Arnoldi-MSI algorithm for computing PageRank problems, (Chinese) *Numer. Math. J. Chinese Univ.* 38 (2016), 257-268.
18. C.-Q. Gu, C.-C. Shao, A GMRES-in/out algorithm for computing PageRank problems, (Chinese) *J. Shanghai Univ. Nat. Sci.* 23 (2017), 179-184.
19. X.-M. Gu, S.-L. Lei, K. Zhang, Z.-L. Shen, C. Wen, B. Carpentieri, A Hessenberg-type algorithm for computing PageRank problems, *Numer. Algorithms.* 89 (2022), 1845-1863.
20. W.-K. X, X.-D. Chen, A Modified Multi-Splitting Iterative Method With the Restarted GMRES to Solve the PageRank Problem, (Chinese) *Appl Math and Mechanics.* 43 (2022), 330-340.
21. N. Huang, C.-F. Ma, Parallel multisplitting iteration methods based on M-splitting for the PageRank problem, *Appl. Math. Comput.* 271 (2015), 337-343.
22. G. Wu, Y. Zhang, Y.-M. Wei, Accelerating the Arnoldi-type algorithm for the PageRank problem and the ProteinRank problem, *J. Sci. Comput.* 57 (2013), 74-104.
23. B.-Y. Pu, T.-Z. Huang, C. Wen, A preconditioned and extrapolation-accelerated GMRES method for PageRank, *Appl. Math. Lett.* 37 (2014), 95-100.
24. X.-Y. Tan, A new extrapolation method for PageRank computations, *J. Comput. Appl. Math.* 313 (2017), 383-392.
25. C. Wen, Q.-Y. Hu, B.-Y. Pu, Y.-Y. Huang, Acceleration of an adaptive generalized Arnoldi method for computing PageRank, *AIMS Math.* 6 (2021), 893-907.
26. G. Wu, Y.-M. Wei, An Arnoldi-extrapolation algorithm for computing PageRank, *J. Comput. Appl. Math.* 234 (2010), 3196-3212.
27. P.-C. Guo, S.-C. Gao, X.-X. Guo, A modified Newton method for multilinear PageRank, *Taiwanese J. Math.* 22 (2018), 1161-1171.
28. B.-Y. Pu, C. Wen, Q.-Y. Hu, A multi-power and multi-splitting inner-outer iteration for PageRank computation, *Open Math.* 18 (2020), 1709-1718.
29. H.-F. Zhang, T.-Z. Huang, C. Wen, Z.-L. Shen, FOM accelerated by an extrapolation method for solving PageRank problems, *J. Comput. Appl. Math.* 296 (2016), 397-409.
30. C.-Q. Gu, G.-D. Ge, Two-splitting iteration method for computing higher-order PageRank, (Chinese) *Appl. Math. Comput.* 32 (2018), 581-587.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.