Adaptive Ranking of Web Pages

Ah Chung Tsoi
Office of Pro Vice-Chancellor (IT)
University of Wollongong
Wollongong, NSW 2522,
Australia

Gianni Morini
Dipartimento di Ingegneria
dell'Informazione
Universita' degli studi di Siena
Siena, Italy

Franco Scarselli
Dipartimento di Ingegneria
dell'Informazione
Universita' degli studi di Siena
Siena, Italy

Markus Hagenbuchner
Office of Pro Vice-Chancellor (IT)
University of Wollongong
Wollongong, NSW 2522,
Australia

Marco Maggini
Dipartimento di Ingegneria
dell'Informazione
Universita' degli studi di Siena
Siena, Italy

ABSTRACT

In this paper, we consider the possibility of altering the PageRank of web pages, from an administrator's point of view, through the modification of the PageRank equation. It is shown that this problem can be solved using the traditional quadratic programming techniques. In addition, it is shown that the number of parameters can be reduced by clustering web pages together through simple clustering techniques. This problem can be formulated and solved using quadratic programming techniques. It is demonstrated experimentally on a relatively large web data set, viz., the WT10G, that it is possible to modify the PageRanks of the web pages through the proposed method using a set of linear constraints. It is also shown that the PageRank of other pages may be affected; and that the quality of the result depends on the clustering technique used. It is shown that our results compared well with those obtained by a HITS based method.

Categories and Subject Descriptors

H.3.3 [**Information storage and retrieval**]: Information Search and Retrieval; H.5.4 [**Information interfaces and presentation**]: Hypertext/Hypermedia

General Terms

Search Engine, Page rank

Keywords

Adaptive PageRank determinations, Learning PageRank, quadratic programming applications

1. INTRODUCTION

Google is probably one of the most popular internet search engines available today. A major feature of the Google search engine lies in its arrangement of the most "relevant" pages with respect to a user's inquiry [1]. This set of most "relevant" pages is obtained by what is known as the PageRank algorithm [2, 3, 4], in which the web pages are ranked according to the number of links which point to it. A page with many links pointing to it is highly authoritative. In [2], an algorithm was given to determine the PageRank of the web pages according to their link structures. It is shown in [2] that the PageRank of the set of web pages can be computed using the following recursive algorithm:

$$\boldsymbol{X}(t+1) = d\, W\boldsymbol{X}(t) + (1-d)\mathbb{1}_n \qquad (1)$$

where $\boldsymbol{X} \in \mathcal{R}^n$ is an n-dimensional vector denoting the PageRank of the set of n web pages. $\boldsymbol{X}(t)$ denotes the evolution of the PageRank vector at the t-th iteration. W is a $n \times n$ matrix with elements $w_{i,j} = \frac{1}{h_j}$ if there is a hyperlink from node j to node i, and h_j is the total number of outlinks of node j, and $w_{i,j} = 0$ otherwise. $\mathbb{1}_n$ is an n-dimensional vector with all elements equal to 1. d is a damping factor. The PageRank of the set of n web pages is given by the steady state solution of (1).

At the steady state, we have

$$\boldsymbol{X} = (1-d)(I-dW)^{-1}\mathbb{1}_n \qquad (2)$$

Note that the PageRank of the set of web pages is determined once the link structure among the web pages is given, and d fixed. In other words, the PageRank of a set of web pages can be determined uniquely once the link structure of the web pages is fixed, and $d \in (0,1)$ is satisfied [1, 3].

Note also that PageRank is only one among a number of factors which Google uses to arrange the final score[1] of a particular web page.

There are various attempts in altering the PageRank of a set of web pages. There are two perspectives: (a) from the perspective of users, and (b) from the perspective of web site administrators.

There is much discussion in the commercial literature on how to influence the PageRank of web pages from a user's point of view. This is in recognition of the economic gain which might be derived from having highly ranked web pages, as a human web surfer is mostly interested in the first few

[1] In this paper, we make a distinction between "PageRank" and "score". PageRank is the value which is obtained from the steady state solution of (1), while score is the final value which is obtained by taking into account a number of other factors, e.g., the anchor text, keywords, etc.

WWW2003, May 20–24, 2003, Budapest, Hungary.
ACM 1-58113-680-3/03/0005.

pages of returned universal resource locators (URLs) from the search engine query on a particular topic. There are various techniques deployed in the search engine optimization literature, mainly in changing the link structure of web pages which are under the control of the user. As indicated previously, the link structure is one of the factors which will determine the Google's score. Hence, by modifying the link topology, paying special attention to the way that the PageRank is computed, it is possible to raise the PageRank of selected pages under the control of the user. In addition, by exchanging links with other users, it is possible to raise the PageRank of selected pages further.

From the web administrator's point of view, there are also reasons for raising or decreasing the PageRank of certain web pages. For example, if a user uses a "link farm" to artificially inflate their PageRanks, it would be useful if the web site administrator has a way to decrease the influence of the link farm mechanism in its determination of the PageRanks for other web pages. In the case that the web site is used as a special purpose portal, there may be situations in which the web site administrator would wish to increase the PageRank of web pages which are related to the topics carried by the portal. For example, the administrator of a portal on wine may wish that pages about wine have a higher rank than other pages which are unrelated to wine. The administrator of a search engine may need to decrease the rank of spamming pages; or the administrator of a site may wish that the energy[2] of his/her site is higher than the energy of a competitor.

In this paper, we will consider possible modifications of PageRanks from a web administrator's point of view. It is assumed that the web administrator has no possibilities of modifying the topology of the link configuration of the web pages. The only mechanism which is opened to the web administrator is to modify the PageRank equation (1) by modifying the "control" variable in that equation.

Consider the PageRank equation (1). It is simple to note that this equation can be written in more conventional notations as follows:

$$\boldsymbol{X}(t+1) = A\boldsymbol{X}(t) + B\mathbf{u}(t) \qquad (3)$$

where $\boldsymbol{X}$ is a n dimensional vector, often called the state of the equation, $\mathbf{u}$ is a m dimensional vector, called the input vector. A is $n \times n$ constant matrix and B is $n \times m$ constant matrix. In the case of PageRank equation, $m = n$, $B = I$, the identity matrix, $A = dW$, and $\mathbf{u}$ has all elements equal to $(1 - d)$. Thus, the issue here is how to design the control variable $\mathbf{u}$ in such a way that the PageRanks of a set of selected web pages are altered.

There are two ways in which we can modify the PageRank by manipulating the control variable $\mathbf{u}$:

- Dynamic control. In this case, the issue is to design a set of controls $\mathbf{u}(t)$, $t = 1, 2, \ldots$ such that the PageRanks is modified (when they reach the steady state).

- Static control. In this case, recognizing the fact that the PageRank is the steady state solution of (1), it might be possible to design $\mathbf{u}$ in such a way that the PageRanks of selected web pages are modified.

In this paper, we will only consider the simpler case of static control design, rather than the more difficult dynamic control design. Thus the problem we wish to address in this paper is: given a set of PageRanks of selected web pages obtained from the steady state solution of (2), is it possible to modify this set of PageRanks by designing a set of controls.

This question as it stands is ill-posed in that we need a criterion to determine the nature of the control variables used and the nature of constraints in which we wish to place on the modified PageRanks. Let us denote the set of PageRanks given by (2) as $\boldsymbol{X}_g$, where g denotes that this is the PageRank as obtained by using Google's PageRank equation. In subsection 1.1, we will first consider the type of constraints which may be placed on the original PageRanks, while in subsection 1.2, we will consider the cost criterion which can be placed on the system to obtain a solution.

1.1 The constraints

In the simplest case, the ranks of selected pages are set to predefined target values. For example, the administrator of a search engine may notice that a page is ranked lower than desired and may decide to increase its rank by 50%. Thus, the target for that page will be the original rank times 1.5.

Formally, let us assume that pages $p_1, \ldots, p_t$ have the targets $y_1, \ldots, y_t$, respectively, whereas the ranks of other pages are undefined. The constraint can be written as follows:

$$S\boldsymbol{X} = \boldsymbol{Y} \qquad (4)$$

where $\boldsymbol{Y}' = [y_1, \ldots, y_t]$, the superscript $'$ denotes the transpose of a vector, or a matrix, and $S \in \mathcal{R}^{t \times n}$ is a projection matrix such that $S\boldsymbol{X} = [x_{i_1}, \ldots, x_{i_t}]$.

In other cases, one may wish to establish an ordering on the pages. For example, we can enforce $x_i \geq x_j$ by the inequality $\boldsymbol{V}\,\boldsymbol{X} \geq 0$, where $\boldsymbol{V} = [v_1, \ldots, v_n]$, $v_i = 1, v_j = -1$ and $v_k = 0, k \neq i, j$. Similarly, one can constrain the energy of the site to be larger than a threshold b, by the inequality $\mathbb{1}'Sx \geq b$.

More generally, a set of r rules on the PageRanks can be formally defined as follows:

$$B\boldsymbol{X} \geq \boldsymbol{b} \qquad (5)$$

where $B \in \mathcal{R}^{r \times n}$ and $\boldsymbol{b} \in \mathcal{R}^r$. These rules can constrain the ordering between two pages; they can enforce the ordering between the energy of two communities; they can establish a range of desired values for the rank of a page and so on.

1.1.1 Additional constraints

Most likely administrators have no a priori information concerning the range of PageRanks of vector $\boldsymbol{X}$; therefore a common constraint will be to deal with the order of the pages, rather than the PageRanks. For example, if our set of pages consists of $\{p_1, p_2\}$ and we wish that the rank x_1 be more important than x_2, it seems natural to write

$$x_1 \geq \; c\, x_2$$

where $c \geq 1$ is a coefficient chosen by the administrator.

The following example shows that such a constraint does not necessarily ensure the desired result:

$$\boxed{\begin{array}{ll} x_1 = \;\; -2 & x_{ap1} = 2 \\ x_2 = -1.5 & x_{ap2} = 1 \end{array}} \quad \boxed{\begin{array}{cccc} x_1 & \geq & 2x_2 & \text{True} \\ x_{ap1} & \geq & x_{ap2} & \text{False} \end{array}}$$

where x_{ap} is the absolute position induced by the rank. In order to avoid this problem, we need to enforce the rank of

[2] The sum of the PageRank of the web pages in the site [1] as a measure of the collective "power" of a set of web pages.

each page involved in the constraints to be positive. In our example the set of constraints will be:

$$\boxed{\begin{array}{lll} x_1 - 2x_2 & \geq & 0 \\ x_1 & \geq & 0 \\ \quad\; x_2 & \geq & 0 \end{array}}$$

Hence, an additional constraint on the variables is that the state variables $\boldsymbol{X}$ for which the constraints are satisfied must satisfy further that $\boldsymbol{X} \geq 0$.

1.2 Cost

It is reasonable to adopt the following assumption: apart from the web pages whose PageRanks need to be modified, for the rest of web pages in the web site, we do not wish to perturb their current rank if possible.

Let $\boldsymbol{X}_g$ be the PageRank of a set of web pages as obtained from (2). Let $\boldsymbol{X}_a$ be the modified PageRanks the same set of web pages when we apply the control. Then, it is reasonable to expect that the following cost function can be imposed:

$$J = \|\boldsymbol{X}_a - \boldsymbol{X}_g\|_p \tag{6}$$

where $\|\cdot\|_p$ denotes the p-norm of the set of vectors, $p \in \mathbb{N}^+$. In this paper, we will only consider the case where $p = 2$.

1.3 Summary

The problem of modifying the PageRank can be posed as follows:

$$\min_{\boldsymbol{E}} J = \|\boldsymbol{X}_a - \boldsymbol{X}_g\|_2 \tag{7}$$

where $\boldsymbol{E}$ is an n dimensional vector, denoting the set of control variables, subject to the constraints:

- $$\boldsymbol{X}_g = (1-d)(I - dW)^{-1}\mathbb{1}_n,$$
- $$\boldsymbol{X}_a(t+1) = dW\boldsymbol{X}_a(t) + \boldsymbol{E} \tag{8}$$
- $$B\boldsymbol{X}_a \geq \boldsymbol{b} \tag{9}$$
- $$\boldsymbol{X}_a \geq 0.$$

The structure of the paper is as follows: In Section 2, we will first consider a solution of the problem at hand. In Section 3, we will show how this solution can be modified to solve the more general case. In Section 4, we will show how the constraints can be relaxed. In Section 5, we will present a set of experimental results designed to verify the behaviour of the proposed algorithm, and to show what types of solutions are possible. Conclusions are drawn in Section 6.

2. FIRST SOLUTION

Since the PageRank is obtained as the steady state solution of (1), it is reasonable to infer that we will only be interested in the steady state solution of (8). Towards this end, if we define

$$M = (I - dW)^{-1} \tag{10}$$

then we can write the solution of (8) as

$$\boldsymbol{X}_a = M\boldsymbol{E} \tag{11}$$

which is the same as PageRank except for the vector $\boldsymbol{E} \in \mathcal{R}^n$ in place of $(1-d)\mathbb{1}_n$. We can substitute (11) in (9) to obtain

$$B\,M\boldsymbol{E} \geq \boldsymbol{b} \tag{12}$$

Now we can consider the cost function J as follows:

$$\begin{aligned} \|\boldsymbol{X}_a - \boldsymbol{X}_g\|_2 \;&=\; \|M(\boldsymbol{E} \;-\; \mathbb{1}_n)\|_2 \\ &= (\boldsymbol{E}^T - \mathbb{1}_n^T)M^T\,M(\boldsymbol{E} - \mathbb{1}_n) \\ &= \boldsymbol{E}^T M^T M\boldsymbol{E} - 2\,\mathbb{1}_n^T\,M^T\,M\boldsymbol{E} + \mathbb{1}_n^T\,M^T\,M\mathbb{1}_n \end{aligned} \tag{13}$$

where the constant term $\mathbb{1}_n^T\;M^T M\mathbb{1}_n$ can be omitted in the minimization of the function

$$f(\boldsymbol{E}) = \boldsymbol{E}^T\;M^T M\boldsymbol{E} -\;2\,\mathbb{1}_n^T\;M^T M\boldsymbol{E} \tag{14}$$

Finally, we can use (14) as cost function in

$$\begin{cases} \min_{\boldsymbol{E}} \quad \boldsymbol{E}^T\;M^T M\boldsymbol{E} -\;2\,\mathbb{1}_n^T\;M^T M\boldsymbol{E} \\ \\ B\,M\boldsymbol{E} \geq \boldsymbol{b} \end{cases} \tag{15}$$

Notice that (15) is a standard positive definite quadratic programming problem with an inequality constraint set which can be solved very efficiently [5]. The problem fits in the positive definite quadratic programming problem because $M^T M$ is positive definite with an inequality constraint set.

The solvability of this problem is given as follows:

1. $M^T M$ is positive definite. In this case, it is satisfied.
2. $B\boldsymbol{q}^* \geq \boldsymbol{b}$, and $\hat{B}\boldsymbol{q}^* = \boldsymbol{b}$, where the superscript $*$ denotes the optimal solution and $\hat{B}$ denotes the reduced matrix B which contains only t rows in which the constraints are satisfied, assuming that there are only t constraints which are active.
3. $\hat{B}^T\lambda^* = -2M\mathbb{1}_n$, where λ is the set of Lagrange multipliers, and $\lambda_j^* \geq 0$, $j = 1, 2, \ldots, t$.

Later, in Section 4, we introduce a method to compute a sub-optimal solution when the constraints in (15) do not have feasible solutions.

3. PRACTICAL SOLUTION

For the world wide web, n, the dimension of the PageRank equation, or the modified PageRank equation can be in the region of billions. Hence it would not be possible to solve the quadratic programming problem formulated in Section 2, apart from some very simple problems of which n is of low order. In this section, we will introduce a practical method for solving the situation when n is large. The key is to group pages in the world wide web together into clusters, and thus reducing the number of dimension of the state space which needs to be considered.

3.1 Reduce the complexity

Let us consider a partition $C_1, \ldots, C_k$ of web pages. In other words, we wish to partition the total number of web pages into k clusters. These clusters can be arranged according to some criteria, e.g., approximately the same PageRank, approximately the same score. For sake of simplicity, we further assume that the pages are ordered so that the pages in cluster C_1 are $p_1, \ldots, p_{n_1}$, the pages in cluster C_2 are $p_{n_1+1}, \ldots, p_{n_2}$ and so on. In our approach, the vector E is defined by k parameters (control variables), one parameter for each cluster. The main reason why we use this technique is that the number of control variables in the modified PageRank equation is the same as the number of states. Hence, if the number of states is reduced through clustering from n

to k, then the corresponding number of control variables is reduced as well. More precisely, we have

$$\begin{array}{c} E' = (e_1, \ldots e_1, e_2, \ldots e_2, \ldots e_k, \ldots e_k) \\ \quad\quad \uparrow \quad\quad\quad \uparrow \quad\quad\quad\quad \uparrow \\ \quad\quad C_1 \quad\quad C_2 \quad \ldots \quad C_k \end{array} \tag{16}$$

Intuitively, the parameter e_i controls the a priori rank given to pages of class C_i. This method will determine the a priori rank in order to satisfy the constraint (4) and/or (9). In this way, we consider that the pages are grouped into classes. Thus, for example, increasing the rank of a page about wine will probably produce also an increase in the rank of other pages about wine[3]. Moreover, the method gives consideration to the connectivity (the connection topology) of the web pages, since whereas the parameters $e_1, \ldots, e_k$ control the a priori rank given to the web pages, the constraints control the final rank x of these web pages. For example, the algorithm increases the rank of pages on wine may also increase the rank of their parent pages[3], e.g. the pages on cooking.

Let $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_n$ be the columns of $(I - dW)^{-1}$. Note that

$$\begin{aligned} x &= e_1 \boldsymbol{V}_1 + \ldots + e_1 \boldsymbol{V}_{n_1} + e_2 \boldsymbol{V}_{n_1+1} + \ldots + e_k \boldsymbol{V}_{n_k} \\ &= \sum_{i=1}^{k} e_i \boldsymbol{A}_i \end{aligned}$$

where $\boldsymbol{A}_i$, $i = 1, \ldots, k$, are the vectors obtained by summing all the columns $\boldsymbol{V}_{n_{i-1}+1}, \ldots, \boldsymbol{V}_{n_i}$ that correspond to the class i. We define further:

$$A = [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_k] \qquad \boldsymbol{E}_r = \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_k \end{pmatrix} \tag{17}$$

Thus, we can write $\boldsymbol{X}_a = A\ \boldsymbol{E}_r$ which allows us to rewrite (12) as follows:

$$H\boldsymbol{E}_r \geq b \tag{18}$$

where $H = B\,A$. Moreover, note that the vectors $\boldsymbol{A}_i$ fulfill

$$\boldsymbol{A}_i = (I - dW)^{-1} \boldsymbol{O}_i$$

where $\boldsymbol{O}_i = [0, \ldots, 0, 1, \ldots, 1, 0, \ldots, 0]$ is the vector where the j-th component is 1 if j-th page belongs to i-th class and 0, otherwise. Thus, the vectors $\boldsymbol{A}_i$ can be easily computed, using the same approach adopted for PageRank, by

$$\boldsymbol{A}_i(t+1) = dW\boldsymbol{A}_i(t) + \boldsymbol{O}_i \tag{19}$$

Let us define the clustering matrix

$$U = [\boldsymbol{O}_1, \boldsymbol{O}_2, \ldots, \boldsymbol{O}_k] \tag{20}$$

We can use U in the following equation:

$$\boldsymbol{E} = U\ \boldsymbol{E}_r \tag{21}$$

Since $X_a = M\ E = M\ U\ \boldsymbol{E}_r$, so we can write $A = M\ U$.

Now we can focus on the cost function (14). We use (21) to write:

$$f(\boldsymbol{E}) = \boldsymbol{E}^T\ M^T M \boldsymbol{E} -\ 2\ \mathbb{1}_n^T\ M^T M \boldsymbol{E} =$$

$$\boldsymbol{E}_r^T\ U^T M^T M U \boldsymbol{E}_r -\ 2\ \mathbb{1}_n^T\ M^T M U \boldsymbol{E}_r$$

$$= \boldsymbol{E}_r^T\ A^T A \boldsymbol{E}_r -\ 2\ \mathbb{1}_n^T\ M^T A \boldsymbol{E}_r$$

In the following, we use $Q = A^T A$ for the quadratic term and $\boldsymbol{p} = -2\ M^T A\ \mathbb{1}_n$ for the linear term in the cost function

$$f(\boldsymbol{E}) = \boldsymbol{E}_r^T\ Q\ \boldsymbol{E}_r\ +\ \boldsymbol{p}\,\boldsymbol{E}_r \tag{22}$$

Consequently, in order to find the optimal solution $\boldsymbol{E}_r^*$, we proceed to solve the quadratic programming problem[4]

$$\begin{cases} min\ \ \boldsymbol{E}_r^T\ Q\ \boldsymbol{E}_r\ +\ \boldsymbol{p}\,\boldsymbol{E}_r \\ \\ H\ \boldsymbol{E}_r \geq \boldsymbol{b} \end{cases} \tag{23}$$

Note that $H \in \boldsymbol{R}^{(t \times k)}$, where the number of constraints t and the number of clusters k is small, and $n >> k$.

4. RELAXED CONSTRAINTS PROBLEM

In the event that administrators define contradicting constraints then (23) has no feasible solutions. In order to compute a sub-optimal solution, we introduce a method to regulate the strength of the constraints. Instead of forcing the algorithm to fulfill the constraints, we add a new term to the cost function:

$$\begin{cases} min\ (1{-}s)(\boldsymbol{E}_r^T Q \boldsymbol{E}_r + \boldsymbol{p}\boldsymbol{E}_r) + s((H\boldsymbol{E}_r - \boldsymbol{b})^T\ I\ (H\boldsymbol{E}_r{-}\boldsymbol{b})) \\ \\ \forall \boldsymbol{E}_r \end{cases} \tag{24}$$

where the coefficient $s \in [0, 1]$ is used to balance the importance between constraints and the original cost function.

We wish to express (24) in a standard quadratic programming formulation.

We first focus on the second term in (24)

$$(H\boldsymbol{E}_r - \boldsymbol{b})^T\ I\ (H\boldsymbol{E}_r - \boldsymbol{b}) = (\boldsymbol{E}_r^T H^T - \boldsymbol{b}^T)\ (H\boldsymbol{E}_r - \boldsymbol{b}) =$$

$$\boldsymbol{E}_r^T H^T H \boldsymbol{E}_r - \boldsymbol{E}_r^T H^T \boldsymbol{b} - \boldsymbol{b}^T H \boldsymbol{E}_r + \boldsymbol{b}^T \boldsymbol{b}$$

Notice that we can remove the constant term $\boldsymbol{b}^T \boldsymbol{b}$ without affecting the solution. We can substitute in (24)

$$(1{-}s)(\boldsymbol{E}_r^T\ Q\ \boldsymbol{E}_r\ +\ \boldsymbol{p}\,\boldsymbol{E}_r) + s(\boldsymbol{E}_r^T H^T H \boldsymbol{E}_r - 2\boldsymbol{b}^T H \boldsymbol{E}_r) =$$

$$\boldsymbol{E}_r^T\ (\ (1{-}s)Q\ + sH^T H\ )\ \boldsymbol{E}_r\ +\ (\ (1{-}s)\boldsymbol{p}\ -\ 2\,s\,\boldsymbol{b}^T H\ )\ \boldsymbol{E}_r$$

Let us define

$$Z = (1 - s)Q\ + sH^T H \tag{25}$$

$$\boldsymbol{a} = (\ (1 - s)\boldsymbol{p}\ -\ 2\,s\,\boldsymbol{b}^T H\ )^T \tag{26}$$

Matrix Z is positive semi-definite as well and we can finally write (24) in the following manner:

$$\begin{cases} min\ \ \ (\boldsymbol{E}_r^T\ Z\ \boldsymbol{E}_r) + \boldsymbol{a}^T\ \boldsymbol{E}_r \\ \\ \forall \boldsymbol{E}_r \end{cases} \tag{27}$$

This approach allows us to find a sub-optimal solution for every constraint set. The parameter s influences the resulting rank vector X_a in the following manner:

$$\begin{cases} s \rightarrow 1 & \textit{The order induced by } X_a \textit{ and } X_p \textit{ is similar.} \\ s \rightarrow 0 & X_a \textit{ is the closest solution to the optimal.} \end{cases}$$

[3] This is assuming that the clustering method used is related to the content of the pages. This would not be true for clustering methods based on other criteria, e.g., scores.

[4] The standard quadratic programming formulation for the function cost is $min\ \frac{1}{2}x^T Q x + c^T x$ where Q is a $n \times n$ symmetric matrix.

4.1 Remarks

Our proposed method can be summarized as follows:

Step 1 Use a clustering algorithm in order to split the pages of the Web into clusters $C_1, \ldots, C_k$.

Step 2 Compute the A_i by solving the related k system defined by (19).

Step 3 If (9) has a feasible solution, solve the quadratic programming problem (23) in order to compute the optimal set of parameters $e_1, \ldots, e_k$

Step 4 If (9) has no feasible solutions, solve the quadratic programming problem (27) to compute a sub-optimal set of parameters $e_1, \ldots, e_k$.

Step 5 Compute the rank as $x = \sum_{i=1}^{k} e_i A_i$

The complexity of the algorithm is determined by steps 1–3. The computational cost of the clustering technique depends on the adopted clustering method. The cost of step 2 is k times the cost of that of the computation of PageRank. Step 3 requires to find a solution of a quadratic programming problem in k variables and t constraints. Provided that the number of constraints t and the number of classes k are not large, the problem can be solved in a reasonable time.

Note that the above algorithm works also if we use (4) instead of (9). Moreover, the method can be extended to the case when the clustering algorithm produces one or more classes for each page. In fact, let $L(p) = [c_1^p, \ldots, c_k^p]$ be a vector, where c_i^p measures the probability that page p belongs to class C_i. Let H_i be the vector $[c_i^1, \ldots, c_i^n]$ that represents how much each page belongs to class C_i.

We can consider the following dynamic system

$$x(t+1) = dWx(t) + e_1 H_1 + e_2 H_2 + \ldots + e_k H_k$$

where $e_1, \ldots, e_k$ are parameters that can be computed using the previously adopted reasoning. In this case, $A_i = (I - dW)^{-1} H_i$ will be calculated by

$$A_i(t+1) = dWA_i(t) + H_i$$

5. EXPERIMENTS

For the experiments conducted in this section we use a subset of the WT10G data set, as distributed by CSIRO in Australia. It pays special attention to the connectivity among the web pages. Hence this set of web collection is suitable for evaluations of search engine algorithms based on connectivity concepts, e.g., PageRank method. Some of the properties of WT10G dataset are as follows:

- 1,692,096 documents from 11,680 servers.
- 171,740 inter-server links (within dataset)
- 9,977 servers with inter-server in-links (within dataset)
- 8,999 servers with inter-server out-links (within dataset)
- 1,295,841 documents with out-links (within dataset)
- 1,532,012 documents with in-links (within dataset)

Instead of using the entire WT10G data set, we have chosen to use a subset comprising 150,000 documents. Such a subset is sufficiently large to evaluate our proposed algorithm while reducing the time needed to conduct individual experiments and hence, allowing a reasonable turn round time of tasks.

In this section we wish to investigate the following issues:

1. Is the proposed algorithm effective in rearranging the pages as desired?
2. How does the application of constraints on some pages affect the ranking of other pages in the collection?
3. The effects of the number of clusters on the performance of the algorithm.
4. The effect of clustering methods on the performance of the algorithm.

It is possible to combine tasks (1) and (2) since the answer to (2) allows the drawing of conclusions on (1). Experimental results are summarized in the following subsections.

5.1 Constraints effect

For this initial set of experiments we chose to cluster the pages into 15 equally sized clusters. The clusters are formed by sorting the pages according to their PageRanks in descending order as shown in Figure 1.

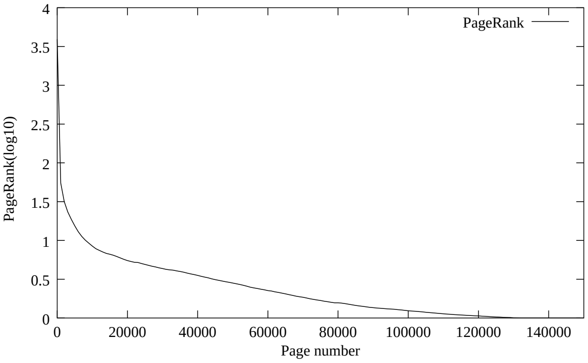


Figure 1: Rank distribution in the subset of the web collection with 150,000 pages.

The $10,000$ highest ranked pages are assigned to the first cluster. From the remaining set of pages we assign the next $10,000$ highest ranking pages to cluster 2, and so on.

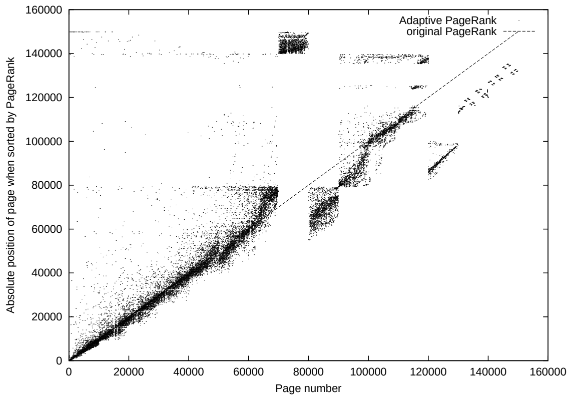
In order to simplify the evaluation, we consider only one type of constraints, viz., to swap the importance of two pages located at a distance Δ apart. Our aim is to find the effectiveness of the proposed algorithm, and to understand the influence of a single constraint on the page order of the rest of the web collection. Thus, we consider a page p with absolute position $p_{ap} = pos$ and page q with absolute position $q_{ap} = p_{ap} + \Delta$, and impose a constraint as follows:

$$x_q \quad - \quad x_p \quad \geq \quad 0$$

We give results from using $pos \in \{10, 1000, 5000, 20000\}$, and $\Delta = 1000$. For example, $pos = 10$ implies that the task is to swap the position of the 10-th page with the page located at position 1010 relative to PageRank. Results are presented in Figures 2 to 5.

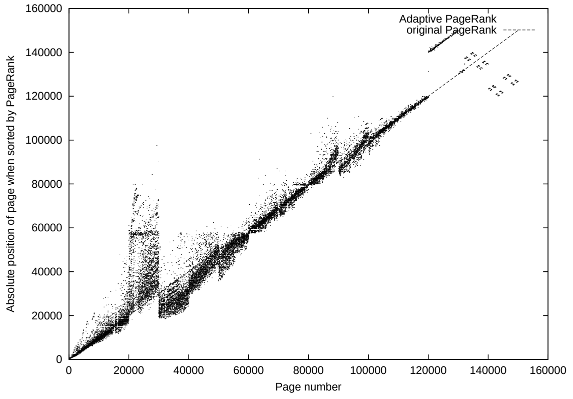
From Figures 2 to 5 we draw the following observations:

- The proposed algorithm is effective in modifying the PageRank as desired.



Page-ID	PageRank		Adaptive Rank	
WTX00...	rank	abspos	rank	abspos
7-B27-495	946.3	10	14.25	868
1-B38-30	54.49	1010	14.66	840
Std. deviation of page pos. changes = 22921.9				

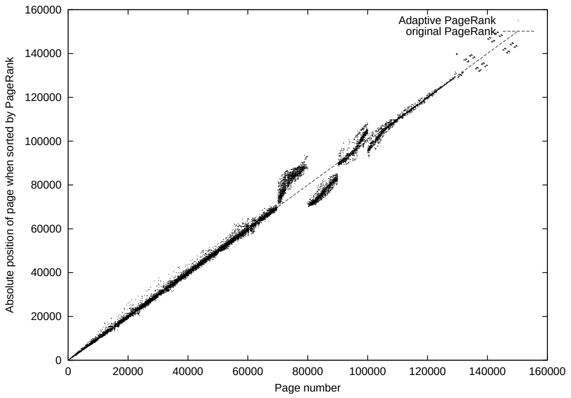
Figure 2: Test of the change in absolute position (abspos) with a single constraint ($\Delta = 1000$ and $pos = 10$). The y-axis represents the absolute position of a page (when sorted by rank in a descending order), the x-axis gives the original order of the web pages ranked by Google's PageRank method. A data point located at the diagonal indicates that the rank is unchanged. The number of clusters is fixed at 15.



Page-ID	PageRank		Adaptive Rank	
WTX00...	rank	abspos	rank	abspos
4-B13-2	54.92	1000	8.99	1765
5-B33-295	31.66	2000	9.13	1731
Std. deviation of page pos. changes = 10665.1				

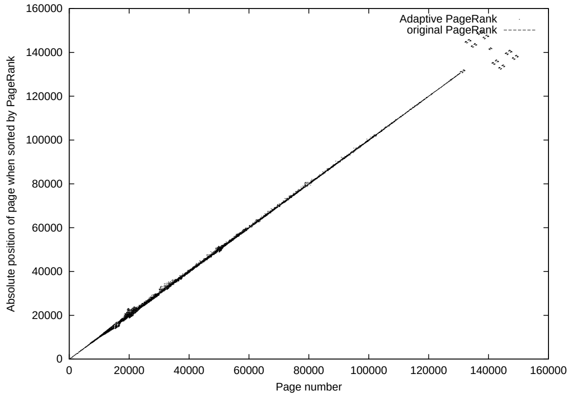
Figure 3: Test of the change in absolute position with a single constraint ($\Delta = 1000$ and $pos = 1000$). The y-axis represents the absolute position of a page (when sorted by rank in a descending order), the x-axis gives the original order of the web pages ranked by Google's PageRank method. The number of clusters is 15.

- It is observed that a constraint on highly ranked pages disturbs the PageRank of the rest of the web collection more significantly. For example, for $pos = 10$, it is observed that the standard deviation of the PageRank is 22921 as compared to 3260.1 when $pos = 20,000$.
- It is noted that the perturbations of the pages appear in blocks. We found that this is due to the clustering of the web pages. Pages belonging to a cluster are bound together by the same parameter. This finding implies that the quality of the result is influenced by the number of clusters used.
- It is observed that when swapping the positions of two pages, the effect on lower ranked pages is stronger than on higher ranked pages.



Page-ID	PageRank		Adaptive Rank	
WTX00...	rank	abspos	rank	abspos
4-B31-314	15.40	5000	3.52	5873
8-B46-339	12.97	6000	3.51	5898
Std. deviation of page pos. changes = 3932.1				

Figure 4: Test of the change in absolute position with a single constraint ($\Delta = 1000$ and $pos = 5000$). The y-axis represents the absolute position of a page (when sorted by rank in a descending order), the x-axis gives the original order of the web pages ranked by Google's PageRank method. The number of clusters is 15.



Page-ID	PageRank		Adaptive Rank	
WTX00...	rank	abspos	rank	abspos
6-B48-559	5.51	20000	1.38	22452
6-B45-249	5.35	21000	1.38	22439
Std. deviation of page pos. changes = 3260.1				

Figure 5: Test of the change in absolute position with a single constraint ($\Delta = 1000$ and $pos = 20,000$). The y-axis represents the absolute position of a page (when sorted by rank in a descending order), the x-axis gives the original order of the web pages ranked by Google's PageRank method. The number of clusters is 15.

- The observation that the location of higher ranked pages are perturbed by applying constraints on lower ranked pages can be explained by the fact that lowly ranked pages can have parents which are highly ranked. Hence, if we perturb the rank of the lowly ranked pages, then this can have an effect on parent pages which might be ranked higher. In other words, if we perturb the PageRanks of web pages, then we will perturb their associated ancestors as well. The extent to which the perturbation manifests itself depends on the original rank of the pages affected.

Further explanations of the results can be drawn by considering the distribution of the original PageRanks (Figure 1). From Figure 1 we find that only a small percentage of the pages have significant ranks. Thus, if the ranks are perturbed, it may be conceivable that the ranks of the perturbed pages are such that it takes very little effort for them to be different from the original ranks. In other words, the wide ranging perturbation observed in Figure 2 and Figure 3 may be due to the fact that there is not much difference in the ranks among the web pages in the mid-range and the end of the range of the distribution of ranks as shown in Figure 1. This explanation might be particularly appropriate to explain the relatively wildly perturbed ranks towards the end of the spectrum, e.g., in the range when the original ranks are between 120,000 and 150,000. From Figure 1, it is observed that the ranks of these pages are almost 1 ($\log_{10} 1 = 0$ as shown in the Figure), hence if they are perturbed, then it is easily conceivable that their relative positions may alter relatively wildly while the actual rank values change little.

The findings in this section suggest that the quality of the results is influenced by the number of clusters chosen. The effect of this parameter is investigated next.

5.2 Number of clusters

We investigate the influence of the number of clusters which were introduced to reduce the complexity. We conduct experiments that gradually increase the number of clusters used from 5 to a maximum of 100 clusters. The effect is evaluated by considering the cost function, and the standard deviation of the absolute position as a measure. The result is given by Figure 6.

It is observed that the cost function decreases with the number of clusters used, leveling out at around 60 clusters. Thus, it can be concluded that for the given data set, the cost function does not improve significantly even if we increase the number of clusters beyond 60. The standard deviation of the absolute position as a function of the variation of the number of clusters shows a similar behaviour.

The experiments confirmed that:

- It is possible to reduce the level of complexity of the algorithm by using the idea of clustering.
- The number of clusters may be quite small in comparison with the total number of web pages in the collection.

Each of these experiments required only minutes to execute on a dual headed Xeon-2GHz environment with 4 GB RAM; specifically, 2 minutes were required when using 15 clusters, and 7 minutes when using 50 clusters.

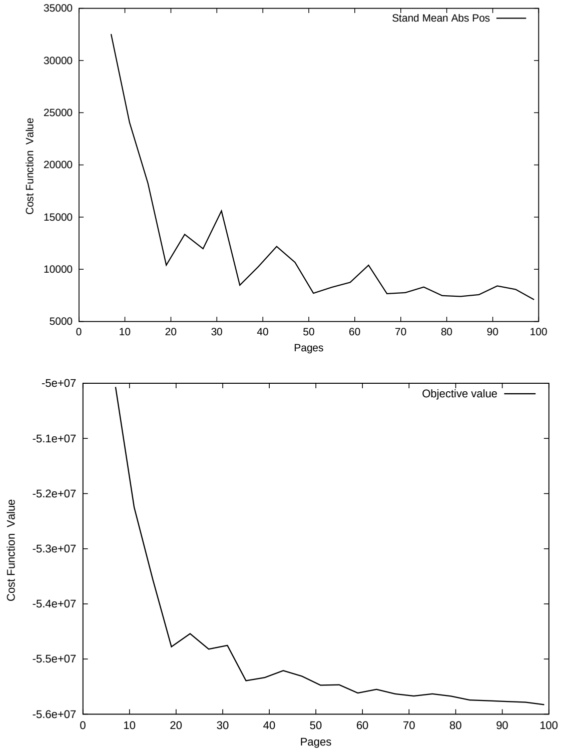


Figure 6: Evaluate the behaviour of the proposed algorithm as a function of the number of clusters. The graph on the top shows the variation of the standard deviation of the absolute position, while the graph on the bottom shows the variation of the cost function as a function of the number of clusters.

5.3 Clustering techniques

A number of ways in which web pages can be clustered are considered in the following:

(a) Clustering by score This simple method uses a classifier to assign a coefficient of affinity about a specific topic to each page in the web graph. We refer to this coefficient as the score s_p for the page p. Given a fixed number of cluster k, we compute the score range

$$r = \frac{s_{max} \;-\; s_{min}}{k}$$

A page p belongs to cluster i if $i = Mod\ \left(\frac{s_p}{r}\right)$. An effect is that clusters will be of different size. The dimension of each cluster depends on the distribution of the score in the graph.

(b) Clustering by rank This method suggests to use the PageRank as computed in (2). Given a fixed number of clusters k we compute the rank range

$$r = \frac{x_{max} \;-\; x_{min}}{k}$$

A page p belongs to cluster i if $i = Mod\ \left(\frac{x_p}{r}\right)$. The dimension of each cluster depends on the distribution of the rank in the web graph.

(c) Clustering by rank with fixed cluster dimensions The dimension of each cluster can be fixed to $n_c = \frac{n}{k}$, where n is the dimension of the web graph. This is done by ordering the pages of the graph according to the rank as computed in (2), and assign the first n_c pages to the cluster C_1, the second set of n_c pages to the cluster C_2 and so on. This clustering method was used for the experiments shown earlier in this section.

(d) Clustering by rank with variable cluster dimensions using a set regime The idea of this method is motivated by observations made on experimental findings made earlier. The idea is to treat highly ranked pages differently because they play a critical role. The size of the cluster is to be smaller for clusters that contain more relevant pages, while we can tolerate larger dimensions for clusters that contain relatively irrelevant pages. We define a coefficient b and a multiplication factor m. We order the pages of the web graph according to the rank as computed in (2) and we assign the first b pages to the cluster C_1, the second set of $m \times b$ pages to the cluster C_2 and so on. For example, for $b = 10$ and $m = 2$ the resulting cluster dimensions will be $\{10, 20, 40, 80, 160, 320, 640, \ldots\}$

Some of these clustering techniques were tried in experiments where the result is given in Figure 8. Case (b) was not attempted as this case is approximated by case (d). For the case where we cluster by score using variable dimensions the cluster sizes are as shown in Figure 7.

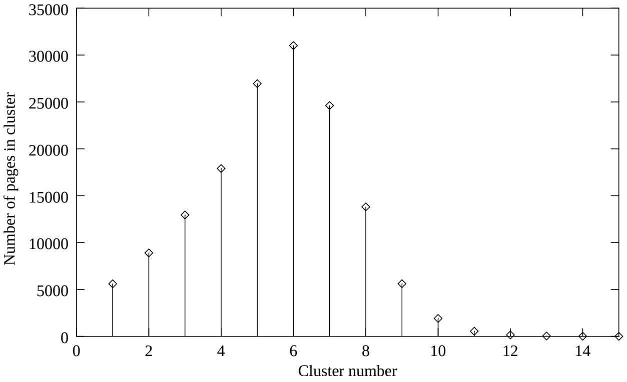


Figure 7: The distribution of web pages in the clustering by scores using variable dimension method.

The constraint used in this experiment is to swap the position of two web pages located at 1000 and 2000.

From Figure 7 it is observed that:

- The clustering scheme has a significant influence on the quality of the result.
- To build clusters from page scores generally produces the worst performance in terms of the perturbation on the PageRanks.
- Clustering methods based on rank gives good results.
- The clustering by ranks using a variable dimension by considering the magnitude of the PageRank appears to be working best.

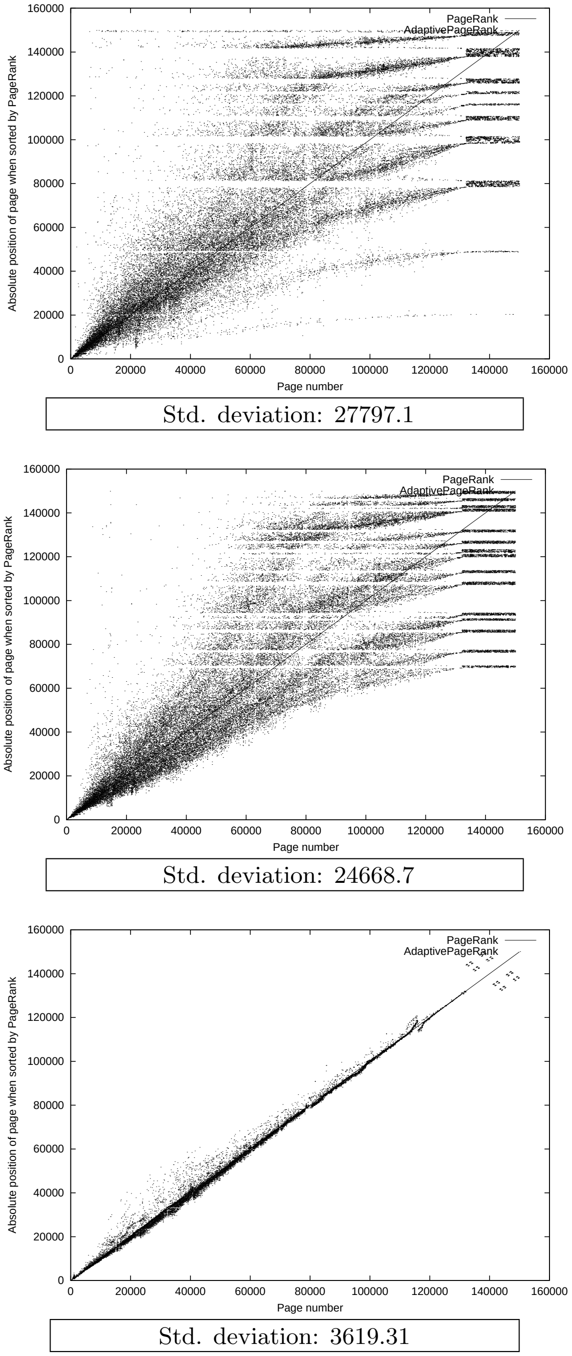


Figure 8: Results of experiments on the effects of various clustering methods on the performance of the proposed algorithm. The upper plot uses clustering by score with variable dimension, the center plot uses clustering by scores with fixed dimension. The lowest plot gives a graph using clustering by rank with variable dimensions using a set regime. The clustering by rank with fixed dimensions was shown in Figure 3

5.4 Applying constraints on pages from a web community

In this experiment, we investigate the effect of a constraint that involves pages belonging to the same site [5]. The question which we like to answer is: are the changes in the absolute position of the web pages caused by the proposed algorithm primarily affect pages from within the same community? In other words, could the changes in the PageRanks mostly come from the same community. This is quite a reasonable hypothesis in that if we wish to swap two pages in the same community, then most of the changes in the PageRanks might come from the same community, and only to a lesser extent come from other web pages un-related to the community.

We carry out an experiment to evaluate this proposition. We chose a community "Stanford University". In our web collection of 150,000 web pages we found 105 web pages which are from the Stanford community.

In order to minimize the effect of clustering, we use 50 clusters, where each cluster is of dimension 2000. The constraint is to swap the position of two pages from the Stanford community, one located at (absolute) position 4377, and the other located at absolute position 6246. The results of the experiment are shown in Figure 9.

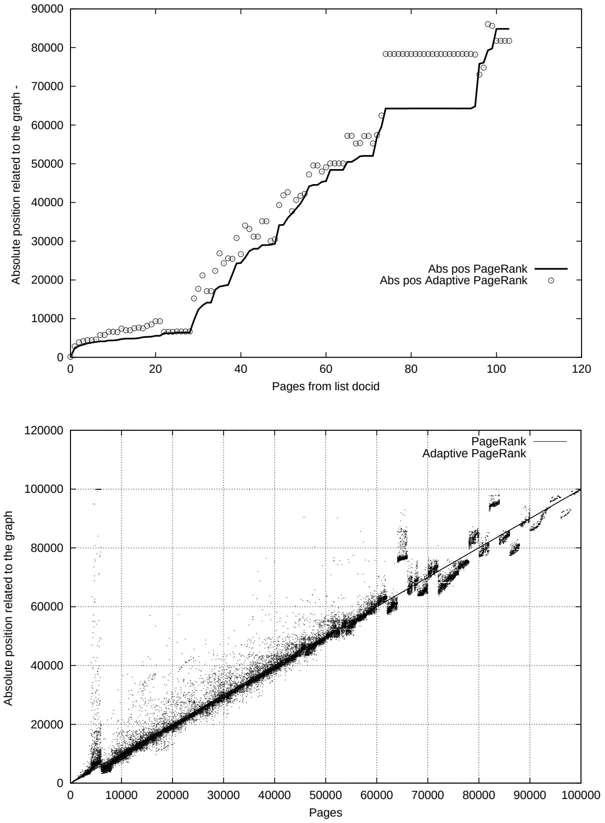


Figure 9: Results of restricting the perturbed web pages to the same community. The graph on top shows the change in absolute positions of web pages related to the community. The bottom graph shows the change in absolute positions relative to all web pages in the collection.

From Figure 9, it is observed that the perturbation of the web pages is not restricted to within a site to which a constraint was placed. Also, we did not observe that the perturbation within a site is more violent than on pages external to the site.

This result can be understood by the fact that even though the Stanford community has many connections among themselves, nevertheless, the web pages are connected to links outside the community. By swapping the ranking of two web pages within the community, it also affects the relative ranking of those web pages external to the community. Hence, it is not surprising to see that the PageRanks of web pages external to the community also change.

5.5 Comparisons with other methods

To the best of our knowledge, the only alternative approach to alter the ranking of web pages from an administrator's point of view has been suggested by Chang et al. [6]. The underlying idea is to extend Kleinberg's HITS algorithm [7] by altering the eigenvector such that a given target page is given more weight. This is performed by using a gradient descent method on the elements of the link matrix W. Chang et al. [6] indicated that their algorithm not only increases the rank of a given page but also increases the rank of all pages that are similar to the given page.

A set of experiments was conducted to allow a qualitative comparison of our proposed algorithm with that proposed in [6]. The first striking difference of the two algorithms is the computational complexity. Chang et al.'s algorithm relies on a recursively applied multiplication of link matrices which resulted in non-sparse matrices. As a consequence, it is unlikely that Chang et al.'s algorithm is able to process link matrices which arise from the live world wide web. In practice, we found that we were unable to execute experiments for $n > 10,000$ on a dual Xeon-2GHz system equipped with 4GB of memory due to memory requirements. In actual fact we had to reduce n even further to allow experiments to be executed within a reasonable amount of time.

We executed the algorithm using sub-sets of size 4000, 7000, and 10,000, and employed training parameters as suggested in [6]. Results are then compared with those obtained by using our algorithm. The result is given in Figure 10. We found that the algorithm in [6] provides an effective method for raising (or lowering) the rank of a given document. However, we also found that other pages which are considered to be 'similar' also rose by rank. This 'similarity', however, is related to the number of inlinks and outlinks rather than to the actual content of a page. Secondly, Chang et al.'s algorithm is not effective for pages which have no inlinks (from within the training set). These pages are generally ranked lowest. In Figure 10 (top left graph), it is observed that only the first 2150 pages feature inlinks[6]. Hence only pages that are ranked high are affected. A further observation was that an experiment required 18 hours for a sub-set of size 4000, and 48 hours for a sub-set of 10,000 pages on the Xeon-2GHz to complete whereas our proposed algorithm completed each run within a few seconds.

Overall, the algorithm in [6] is a simple method which is effective in altering the rank of a given web page from within a limited set of pages. The degree to which a given page is affected cannot be set, nor does the algorithm allow to limit the effect on other pages. In contrast, our proposed algorithm can incorporate these constraints transparently.

[5] A *site* is a collection of pages from the same domain (e.g. stanford.org).

[6] The smaller the sub-set of web-pages, the less connectivity between pages, causing more pages to be isolated.

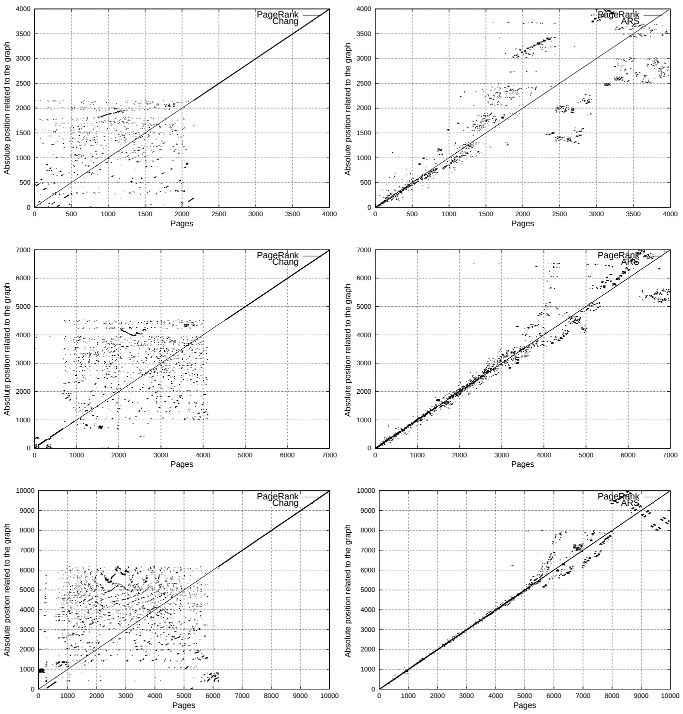


Figure 10: Comparing Chang et al.'s algorithm (left) with our algorithm (right). Rising the rank of page at absolute position 1000 (top row), 2000 (center), and 4000 (bottom). Sub-sets are of size 4000 (top), 7000 (center), and 10,000(bottom)

6. CONCLUSIONS

In this paper, we first consider the possibility of modifying the ranking of web pages belonging to a web collection by allowing the design of a set of control variables. We formulate the problem as a standard quadratic programming problem in minimizing a cost function which is the deviation of the absolute position of the web pages after being processed by the proposed algorithm, and that of the original ranking as given by Google's PageRank algorithm, subject to a number of constraints. Then we carry out a set of experiments designed to evaluate the validity and understand the behaviours of our proposed algorithm.

It is found that it is possible to find a solution to the given task. In addition, it is found that the PageRanks of all web pages are affected when placing a constraint on some of the web pages. The effect on the global PageRanks depends on the rank of the pages to which a constraint is applied, and on the clustering method chosen. If these pages are located in the relatively high ranking range, then the PageRanks of the web pages will be perturbed more violently. On the other hand, even if constraint affected pages are located in the relatively low rank region, it is observed that higher ranked pages may also be perturbed. This is explained by the fact that these highly ranked pages are the ancestors of the lowly ranked pages. Hence by altering the PageRanks of the lowly ranked pages, it is possible that the ancestors of these lowly ranked pages would need to be perturbed as well. The effect of PageRank perturbation is minimized by choosing a sufficiently large number of clusters, where the clusters are formed with respect to the magnitude of the rank of pages. Also, moderate constraints placed on lower ranked pages significantly reduce PageRank perturbations.

An issue which we have not addressed in this paper is the nonlinear relationship between rank and position. This is best illustrated as follows: Ideally we do not wish to perturb the order $\boldsymbol{P}_g$ of the web pages induced by the function $h(\cdot)$ applied on the rank vector $\boldsymbol{X}_g$ computed in (2) (see example in Figure (11). The function $h(\cdot)$ is highly nonlinear.

$$\boldsymbol{X}_g = \begin{pmatrix} 1.2 \\ 2.3 \\ 0.154 \\ 0.72 \\ 1.41 \end{pmatrix} \rightarrow h(\boldsymbol{X}_g) = \boldsymbol{P}_g = \begin{pmatrix} 4 \\ 1 \\ 2 \\ 5 \\ 3 \end{pmatrix}$$

Figure 11: Example of order $\boldsymbol{P}_g$ induced by $\boldsymbol{X}_g$.

Because this mapping function is highly nonlinear, it could happen that while the constraints on the web pages are satisfied, the absolute position of the resulting web pages could be worse off. For example, if we wish to swap the position of two web pages located in position p and q, and $p < q$. After we process this using the proposed algorithm, the two pages are located in positions p_1 and q_1 respectively, with $q_1 < p_1$. It could happen that $q < p_1$. In other words, the overall positions of the two swapped pages are worse off than before the modification. This seems to defeat the purpose of the modification, in that we wish to improve the position of particular web pages, with respect to others. Thus, it would be useful to have an algorithm which will preserve the absolute position of the designated web pages.

Secondly, in our experiments we have only considered a single constraint, viz., to swap the position of two designated web pages. We could have imposed more constraints. However, the picture is more complex, as it would be difficult to draw conclusions on observations on experimental results. It would be useful if there is a more systematic method for finding out the effects of multiple constraints on the modification of PageRanks. These tasks are presented as a challenge for future research.

7. REFERENCES

[1] Bianchini, M., Gori, M., Scarselli, F. "Inside PageRank", Tech. Report DII 1/2003, University of Siena, Italy, 2003.

[2] Brin, S., Page, L. "The anatomy of a large scale hypertextual web search engine". *Proceedings of the 7th WWW conference*, April, 1998.

[3] Ng, A.Y., Zheng, A.X., Jordan, M.I. "Stable algorithms for link analysis", *in Proceedings of IJCAI-2001*, 2001.

[4] Zhang, D., Dong, Y. "An efficient algorithm to rank web resources", *in Proceedings of the 9th WWW Conference*, Elsevier Science, 2000.

[5] Gill, P., Murray, W., Wright, M., *Practical Optimization.* Academic Press, 1981.

[6] Chang, H., Cohn, D., McCallum, A.K., "Learning to Create Customized Authority Lists", *Proc. 17th International Conf. on Machine Learning*, 2000.

[7] Kleinberg, J., "Authoritative sources in a hyperlinked environment", *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.