

# Shifted power-GMRES method accelerated by extrapolation for solving PageRank with multiple damping factors

Luca Lombardo

## Abstract

Starting from the seminal paper published by Brin and Page in 1998, the PageRank model has been extended to many fields far beyond search engine rankings, such as chemistry, biology, bioinformatics, social network analysis, to name a few. Due to the large dimension of PageRank problems, in the past decade or so, considerable research efforts have been devoted to their efficient solution especially for the difficult cases where the damping factors are close to 1. However, there exists few research work concerning about the solution of the case where several PageRank problems with the same network structure and various damping factors need to be solved. In this paper, we generalize the Power method to solving the PageRank problem with multiple damping factors. We demonstrate that the solution has almost the equative cost of solving the most difficult PageRank system of the sequence, and the residual vectors of the PageRank systems after running this method are collinear. Based upon these results, we develop a more efficient method that combines this Power method with the shifted GMRES method. For further accelerating the solving phase, we present a seed system choosing strategy combined with an extrapolation technique, and analyze their effect. Numerical experiments demonstrate the potential of the proposed iterative solver for accelerating realistic PageRank computations with multiple damping factors.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Overview of the classical PageRank problem . . . . .	3
<b>2</b>	<b>The shifted power method for PageRank computations</b>	<b>4</b>
2.1	The implementation of the shifted power method . . . . .	4

# 1 Introduction

The PageRank model was proposed by Google in a series of papers to evaluate accurately the most important web-pages from the World Wide Web matching a set of keywords entered by a user. Nowadays, the model is routinely adopted for the analysis of many scientific problems far beyond Internet applications, for example in computational chemistry, biology, bioinformatics, social network analysis, bibliometrics, software debugging and many others. For search engine rankings, the importance of web-pages is computed from the stationary probability vector of the random process of a web surfer who keeps visiting a large set of web-pages connected by hyperlinks. The link structure of the World Wide Web is represented by a directed graph, the so-called web link graph, and its corresponding adjacency matrix  $G \in \mathbb{N}^{n \times n}$  where  $n$  denotes the number of pages and  $G_{ij}$  is nonzero (being 1) only if the  $j$ th page has a hyperlink pointing to the  $i$ th page. The transition probability matrix  $P \in \mathbb{R}^{n \times n}$  of the random process has entries

$$P(i, j) = \begin{cases} \frac{1}{\sum_{k=1}^n G_{kj}} & \text{if } G_{ij} = 0 \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

To ensure that the random process has a unique stationary distribution and it will not stagnate, the transition matrix  $P$  is usually modified to be an irreducible stochastic matrix  $A$  (called the Google matrix) as follows

$$A = \alpha \tilde{P} + (1 - \alpha)ve^T \quad (2)$$

In 2 we define  $\tilde{P} = P + vd^T$  where  $d \in \mathbb{N}^{n \times 1}$  is a binary vector tracing the indices of the damping web pages with no hyperlinks, i.e.,  $d(i) = 1$  if the  $i$ th page has no hyperlink,  $v \in \mathbb{R}^{n \times n}$  is a probability vector,  $e = [1, 1, \dots, 1]^T$  and  $0 < \alpha < 1$ , the so-called damping factor that represents the probability in the model that the surfer transfer by clicking a hyperlink rather than other ways. Mathematically, the PageRank model can be formulated as the problem of finding the positive unit eigenvector  $x$  (the so-called PageRank vector) such that

$$Ax = x, \quad \|x\| = 1, \quad x > 0 \quad (3)$$

or, equivalently, as the solution of the linear system

$$(I - \alpha \tilde{P})x = (1 - \alpha)v \quad (4)$$

In the past decade or so, considerable research attention has been devoted to the efficient solution of problems 3 4, especially when  $n$  is very large. For moderate values of the damping factor, e.g. for  $\alpha = 0.85$  as initially suggested by Google for search engine rankings, solution strategies based on the simple Power method have proved to be very effective. However, when  $\alpha$  approaches 1, as is required in some applications, the convergence rates of classical stationary iterative methods including the Power method tend to deteriorate sharply, and more robust algorithms need to be used.

One area that is largely unexplored in PageRank computations is the efficient solution of problems with the same network structure but multiple damping factors. For example,

in the Random Alpha PageRank model used in the design of anti-spam mechanism [1], the rankings corresponding to many different damping factors close to 1 need to be computed simultaneously. This problem can be expressed mathematically as solving a sequence of linear systems

$$(I - \alpha_i \tilde{P})x_i = (1 - \alpha_i)v \quad \alpha_i \in (0, 1) \quad \forall i \in \{1, 2, \dots, s\}S \quad (5)$$

Conventional PageRank algorithms applied to 5 would solve the  $s$  linear systems independently. Although these solutions can be performed in parallel, the process would still demand large computational resources for high dimension problems. This consideration motivates the search of novel methods with reduced algorithmic and memory complexity, to afford the solution of larger problems on moderate computing resources. We can write the PageRank problem with multiple damping factors given at once (5) as a sequence of shifted linear systems of the form:

$$\left(\frac{1}{\alpha_i}I - \tilde{P}\right)x^{(i)} = \frac{1 - \alpha_i}{\alpha_i}v \quad \forall i \in \{1, 2, \dots, s\} \quad 0 < \alpha_i < 1 \quad (6)$$

Shifted Krylov methods may still suffer from slow convergence when the damping factor approaches 1, requiring larger search spaces to converge with satisfactory speed, which in turn may lead to unaffordable storage requirements for large-scale engineering applications. As an attempt of a possible remedy in this situation, we present a framework that combines shifted stationary iterative methods and shifted Krylov subspace methods. In detail, we derive the implementation of the Power method that solves the PageRank problem with multiple damping factors at almost the same computational cost of the standard Power method for solving one single system. Furthermore, we demonstrate that this shifted Power method generates collinear residual vectors. Based on this result, we use the shifted Power iterations to provide smooth initial solutions for running shifted Krylov subspace methods such as GMRES. Besides, we discuss how to apply seed system choosing strategy and extrapolation techniques to further speed up the iterative process.

## 1.1 Overview of the classical PageRank problem

The Power method is considered one of the algorithms of choice for solving either the eigenvalue 3 or the linear system 4 formulation of the PageRank problem, as it was originally used by Google. Power iterations write as

$$x_{(k+1)} = Ax_k = \alpha \tilde{P}x_{(k)} + (1 - \alpha)v \quad (7)$$

The convergence behavior is determined mainly by the ratio between the two largest eigenvalues of  $A$ . When  $\alpha$  gets closer to 1, though, the convergence can slow down significantly.

As stated in [2] The number of iterations required to reduce the initial residual down to a tolerance  $\tau$ , measured as  $\tau = \|Ax_k - x_k\| = \|x_{k+1} - x_k\|$  can be estimated as  $\frac{\log_1 0\tau}{\log_1 0\alpha}$ . For example, when  $\tau = 10^{-8}$  the Power method requires about 175 steps to converge for  $\alpha = 0.9$  but the iteration count rapidly grows to 1833 for  $\alpha = 0.99$ . Therefore, for values of the damping parameter very close to 1 more robust alternatives to the simple Power algorithm should be used.

## 2 The shifted power method for PageRank computations

In this section we consider extensions of stationary iterative methods for the solution of PageRank problems with multiple damping factors. We look in particular at the Power method, the Gauss-Seidel method, and the GIO iteration scheme. We are concerned with how these methods can be executed with the highest efficiency for solving such problems, especially with the question: for each method, whether there exist an implementation such that the computational cost of solving the PageRank problem with multiple damping factor is comparable to that of solving the ordinary PageRank problem with single damping factor.

### 2.1 The implementation of the shifted power method

Inspired by the reason why shifted Krylov subspaces can save computational cost, we investigate whether there are duplications in the calculations of multiple linear systems in this problem class by the stationary iterative methods, so that the duplications in the computation can be deleted, or in other words, the associate operations can be computed only once and used for all systems. We first analyze the Power method applied to the sequence of linear systems in 4. It computes at the  $k$ th iteration approximate solutions  $x_k(i)$  ( $1 \leq i \leq s$ ) of the form

$$\alpha_i^k \tilde{P}^k x_k^{(i)} + (1 - \alpha_i^k) \sum_{j=0}^{k-1} \alpha_i^j \tilde{P}^j v \quad (8)$$

If the  $s$  systems in 4 are solved synchronously, that is all  $x_k^{(i)}$  are computed only after all previous approximations  $x^{(j)}_{k-1}$  are available, then the computation can be rearranged efficiently as follows:

- at the first iterations
  - compute and store  $\mu_1 = \tilde{P}x_0$  and  $\mu_2 = v$ ;
  - compute and store  $x_1^{(i)} = \alpha_i \mu_1 + (1 - \alpha_i) \mu_2$ ;
- at any other subsequent iteration  $k > 1$ 
  - compute and store  $x_k^{(i)} := (1 - \alpha_i) \sum_{j=0}^{k-2} \alpha_i^j \tilde{P}^j v = x_{k-1}^{(i)} - \alpha_i^{k-1} \mu_1$ ;
  - compute and store  $\mu_1 = \tilde{P}\mu_1$  and  $\mu_2 = \tilde{P}\mu_2$ ;
  - compute and store  $x_k^{(i)} = \alpha_i \mu_1 + x_k^{(i)} + (1 - \alpha_i) \alpha_i^{k-1} \mu_2$ .

This implementation requires at most 2 matrix-vector products at each step, which is a significant gain compared to the  $s$  matrix-vector products required by the standard Power method to compute  $x_{k+1}^{(i)}$ , especially when  $s \gg 2$ . This is close to the computational cost, i.e. 1 matrix-vector product per iteration, of using the Power method for computing PageRank with single damping factor.

An efficient implementation can compute and store  $\mu = \tilde{P}v - v$  at the first iteration and store  $\mu = \tilde{P}^{k-1}(\tilde{P}v - v) = \tilde{P} \cdot (\tilde{P}^{k-2}(\tilde{P}v - v))$  at each  $k$ th iteration ( $k > 1$ ), and finally from each approximate solution as  $x_k^{(i)} = \alpha_i^k \mu + x_{k-1}^{(i)}$ . The residual vector  $r_k^{(i)}$  associated with the approximate solution  $x_k^{(i)}$  has the following expression

$$r_k^{(i)} = Ax_k^{(i)} - x_k^{(i)} = x_{k+1}^{(i)} - x_k^{(i)} = \alpha_i^{k+1} \tilde{P}^k (\tilde{P}v - v) \quad (9)$$

Since in general each of the  $s$  linear systems may require a different number of Power iterations to converge, the  $s$  residual norms have to be monitored separately to test the convergence. We summarize the efficient implementation of the Power method that we presented in this section for solving problem 4 in Algorithm 1, and we call it the shifted Power method hereafter.

---

**Algorithm 1** Shifted-Power method for PageRank with multiple damping factors

---

**Require:**  $\tilde{P}$ ,  $v$ ,  $\tau$ ,  $\max_{mv}$ ,  $\alpha_i$  ( $1 \leq i \leq s$ )

**Ensure:**  $mv$ ,  $x^{(i)}$ ,  $r^{(i)}$  ( $1 \leq i \leq s$ )

  Compute  $\mu = \tilde{P}v - v$

  Set  $mv = 1$

**for**  $do i = 1 : s$

    Compute  $r^{(i)} = \alpha_i \mu$

    Compute  $Res(i) = \|r^{(i)}\|$

**if**  $Res(i) \geq \tau$  **then**

      Compute  $x^{(i)} = r^{(i)} + v$

**end if**

**end for**

**while**  $\max(Res \geq \tau)$  and  $mv \leq \max_{mv}$  **do**

    compute  $\mu = \tilde{P}\mu$

$mv = mv + 1$

**for**  $i = 1 : s$  **do**

**if**  $Res(i) \geq \tau$  **then**

        Compute  $r^{(i)} = \alpha_i^{k+1} \mu$

        Compute  $Res(i) = \|r^{(i)}\|$

**if**  $Res(i) \geq \tau$  **then**

          Compute  $x^{(i)} = r^{(i)} + x^{(i)}$

**end if**

**end if**

**end for**

**end while**

---

Where  $mv$  is an integer that counts the number of matrix-vector products performed by the algorithm. The algorithm stops when either all the residual norms are smaller than the tolerance  $\tau$  or the maximum number of matrix-vector products is reached.

## References

- [1] Paul G. Constantine and David F. Gleich. Random alpha pagerank. *Internet Mathematics*, 6(2), 1 2009.
- [2] Zhao-Li Shen, Meng Su, Bruno Carpentieri, and Chun Wen. Shifted power-gmres method accelerated by extrapolation for solving pagerank with multiple damping factors. *Applied Mathematics and Computation*, 420:126799, 2022.