Implementation of the Multidimensional Max-Flow Algorithm, Software Agents in Real time PageRank application

A graduate project submitted in partial fulfillment of the requirements

For the degree of Masters of Science

In Computer Science

By

Prathyushi Nangia

December 2011
The graduate project of Prathyushi Nangia is approved:

____________________________________  ______________________
Dr. Robert McIlhenny  Date:

____________________________________  ______________________
Dr. Shan Barkataki  Date

____________________________________  ______________________
Dr. Gloria Melara, Chair  Date:

California State University, Northridge
This project was handed over to me by Dr. Gloria Melara. She has been a constant support pillar for me ever since I started my journey as a student of Cal State Northridge. The project is an abstract idea of Ms. Priyanka Bihani who was a student of the Computer Science Department. This MMSA project deals with the implementation of an existing real world application after its integration with the MMSA model. The MMSA model as proposed by Ms Bihani involves optimization strategies which when integrated with the current real world available ready to use applications can improve performance. With this objective in mind I set forth towards implementation.

I have to thank a lot of people who have been active contributors for the successful completion of this project. To start with, I would like to thank God for always watching over me through thick and thin.

Most importantly, I would like to thank Dr. Gloria Melara for having the un-dying faith in me to hand over such an important project to me. My sincere gratitude is also extended to Dr. Shan Barkataki and Prof. Robert McIlhenny for always having the patience to guide me and correct me when I took the wrong path.

Next I would like to show my gratefulness to my mom (Mrs. Meenaxi Nangia) because of whom I stand as an individual with a track record of successful endeavors. She has always been a source of inspiration for me. I would also like to thank my siblings who have always showed confidence in my decisions and have been supportive all throughout my life.

Last but not the least; I would like to thank my love, who always lent that extending hand when I reached out for it in times of distress. Also not to forget my teachers and professors in India as well as in the United States without whom my knowledge would be incomplete.
My life would be an incomplete picture puzzle if I missed a single piece from the above mentioned.

I am truly indebted to you all.
TABLE OF CONTENTS

SIGNATURE PAGE ........................................................................................................ ii
ACKNOWLEDGEMENT ............................................................................................... iii
ABSTRACT .................................................................................................................. vi
OBJECTIVES ........................................................................................................... viii
OVERVIEW ............................................................................................................. ix

INTRODUCTION ........................................................................................................ 1
  Insight 1.1 .................................................................................................................. 1
  The Working Papers – Flowchart & Algorithm 1.2 ................................................. 3
  Analysis of the MMSA Model Algorithm:Algorithm 1.3 ...................................... 6
  Software Agents 1.4 ............................................................................................... 9
  The Max flow Algorithm 1.5 .................................................................................. 11

THE PAGERANK ALGORITHM .................................................................................... 15
  The Insight 2.1 ........................................................................................................ 15
  Markov Chain Algorithms – The Inspiration 2.2 ................................................... 17
  Evolution 2.3 .......................................................................................................... 18
  Key Concepts 2.4 ................................................................................................... 19
  PageRank in Action 2.5 ......................................................................................... 22

THE PAGERANK APPLICATION .................................................................................. 24
  The Real world Application 3.1 .............................................................................. 24
  The Application steps 3.2 ....................................................................................... 29
  Application Summary 3.3 ..................................................................................... 29

MODIFIED APPLICATION – INTEGRATION OF THE MMSA AND PAGERANK ....... 30
  Insight 4.1 .............................................................................................................. 30
  Modularization 4.2 ............................................................................................... 30
  The NEW MMSA process 4.3 .............................................................................. 36
  Application – Explained 4.4 .................................................................................. 43

FEASIBILITY & COMPARISON SCENARIOS .............................................................. 49
  Feasibility of the Model Proposed 5.1 .................................................................. 49

FUTURISTIC APPROACH ......................................................................................... 60

CONCLUSION .......................................................................................................... 61

REFERENCES .......................................................................................................... 64
ABSTRACT

Implementation of the Multidimensional Max Flow Algorithm Software Agents Model

By

Prathyushi Nangia

Master of Science in Computer Science

The MMSA model works closely with optimization of algorithms and applications. The difference between an application and the MMSA integrated application is the ability of the MMSA to provide optimized solutions to the already existing one. The key ingredients of the MMSA model are software agents and the max flow algorithm. The successful implementation therefore depends on choosing the right Software Agent type and the best variation of the Max Flow algorithm.

The aim of this project is to take a PageRank application into consideration and evaluate two variances – one could be called its original form as used by us at the present, and the other could be the optimized form or the modified form obtained after implementing the MMSA model. These two variances are then evaluated on performance basis.

The final step of the project was to prepare a data analysis sheet that could help us identify quickly and without fail the differences in performance between the original application and the modified application. The testing of the modified application was based on a list of elements that affect the performances of an application. A detailed comparison study was carried out to get appropriate information that satisfies the tester and the observers.

Thus, the project has a process that consists of phases. The normal software development life cycle has been followed but the approach was more agile in nature. The application has been through a lot of ad-hoc and exploratory testing. The application has been developed in
JavaScript. There is no database involved. The backend has been coded in HTML, JQuery 1.6.2 and JavaScript. The frontend or the user interface has been designed in terms of HTML and CSS. The graphs are generated using the software called the Fusion charts. This tool has been developed in JavaScript and is embedded in the MMSA application code. The MMSA is a web application and has been hosted on a server. The proposed browser to use this application is Mozilla Firefox. To access the application please visit:

http://prathyushinangia.com/project/home.html
OBJECTIVES

“Testing the Multidimensional, Max Flow algorithm and Software Agents (MMSA) model in a real time PageRank application and evaluating on the basis of the data obtained, for verification of the abstract theoretical model.”

However, a combination of successful fulfillment of these objectives was required to achieve the stated objective.

- Search for applications that calculate the PageRank of a given web page, compare such applications and select the best criteria for MMSA implementation.
- Research about which Software Agent and Max Flow version should be used for the implementation.
- Compare the algorithm and flowchart of the real world PageRank application with that of the MMSA model.
- Calculate the complexity of the MMSA implementation.
- Study the differences in distributed and sequential implementation of the MMSA model.
- Design the UI for the application.
- Implement the PageRank application.
- Compare the original PageRank and the MMSA integrated application for analysis.
OVERVIEW

The main contribution of this model is to validate the successful implementation of the MMSA model and to ensure this implementation optimizes the originally obtained solutions. The objectives of this project are achieved in chapters 1 through 7. Chapter 1 deals with the introduction MMSA model and its implementation strategies. Chapter 1 also introduces the various technologies used to build an application that embeds the MMSA nature - the Max Flow algorithm and Software Agents. Chapter 2 describes the PageRank algorithm and all the relevant details that should be known before implementing a full-fledged application into work. The PageRank application terminology and methodology is also mentioned. The chapter 3 deals with the detailed PageRank application available on the web. It also helps visualize what an application would look like, including the nitty-gritty information of MMSA implementation. Chapter 4 explains the principles of the integration of the PageRank application and the MMSA model. It deals with the modified (MMSA integrated) application implementations. Chapter 5 includes the feasibility and comparison scenarios of the MMSA PageRank application. Chapter 6 details out the summary and the futuristic approach of this project implementation. Chapter 7 includes conclusion of the project. References are also included.
INTRODUCTION

The web world has grown far more than we could have imagined a decade back. The progress has been spectacular just like the advancements in the other spheres of our day to day life. We are always on the run to find solutions, alternatives and expansions. Improvements in a specific sense could be optimizations.

The MMSA model is a new robust multidimensional model that provides a powerful problem solving methodology. A popular graph based algorithm called the Max Flow Algorithm, and Software Agents, a field in Artificial Intelligence, both have various applications and are the major components of implementation of the Multidimensional Max Flow Software Agent (MMSA) model. The MMSA model shows automation and implements a cooperative behavior between the synchronization of other agents. This autonomous process works without human interference by agents carrying out their assigned tasks. The MMSA process is as shown below in figure 1.1. The figure gives an overall process of the MMSA model, which is a suitable problem-solving approach for various applications. It illustrates the order and flow of events, criteria, parameters, graph layouts, validation, and optimization. These are key concepts in explaining the way the MMSA model works. Here, events, criteria, and parameters are application dependent. The event is a real world phenomenon that initiates the MMSA model process.
A criterion specifies the goal for the model to satisfy. The negotiation process selects the relevant parameters based on a criterion to generate graph layouts. The setup is of a directed graph with dynamic vertices, weights on edges, and modeling movement of flow. The functionality derived from Software Agents pertains to the behavior of cooperation and primarily negotiation done by interaction among the agents in a coordinated way to accomplish goals. The negotiation function is an important agent function. Having one or more applicable parameters generates graphs for each parameter. These graphs are generated and then validated. After the validation process, the graphs are optimized using the Max Flow algorithm.
The MMSA gets the optimization capability from the Max Flow Algorithm. The Max Flow is a mathematical model that provides data optimization standards. This mathematical model is used explicitly in graph theories, to calculate weights, capacities, distances and differences that exist between the nodes present in a graph.

The data that is collected to be used as a graph for the Max Flow algorithm input is collected using Software agents. Software Agents are used to automate the search criteria and hence limit the number of options available in the direction the user chooses. Software Agents assist in data analysis and data collection for the application, as any optimization algorithm requires data, certain criteria and events that occur to trigger the Software agents to start action.

1.1 Insight:
The Multi-dimensional Max Flow Software Agent Model is an optimization algorithm. This model was developed with the objective of enhancing performance, improving accuracy, mitigating longer time durations required for calculations.

The MMSA model involves the use of two methodologies. The integration of two vital components – Max Flow algorithm and Software Agents leads to the creation of a framework for a new robust, multidimensional model called MMSA (Multidimensional, Max Flow Algorithm, and Software Agents) model. The MMSA Algorithm involves the terms like events, parameters, criteria, graph layouts, optimization, problem solving, and validation. These terms together combined makes the process flow of the MMSA model. This model exhibits flexibility and dimensionality. The synergy of the integration of the two major components of the MMSA model widens the scope of problems analyzed and provides
optimized solutions for these problems under consideration. Max Flow algorithm is the popular graph based algorithm and software agents are an upcoming field in Artificial Intelligence. Both these have different applications and are the building bricks of this project.

---

**Figure 1.2: "Levels of the MMSA Model"**

The set up required is of a directed graph with dynamic vertices, weights on edges and modeling movement flow. The dynamic vertices are vertices in a directed graph modeled as software agents to provide a powerful functionality. This functionality derived from software agents pertains to behavior of cooperation and primarily negotiation is done by interaction among the agents in a coordinated way to accomplish goals. The negotiation process is an important agent...
function carried on, while interacting with the vertices for generating graph layouts. Having one or more applicable parameters generates one or more graphs for each parameters thereby adding dimensionality to the model. The valid graphs generated are optimized using the max flow algorithm. The Max Flow Algorithm is a greedy algorithm by nature that tries to determine an optimum solution. Applying the Max Flow Algorithm determines an optimum solution.

To start with laying out a design structure for the Web application, along with integrating different modules into it we design our flowcharts to get our data flow and the Algorithm to obtain a pseudo code framework.

1.2 The Working Papers – Flowchart & Algorithm:

Two important outputs – Flowchart and Algorithm are required to work on a system after the analysis of what methodologies and technologies will be used has been done. The designing phase requires details on all the aspects that could impact the project in either a positive or a negative manner. Risk analysis has to be done very often. The approach we have applied in the development of this project is agile in nature. The development phase involves a number of deliverables. Amongst all the products delivered during this phase, the Algorithm and Flowchart are critical.

Algorithm -

This algorithm implementation can give optimized results when using the sequential approach. When implementing this algorithm in a distributed approach the improvement in the algorithm execution time does not make a huge difference. Instead it incurs the cost of the additional processors required for the distributed implementation considering the data sets are small. However, the difference can be seen if we have as an input a large data set. For large data sets, the implementation costs for more than one processor can be mitigated as the work process
increases in terms of time. The difference in the execution time cannot be further enhanced by using other alternatives, processors or iterations.

Algorithm of the MMSA Model:

MMSA Model Algorithm (G, s, t):

1. While there exists an event M
2. Do generate graphs (Graph generation)
3. If graphs are valid (graph validation)
4. Apply the Edmonds–Karp on the Max Flow algorithm (Graph Optimization)
5. Return optimum value of Max (of all the flows)
6. Else
7. Return (“ No Solution exists “)

1.3 Analysis of the MMSA Model Algorithm:

The MMSA model algorithm is an iterative algorithm. In line 1, it begins by generating a random event M. Once an event is generated, it is then time for the negotiation process to get graph layouts done according to line 2. The negotiation process selects a criterion related to the event. One or more parameters are then selected by the negotiation process. The broadcast action of the negotiation process connects the vertices to form directed edges and assigns capacities (or weights) on the edges based on parameter values. Then in line 3, the graphs that are generated are validated. In line 4, assuming the graphs generated are valid, the Edmonds-Karp Algorithm version of the Max Flow algorithm is applied. A flow value is obtained from each graph generated. In line 5, the optimum flow is given by returning the maximum value of all flows, denoted as, \( \text{Max} (\bigcup_i f_i) \). Now, as mentioned earlier, since an event is like a real world
phenomenon, which can happen at any time, it may or may not trigger a solution for the Max Flow algorithm. So if the graph that was generated in line 2 is not a valid graph (determined in line 3), then it goes to line 7, as the Max Flow algorithm does not apply to it and returns a message that “No solution exists.” Next, again when an event occurs, the whole while loop (lines 2 to 7, as applicable) will be run again until there are no more events remaining.

Flowchart -
The phases of the MMSA model are shown with the help of the flowchart for describing the MMSA process. The process follows the general approach of checking first for an event that triggers the MMSA process. If any such event is encountered, a criterion K is followed and thus parameters are recognized and collected to help get data to generate graphs. After the parameters are identified one or more graphs are created for a parameter. Thus, we need to collect data input from the user and if an event occurs. This event leads to data collection for a specific criteria and parameters. Data is narrowed down based on all user selections. These graphs generated through the narrowed selections are then provided as input for the Max Flow Algorithm. These graphs undergo the process of generation, validation and optimization. Max Flow algorithm comes into play when the graphs are validated and ready to be optimized.

The MMSA Flowchart is as shown in figure 1.3.
Start

Is there an event?
Yes

Select a relevant criterion

Select one or more relevant parameters

Is there a parameter?
Yes

P_i-D Graph Generations
For each parameter generate a graph layout

P_i-D Graph Validations
Validate each graph layout

Is there a graph?
Yes

P_m-D Graph optimization
Optimize all graph layouts by applying Max-flow Algorithm

Selection of optimum value $\text{Max}(U_{fi})$

Stop

NO

NO

No

Figure 1.3: "MMSA Flowchart View"
1.4 Software Agents:

An **Agent** is authorized to act for another agent or human. Agents possess the characteristics of delegacy, competency and amenability. The agent receives percepts and accepts inputs from the user or other agents. They have a percept sequence and can sense a change. A percept can be defined as a complete set of input at a given time. An agent program creates a mapping between percepts sequences and actions. Agents have a specific goal.

![The Agent Cycle](image)

**Figure 1.4: “The Agent Cycle”**

*Software agent* can be defined as a piece of software that acts for a user or another program in relationship with an agency. An agent operates and acts in an environment. A Software Agent is an artificial agent which operates in a software environment. A **software environment** could be
an operating system, computer applications, databases, networks, virtual domains or a combination of two or more of the individual environments.

There are a number of different software agents depending on the type of assistance they offer. One of them is an Intelligent Software Agent.

![Figure 2.2: "A part view of an agent typology"](image)

An **Intelligent Software Agent (ISA)** is an agent that uses Artificial Intelligence to accomplish the goals assigned to it. Artificial Intelligence is an upcoming science that imitates the human presence. Artificial intelligence has given the world a new set of ideas for implementation as human workload can be reduced. Intelligent software agents are designed to work with human – like intelligence. Intelligent software agents or ISA’s are significant contributors in making tasks easier for the humans.

Intelligent agents are autonomous agents and cooperative in nature. A list of the human-like intelligence capabilities that may be possessed by intelligent software agents for the problem domain, include some of the most difficult areas of research that is being pursued in Artificial Intelligence (AI) to date.
Collaborative Agents -

Collaborative agents, as the name suggest collaborate with each other to get a task done. These agents interact with each other to share information. This information is related to the specialized services that they are capable of offering. While each agent may uniquely speak the protocol of a particular operating environment, they generally share a common interface language which enables them to request specialized services as and when required.

Thus the word "agent" by itself generally connotes ISAs in the terms of the present-day research community.

1.5 The Max flow Algorithm:

The Max Flow Algorithm is categorized as an optimization algorithm. This is a computational algorithm. This Algorithm takes as an input a directed graph and is considered a network. The Max Flow Algorithm when applied gives an optimal path or route in the network of graph under consideration. Thus in generality, the Max Flow Algorithm calculates the flow in a network. The flow could consist of any commodity that can be passed in the network. The network can be transportation routes, cables for LAN connections, router placements for Wi-Fi, airline ticketing systems, cargo shipments and similar others. This calculation is based on the weights and capacities of the directed graph. Every edge included in the graph has an associated weight that is independent of the other edges and a capacity which is dependent and related to the path selected. The max flow algorithm has the directed graph as its input and the residual graph as the output. The residual graph has arrows in both the directions as one shows the capacity of the edge and the other direction shows the weight of the edge.
Example of the Max Flow Algorithm -

An example modeled after an application consisting of a flow network is shown in figure 1.5 to demonstrate a company A’s web site link structure. This company is a service provider hence it wants to optimize the pages that share data about services. It is represented as a flow network $G = (V, E)$ with source $s$, being an imaginary page which has no incoming links, and sink, $t$, being a page which has no outgoing links. The company provides services to customers and hence wants to have pages which talk about these services and product lines having higher PageRank. These pages are considered as the nodes in the network and the associated PageRanks are the weights assigned to them initially. The maximum PageRank that any link can possess can be infinity (a general case) and 10 (Google) this can be defined as $c(u, v)$. Company A has an existing link structure, in which each page has an initial PageRank associated with it. The aim is to find the maximum PageRank that can be given to each individual page as to increase the PageRank of the owner critical pages.

The above company A’s PageRank assignment problem seems suitable for modeling flows pertaining to the entire web structure, or in this case a small web site. The amount of PageRank that page A can pass to another page cannot exceed the PageRank that Page A has. This helps in maintaining a stable state while also maintaining the network flow i.e. the weights in the network cannot exceed the capacity of the network. The skew symmetry property as stated earlier pertains mostly for notational convenience. A real world example using the service provider’s web site link structure is shown in figure 1.5. This can be viewed as a net positive flow resulting from PageRank being transferred from one page to another in a web structure. Also remember that a page cannot link to itself, as this is not a valid case in the Max Flow implementation.
In figure 1.7 “each edge is labeled with its capacity”, and “only positive flows are shown”. “If \( f(u, v) > 0 \), edge \((u, v)\) is labeled by \( f(u, v) / c(u, v)\).” Here, “/” is used as a separator with flow to the left of it and capacity to the right of it.

The traversal of the flow network in figure 1.7 is done using the Edmonds–Karp algorithm. The Edmonds–Karp uses a slight variation of the Ford-Fulkerson algorithm. This variation uses a breadth-first search implementation approach for determining an augmenting path. The rest of the steps remain the same as in the Ford-Fulkerson algorithm.

Figure 1.5: The link structure of a web site represented in the form of a graph
Figure 1.6: The Source and Sink nodes joined to a graph structure

Figure 1.7: The PR assigned to each link (Weights of the edges). This graph will be used as the input to the Max Flow algorithm
THE PAGERANK ALGORITHM

2.1 The Insight:

In this rapid paced world, would you like to be left behind? The answer to this obvious question is NO. The Google search engine evolved with a similar concern. Besides ease of use and high performance, the quality of the search results was one of the key factors that lead to the development of the Google search engine. The concept of link popularity was developed after a detailed analysis of the content specific page ranking criteria. This new concept of link popularity dealt with the number of inbound links on a web page. The idea was more the number of pages linking to your page; more is the importance of your web page. This is a simple calculation. However, web sites then came into existence which had any number of random links in order to increase the number of inbound links present on a web page. However, these fake web sites could not survive as the PageRank not only calculates the number of inbound links but also the PageRank that comes with it. All the inbound links do not count equally. So to make it simple, a page ranks high in terms of PageRank, if the documents having a high PageRank link to it.

PageRank thesis constructs page ranking based on the importance of inlinks and outlinks as opposed to the HITS Algorithm working based on the number of inlinks and outlinks. PageRank is associated with the popularity and genuine content on a web page. The underlying assumption in the PageRank thesis is that a web user is more likely to visit more important pages. It is purely a measurement for calculating the relative importance of web pages, but not the only one. The Page Rank algorithm assigns a numerical weighting to each element of a hyperlinked set of documents. The purpose is to measure the importance of this particular document within the set. This numerical weighting that is assigned is referred as PageRank (PR).
Thus, a PageRank for a given element E is denoted as (PR) E, where PR is the PageRank for the element E, where E could be any web page.

This is a web based algorithm which calculates link analysis. Link building and back-links are the two important aspects which should be understood clearly when learning about PageRank.

The PageRank is a mathematical model based on the graph theory containing nodes and edges. To calculate PageRank, we must begin by building a mathematical model of the link structure of the web. We can construct an adjacency matrix L from the graph of the web where

$$L_{ij} = \begin{cases} 
1, & \text{if there exists an edge from node i to node j} \\
0, & \text{Otherwise}
\end{cases}$$

Therefore, when a matrix for the link structure is generated it consists of 1’s and 0’s. This matrix can then be used for a graph generation. Therefore, consider the web world as a graph, where every web page that exists is considered as a vertex or a node for the graph. If there is a link present between two web pages, this link is then considered as an edge between the two connected vertices. Thus PageRank is achieved by calculating a rank for every web page that exists on the graph of the web. The theory that the web is considered as a graph with the vertices symbolizing a web page and the edges being the hyperlinks that are connected to the web page is the underlying basis of the model that is proposed.

PageRank is an important aspect when it comes to ranking a web page. The Google search Engine ranks a web page related to the search key phrase term entered by the user. Higher PageRank results in an article appearing at the top of the search results. These search results obtained indicate the importance of a particular web page. A hyperlink to a page in the web world counts as a vote of support.
2.2 Markov Chain Algorithms – The Inspiration:

A Markov Chain Algorithm is a mathematical model and deals with transition from one state to another. Transitioning is defined as the change of one state to another. The transition process starts at a state and progresses through successive states. This transition occurs in finite or countable states. Each state is obtained using the probability matrix also called the Transition Matrix. The transition either remains in the same state or can change its current state. The next state depends on the current state solely and has no influence of the preceding set of states. This is called the Markov Chain property. Markov Chain Algorithm works on Markov Chain property. This transition matrix stores the probability of the occurrence of an event. This Transition Matrix is stochastic in nature. A Stochastic Matrix is a non-negative matrix where the sum of each row is equal to one. To be more specific, these matrices are also called Row Stochastic Matrices as opposed to Column Stochastic Matrices where each column is equal to one. Thus, Markov Chain is a stochastic process that occurs in a finite time. The events in the transition matrix are known already and have a probability associated with each individual event. The Markov Chain algorithm has many variations and has been applied in many disciplines of study. It has been successful in implementing the PageRank algorithm for Google. The PageRank algorithm has been using this mathematical component to calculate the PageRank vector. Markov Chain Algorithms are used as a statistical tool for the analyzing the trends. This is a tool used to track the behavior of the user in the web world. The user navigation trends are tracked and studied to obtain information about the web structure and user navigation interests. The probability of navigating from one page to another is called the transition probability. The transitions are referred to as “memory-less” transitions as the transition to the next state depends only on the current state and not that of the previous set of states as stated earlier. This can be
depicted with the help of a transition state diagram or a tree diagram that calculates the probability.

![State Transition Diagram](image.png)

Figure 2.1: A State Transition Diagram showing the state transitions along with the transition properties

Coming back to the PageRank algorithm, Brin and Page defined this Markov Chain Algorithm in their terms as a Random surfer Model. This model describes the events of surfing the web of a random surfer from state A to state B.

### 2.3 Evolution:

Consider the web as big as it is today; now assume that Google did not exist. Back in the last few decades we had the web and its resources but the search key phrase would not surely return what you wanted. Then came yahoo search engine and users relaxed. But it did not solve their problem entirely. The name "PageRank" is a trademark of Google, and the PageRank process has been patented (U.S. Patent 6,285,999). However, the patent is assigned to Stanford University and not to Google. Ever wonder why?
PageRank was developed at Stanford University by Larry Page and Sergey Brin as a small part of a research project. Thus, the term PageRank was coined after Larry Page. This research project was about a new kind of search engine, one that would address the need of the hour. Sergey Brin had the idea that information on the web could be ordered in a hierarchy by "link popularity". The concept is defined as - a page is ranked higher if there are more in bound links to it. The first paper about the project was published in 1998. It was co-authored by Rajeev Motwani and Terry Winograd. The paper described in depth the PageRank concept and the initial prototype of the Google search engine. Shortly after the publication, Page and Brin founded Google Inc., the company behind the Google search engine. The PageRank technology is just one of the many factors that lead to the ranking of Google search results. However, the concept of PageRank continues to provide the basis for all of Google's web search tools.

2.4 Key Concepts:

There are a number of elements that could make your understanding of the PageRank a little difficult. Such terms associated with the algorithm are discussed.

Required concepts:

1. Damping factor:

Consider a user randomly surfing the internet. The damping factor is calculated as the probability that the person continues to surf the internet. This damping factor is usually considered to be 0.85 as a fixed value. This number is obtained as a web page is thought of at least having a link to another web page on its site. So some value of the PageRank is passed to the other pages on the web site.
2. Random Surfer Model:

The Random Surfer Model can be best described as the probability of reaching a random web page from a web. This probability is derived from the respective page’s PageRank. The probability that a random link present on a web page is clicked depends on the number of links present.

3. Eigenvector:

Eigenvectors of a square matrix are the non-zero vectors that after being multiplied by the matrix remain proportional to the original matrix. They change only in magnitude not in direction.

4. Adjacency matrix:

Used in Computer science to represent which vertices of a graph are adjacent to which other vertices. Adjacency matrix of a finite graph $G$ on $n$ vertices is $n \times n$ matrix, where the non-diagonal entry $a_{ij}$ is the number of edges from vertex $i$ to vertex $j$ and the diagonal entry $a_{ii} = 0$.

5. Link analysis:

Link analysis in general terms deals with the links that exist on a web page. These links are to other web pages on the web, thus every link would have a PageRank associated with it. The more important or relevant the link is in terms of the search key-phrase entered, higher is the PageRank.
6. Link building:

Link building is the process of creating inbound links to a website. So it can also be defined as the practice of obtaining links from other external web sites available on the web to your own.

7. Link farms:

A link farm is a group of web sites that all hyper-link to every other site in the group, thus providing ample inbound and out bound links to all the participants. Also these groups usually consist of a similar sphere of web sites so that the links are not ignored by Google Search Engine. The Google Search Engine does not like links of unrelated content to be available on a web site and does not count these links when calculating PageRank.

8. In bound links:

In bound links (IBL) are links that other web site have to your web site. Search engine optimization largely depends on the linking capacity of a web site. Along with this, the natural traffic is a bonus.

9. Out bound links:

An outbound-link (OBL) is a link from your website to another website. It is inescapable that if the inbound links of a page influence its PageRank, its outbound links do also have some impact. Outbound links and in bound links are inverse in nature. These links can add value to your site.
10. Dangling links:

The lack of outbound and inbound links on the web is an important aspect. When a web page has no outbound links, its PageRank cannot be distributed to other pages. Such links are called dangling links. Dangling links are analyzed in detail later in chapter 5. This analysis is based on the MMSA PageRank implementation and the reduction of dangling links from the web structure.

2.5 PageRank in Action:

Suppose there are only 5 pages on the web, as shown in the figure 5.1 below. Initially, PageRank (PR) is equally distributed among these pages for a value of 1. So, every document has a (PR) value of 0.20 (based on the probability of 0 to 1).

(PR) A = 0.20
(PR) B = 0.20
(PR) C = 0.20
(PR) D = 0.20
(PR) E = 0.20

Now, if pages B, C, D and E all link to only page A (meaning all links from four pages pointing to A), then each gives PR of 0.20 to page A. This makes the PR of A 0.80. Let B have a link going to page D. Then, page B has a PR value of 0.10 going to page A and 0.10 going to page D (0.20 is divided by 2, as it has links going to pages A and D). Now, if page C has links that go to all pages (0.20 is divided by 4, as it has links to pages A, B, D, and E). This is a simplified method of assigning values to web pages.
PageRank calculation is an iterative process. It gets accurate as the number of iterations increase.

The question is what should be the initial PR assigned to a web page. An initial value is required to calculate the current PR of a web page.

Figure 2.2: "Simplified Workings of the PageRank"
There are a number of web applications available to calculate the PageRank of your web site. These applications are simple to implement and add a lot of value to our web sites. The most famous known PageRank application is the Google PageRank which is the first result of a search for PageRank’s. Let us go through the procedure to obtain Pageranks for the pages of a web site under consideration.

3.1 The Real world Application:

The PageRank Calculator is a variation to the PageRank application. This is a simple calculator that enables you to calculate the PageRank of each web page inside your web site. A change in the web page content can change the PageRank of the web page. This implementation of the PageRank application has slight changes from the original PageRank concept.

After comparing many available applications on the net, Mark Horrell’s application was selected. This selection was done on the basis of the fulfillment of all the aspects such as a real world web application, which can be related to graph theory and networks, etc. required to implement the MMSA model with this application. His is a step by step process to obtain the PageRank of the web page under consideration. This application can be visited at http://www.markhorrell.com/seo/pagerank.asp this application has been made using HTML and ASP, where HTML is used to provide the framework and ASP is used for functionality purposes. We will look into the examples for both odd and even number of pages. For now let us just consider the odd number of pages as an example.

3.2 The Application steps:

The application goes through phases as described in the MMSA model:
Enter the number of pages in your web page ➔

The Google PageRank Calculator

The PageRank Algorithm - Click here for some thoughts on PageRank, and operating instructions for the PageRank Calculator.

About the PageRank Calculator - A brief explanation and some further resources.

Step One: Please input the number of interlinking pages to be analysed (Max. 50)

Enter the number of Web Pages in your web site ➔

Click here to start again.

Figure 3.1 "Step 1"

List the names of the pages that are included in your web site ➔
Figure 3.2: "Step 2"

Specify the links that exist on the web sites individually →
Select the Damping factor D and the number of iterations →
Assign initial PageRank values to each web page →
Step Three: Please assign the links each page points to and a starting value for its PageRank

<table>
<thead>
<tr>
<th>Page No</th>
<th>Page Name</th>
<th>Links to:</th>
<th>Initial PageRank</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Home</td>
<td><img src="image" alt="Home Link Diagram" /></td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>About Us</td>
<td><img src="image" alt="About Us Link Diagram" /></td>
<td>4.7</td>
</tr>
<tr>
<td>3</td>
<td>Services</td>
<td><img src="image" alt="Services Link Diagram" /></td>
<td>2.65</td>
</tr>
<tr>
<td>4</td>
<td>Products</td>
<td><img src="image" alt="Products Link Diagram" /></td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>Contact Us</td>
<td><img src="image" alt="Contact Us Link Diagram" /></td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>Forum</td>
<td><img src="image" alt="Forum Link Diagram" /></td>
<td>3</td>
</tr>
</tbody>
</table>

Step Four: Please assign the damping factor $d$ and number of iterations

<table>
<thead>
<tr>
<th>Damping Factor $d$</th>
<th>Iterations of Equation (Max. 100)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85</td>
<td>20</td>
</tr>
</tbody>
</table>

Submit

Figure 3.3: "Step 3, 4, 5"
Figure 3.4: "The Iterative PageRank calculations"
The calculated PageRank is then listed depending on the initial PageRank assigned to individual pages, the Damping factor D and the number of iterations.

3.3 Application Summary:
This application has been deployed on a server and is a part of the Mark Horrell’s blog. In step 1, the application requires that the user enters the number of pages available within the website that is under consideration. After you enter the number of pages, the application takes you to the next step, where you give equivalent names to the pages of the web site. In step 3, the pages with their names appear and you have to select the pages that are linked to the page selected. Thus, for example when you consider the home page which is in turn connected to About Us page and FAQ page you will select the About Us page and FAQ page from the list of pages available for Home page for selection. After the links have been defined, the user needs to provide three critical values to the application. These three values are the Damping factor D, the initial page ranks of the individual pages and the number of iterations. After assigning values for the respective factors the user hits the submit button to obtain the PageRank values for the pages that have been included in the list for the number of iterations that were set. These iterative values are vital as the PageRank process is iterative and improves as the number of iterations increase. Thus, at iteration 3 you would have a PageRank assigned to a web page but it would not be as accurate as the PageRank value at iteration 15.
MODIFIED APPLICATION – INTEGRATION OF THE MMSA AND PAGERANK

4.1 Insight:
The entire work of integrating and making an application MMSA specific is the combination of the original PageRank Application and the principles of MMSA process. The MMSA PageRank application has been designed on the outlines proposed for the MMSA model. There have been changes, however wherever required for implementation purposes. There have been several integrations of the technologies that are being used. The current modified application has been designed in steps to make is easy for the user to understand the function that happens with each associated step. It makes the process and data flow very transparent. Each step in the MMSA algorithm has certain vital role. Thus, the application has to be modified from its original algorithm to incorporate the MMSA model.

The MMSA PageRank application can be defined as follows:

4.2 Modularization:
The important factors that are included in the MMSA PageRank application are the Software Agents, Max Flow Algorithm, and the original PageRank algorithm. Thus, the MMSA model comprises of other subordinate models. Each of the sub models contribute for the successful integration and synchronization. This leads to the accomplishment of a model, which is to be used for optimizing solutions in the problem solving domain. The flow chart 2.4 explaining the MMSA model has been broken down into modules – these modules provide a better insight into the functionality and data flow in the MMSA model. They also specify the input, output and the sub model used for a specific model. For ease of reference we refer to the modules as boxes as shown in the flowchart.
The first module uses Software Agents and collects data from the user based on certain criteria and parameters.

Figure 4.1: Module one - data collection using Software Agents
The second module uses the output of the first module and then generates graphs. The first graph is user generated. The others are randomly generated to have different link structures based on certain restrictions that the Max Flow input has.
Figure 4.3: Module two - Graph generation using a random generator function

Graph generation is done using the initial data entered by the user and provides a basic structure for the web site. This initial structure is then used to generate random graphs. These graphs are non-reciprocal and do not have cycles.

Figure 4.4: Information about the Graph Generation Process

The data collected in the previous phase is converted into a graph that is then used as the initial input of the Max Flow Algorithm.

The web application generates 4 random graphs at present in addition to the initial graph data entered by the user.

The initial graph is generated using the user input via the Software Agents. However, the other graphs are randomly generated each of which has a different link structure.

All these graphs have a different link structure. Also none of these graphs can be reciprocal or self-linked as Max Flow cannot be applied to such graphs. In addition, these graphs cannot have cycles. Thus, when random graph generation takes place it drops the structures containing cycles.
Module three takes as input the graphs generated and applies Max Flow Algorithm to each. This returns the Max Flow for the different layouts along with the updated link structures.

The web world is considered a directed graph $G (V, E, t, s)$ where

- $V$ = the set of web pages on the web
- $E$ = the set of hyperlinks present on the web pages on the web. These hyperlinks consist of both the incoming and outgoing links
- $t$ = Sink vertex which has no outgoing links. This is provided by the page rank algorithm
- $s$ = Source vertex which has no incoming links to $S$. This node $S$ is connected to the existing graph by joining it with an existing node through the negotiation process.

This assignment can be the basis why the Max Flow can be applied to a PageRank application.

**Figure 4.5: The Graph Optimization process in module three using the Max Flow Algorithm**
The last module takes the optimum Max Flow calculated and uses it as the initial input to the PageRank Application. This data, in terms of the link structure and the Pagerank values associated with the edges are then used in the Mark Horrell’s PageRank application. The application process inside this application is the same as the previous execution. The only difference being the optimum input.

**Figure 4.6: Module four - the input to the PageRank application**

The MMSA PageRank application has two added key points that the original PageRank application does not mention.

Power Pages / Nodes: These pages are identified as having a connection with the source node. As the source node has a capacity that equals to infinity, the source nodes are usually those pages that have a higher PageRank. Such pages are required to share their high PageRank with the pages that have to be optimized based on the input from the user which identifies the To-Be-Optimized pages. You can specify whether a node is a power node before the graph is generated for Max Flow Input. These nodes can accept a capacity equal to infinity and thus have high incoming flow.
To-Be-Optimized Pages / Nodes: These nodes are the user parameter based on which the optimization occurs. When such nodes are encountered the user has selected to have the web site optimized for those pages. Once these pages are identified, the focus is towards optimizing their associated PageRanks. You can specify whether a node is a To-Be-Optimized node before the graph is generated for Max Flow Input. They are connected to the sink node in the graph structure. As the sink nodes have an infinite incoming flow, the To-Be-Optimized nodes can have a high outgoing flow.

As soon as the nodes are identified as a Power or To-Be-Optimized node, the graph is generated with the following steps:

1. Nodes are created, including the source and sink.
2. All the Power nodes are connected to the source node.
3. All the To-Be-Optimized nodes are connected to the sink node.
4. The link structure provided as an input by the user is then used to create and plot the edges that exist between different nodes. Thus completing the graph structure.

4.3 The NEW MMSA process:
Before looking at the details of the MMSA working
The project applied deals with collecting data like number of pages in the web site under consideration, the links that could form the matrix, the initial PageRank, the damping factor and the number of iterations. The sequence of this data collection is:

Get the number of web pages for the web site ➔

Figure 4.8: Step 1 of the MMSA integrated PageRank Application

Get the names of these web pages ➔

Assign the initial PageRank ➔

Chose which pages are power pages and which are to be optimized ➔
Figure 4.9: Step 2 of the MMSA integration
Define the link structure for the web page →

Figure 4.10: Step 3 of the MMSA Integration
The calculated Max Flow Algorithm will be returned. This also has the new link structure that the website should have to optimize the PageRank →
Figure 4.11: Step 4 of the MMSA Integration

These new values are then fed into the original PageRank algorithm to obtain the Optimized PageRank values and the link matrix.
Figure 4.12: Final Step showing graph generation from the MMSA Part of the PageRank Application

The above link structure obtained is then used as the input for the PageRank application. The output generated is shown below. This is the final optimized output.
Figure 4.13 Output generated using the MMSA PageRank application

The red boxes that are highlighted in the figure above are for comparison reasons with figure 3.4 from chapter 3. The range of output data obtained in iterations 1, 2, 3 or 4 in the above figure are obtained in the last four iterations in figure 3.4. Thus, when the MMSA PageRank application is used the output data is achieved in less number of iterations as compared to the original PageRank application. This is directly related to the complexity of the application as the less
number of iterations will require less time and produce optimized results. Thus, a 50% cut down in the iterations required to produce better results is one of our achievements.

The Theoretical aspect of the MMSA model with the PageRank algorithm uses some key concepts. The MMSA process is triggered using an event, which then checks for a certain criteria and finally works for the different parameters. These parameters are the data that is required for graph generation. When relating the theoretical and practical approaches, the events of the MMSA process can be the criteria is the type of pages that need to be optimized and the parameters are the initial PageRank assigned to the application when calculating the optimized PageRank.

4.4 Application - Explained:

The application built on the basis of the principles proposed by the MMSA model is the modified PageRank application and can be accessed [http://prathyushinangia.com/project/home.html](http://prathyushinangia.com/project/home.html). The application is a web based application and uses the combination of languages – HTML and JavaScript. It also uses CSS. The basic functionality uses the JavaScript at the backend to achieve the Max Flow implementation.

The objective of this project implementation is testing the real world approach of the MMSA theoretical model. Thus, for testing purposes the original and modified versions of the PageRank algorithm have been placed on the same page, dividing the page into horizontal sections for ease of view. This horizontal division leads to better comparison view. The placement is such that we have the modified PageRank application in the upper half of the page and the original PageRank application in the lower half.
The modified PageRank application starts with the data collection process which is required as the data input to the Max Flow graph. The event is triggered when the page number entered is greater than 1. The application checks if this event occurred. If yes the application runs and collects the names of the web pages and the initial PageRank assigned to each specific web page. In addition, the application also marks the pages that are power pages and recognizes the pages that are required to be optimized. These power pages are defined as the pages that have the maximum PageRank. As these pages have a high PageRank these pages can help distribute that PageRank that they have. These power pages are connected to the source, s. On the other hand, the pages that are to be optimized in terms of their PageRank values are recognized and connected to the sink, t. thus, after plotting the original graph the source and sink are connected to the respective power and to be optimized pages. The next step of the application is to provide the interlinking matrix that defines the graph’s edges. These edges are the interconnecting nodes that were selected by the user as the linking structure of the web site. Thus, the graph input for the Max Flow Algorithm is generated. Thus, the MMSA graph generation process is obtained. An example can be explained using the figure:

Consider a user input of the five web pages which together make a web site. This web site is for a company that offers services. The link structure that exists between them is given in figure 4.14.
The PageRank associated with each page is:

PR (Home) = 8
PR (Blog) = 4
PR (Services) = 3
PR (News) = 2
PR (Shop) = 5

Create two nodes, source and sink. The source node has an out flow of infinity and the sink node has an in flow of infinity. Furthermore, the pages – services and shop should have a high PageRank as the company is a service provider. These pages are termed as to-be-optimized pages. So these pages should be connected to the sink node. Also the home and news page are power pages, since they provide most of the information and keyword search could be facilitated from these pages. These are connected to the source node. The directed graph in figure 4.13 shows the connected nodes and the PageRank distribution as discussed in chapter 5.
Thus, the given graphs are then connected to the source and sink after the power pages and to be optimized pages are recognized. This is an essential step to obtain optimized results.

This graph obtained is then validated and the Max Flow Algorithm is used to optimize the graph input. The Max Flow Algorithm uses the graph and the weights and capacities which in the practical terms are the PageRank and infinity. The Max Flow algorithm uses the optimization algorithm to provide with results that can be further used for the PageRank application. Figure 4.14 is the optimized graph obtained using the Max Flow Algorithm. This directed graph shows the new link structure that should be incorporated in to the web site.
This new link matrix provides the new linking structure which has to be used as the input for the PageRank application. For calculating the PageRank we feed this new link matrix into the Mark Horrell’s PageRank application. This application can be accessed here: http://www.markhorrell.com/seo/pagerank.asp.

Convergence and Power Method:

The term convergence walks hand in hand with the PageRank Algorithm. Convergence can be defined as the process of obtaining a stable value over a finite period of time. As the PageRank algorithm is an iterative process, the PageRank vector stabilizes over a number of iterations. The lower the initial PageRank is the faster the Vector stabilizes. This is important as the less number of computations can reduce the computational cost of the algorithm. The web is a huge structure. Thus, computing the PageRank for all its pages involved high amounts in time and expense. This is a concern to the relevance of the web structure that we use, as Google crawls the web once a
month. During this period between two crawls, the web structure can witness drastic changes. This property interests many researchers.

To obtain convergence we use what is called the “Power Method”. The Power Method is the most popular calculation method used for eigenvector calculations. This is because of computation time, complexity and the issues posed by the use of other calculation methods. To find the vector $x^{k+1}$ from $x^k$ as $x^{k+1} = Ax^k$ until $x^{k+1}$ converges within a desired tolerance. When $x^{k+1}$ converges, the vector obtained is the eigenvector for the dominant vector and the given matrix. The PageRank is an application of the Power Method. The PageRank can be given as:

Line 1 initialize ranks $R_0$
Line 2 while (not converged)
Line 3 for each vertex i
Line 4 $R_{k+1}(i) = (1-d) + d \sum_{j \in B_i} \frac{R_k(j)}{N_j}$
Line 5 end
Line 6 end

The Power Method or the Power Iteration is a simple algorithm and computes only a single value. This value is an absolute value obtained over a number of iterations. This absolute value is the dominant eigenvalue. Thus, the PageRank algorithm converges after a number of iterations and hence is called a stable algorithm. The complexity for this convergence is $O(\log(n))$ iterations on expander graphs. Expander graphs are special graphs that have strong connectivity properties persisting in originally sparse graphs. We will include the convergence properties of the PageRank application for analyzing the improvements obtained while using the Multidimensional Max Flow Software Agent PageRank application.
5.1 Feasibility of the Model Proposed:

The Abstract model of the MMSA process was proposed as a thesis work by Priyanka Bihani, student of the Computer Science Department at California State University. She presented an abstract thesis model of the Multidimensional Max Flow Software Agent Model (MMSA). My project dealt with implementation of the MMSA model. The implementation had various cut offs, implementation scenarios, achievements, improvements and mitigations. The achievements, improvements and mitigations were actual specific cases where some ideas of the MMSA were applied successfully and others were improved and worked upon or deprecated. During the entire Software Development Life Cycle, the project faced delays and non-feasible approaches towards the practical implementation.

The entire project was simple when it came to methodologies like software agents, their selection, Max Flow Algorithm and obviously the PageRank Algorithm. However, when these words came together to form a project, it summed up to be Multidimensional Max Flow Algorithm Software Agents model. This MMSA model was a lot of study and research of in depth topics related to the core. The MMSA model has characteristics such as flexibility, expandability, dimensionality and robustness. The feasibility issues that were a challenge when it came to the MMSA implementation are discussed:

- Application variation –

  The Google PageRank application has variations. The links below are of the variations that can be found on the web.

  http://www.page-rank-calculator.com/
This application uses the link of a web page as an input and returns the calculated PageRank. This is a Google application. This application requires analyzing the link structure of the given web page link and then creating a graph out of the links found when traversing the web page requested.

http://www.webworkshop.net/pagerank_calculator.php

The web work shop application is a calculator similar to the one used in this project implementation. However, it takes the out bound links into consideration. The out bound links are also critical when it comes to calculating the PageRank for a web page. This application is selected for the implementation of the MMSA model.

http://www.prchecker.info/

The PR checker is an application that can be included in the web page and it will be a track of all the changes that happen in your web page structure and keep calculating the PageRank. Also it will suggest the link structure that should be maintained in the web site.

- Max Flow Algorithm variations –

  The Max Flow Algorithm is an optimization algorithm and is used in network and routing problems. There are variations of the Max Flow algorithm based on the time complexity calculated for these different approaches. The two most appropriate variations are the Ford Fulkerson and the Edmonds Karp Algorithm. The first is exponential in nature and the latter is linear. Thus, Edmonds Karp algorithm was selected for the implementation purpose. Also, the Max Flow graph layout does not allow cycles or reciprocal links. This is an added check included in the MMSA application to avoid graph layouts having such
structures. Thus, the randomly generated graphs using the criteria and parameters strictly follow the rule and eliminate any such layout showing cycles and reciprocals.

- Events, criteria and parameters –

The determination of the events, criteria and the parameters mentioned in the theoretical MMSA model proposal had to be relevant in the practical implementation of the MMSA model. The data collection process was initiated when an event is triggered. The MMSA model works on an event. This event is essential to start the process flow. Correlating to the criteria and parameters between the theoretical and practical approaches is a critical decision. This event mentioned has been taken as an event that the user visits the website. Following the event is the criteria selection. The criteria, in the practical approach are the user data entered using collaborative intelligent software agents. At last the parameters are entered. This is done by identifying the pages as Power pages or To-Be optimized pages.

- Desktop and Web application -

The MMSA model does not imply any restrictions on the platforms or the languages that can be used in developing the application. We have researched a lot about the web based application. However, when looking for a desktop application for PageRank calculation none were found. As the PageRank is a web based term, web based PageRank calculators are used more. Plus they are readily available and are free to use. Also the variations of the application make it much easier to access on the web.

- Distributed and Parallel processing –
For small data sets it doesn't really affect the kind of processing applied. Time efficiency is not a matter of concern for small data sets. But when the data sets are large, time efficiency should be considered. Time efficiencies improve if applications using large scale data sets process the independent layers or levels of the model in parallel. There are sequential dependencies in certain parts of the MMSA model process. However, there are some parts of the model where there are no sequential dependencies and the processing can be done in parallel. There are various dimensions or levels generated for multiple graphs in the model. The graph layouts are generated depending on the parameters selected. For each applicable parameter, the graph generation can be processed independently, as it does not impact other applicable parameters. This can be seen in figure 1.3. This is also illustrated as an example in figure 3.1 with the vertical regions marked giving instances of parallel processing. There can be more parts processed in parallel for the MMSA model. This makes it suitable for parallel or distributed processing for applications involving large data sets.

- **Dangling links** –

Dangling links are defined as links that have a huge amount of PageRank inflow but does not have a PageRank outflow. Such links are dead links and are no good to the link structure. However there can prove disastrous as they can absorb all the PageRank flow in the network. As suggested by Brin and Page, to calculate the PageRank in the present scenario, the dangling links are taken out of the web structure and then the PageRank is calculated. After the calculations have been completed the dangling links are brought back into the network and added. This is a chaotic situation because of the huge number of dangling links in the web structure. Also these links are first searched for leading to a
costly search before implementing the PageRank algorithm. We have developed a model tool that helps remove dangling links. This is achieved with the help of the Max Flow Algorithm. Here the Max Flow plays a vital role. The Max Flow algorithm takes as input the link structure provided by the user. This is then used to calculate other random graphs. Once these graphs are generated the Max Flow for each graph is calculated. Thus, the optimum is selected and the others are discarded. This optimum graph depicting the best linked structure has a Max Flow in the network. We will consider three examples that have dangling links in different positions in a graph – At the Start, Near the Start, Middle, near the end and at the end.

<table>
<thead>
<tr>
<th>Number of Nodes</th>
<th>Dangling Link Position</th>
<th>Number of In-Flow links</th>
<th>Max Flow execution time in ms</th>
<th>Contains dangling links in the Final Structure</th>
<th>Selected a value lower than the Max Flow Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Start</td>
<td>2</td>
<td>31</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>Start</td>
<td>3</td>
<td>52</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>Start</td>
<td>4</td>
<td>108</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>15</td>
<td>Start</td>
<td>2</td>
<td>91</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Middle</td>
<td>1</td>
<td>32</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Thus, the Max Flow of the network when the network has a dangling link at the start or the middle of the network is 0 as shown above. Having a dangling link at the end has different outcomes.

From table 1 above, it is clear that the Max Flow algorithm generates other multiple optimum solutions when it takes as input a network diagram with dangling links. If this could be escalated to a bigger picture having web sites in places of web pages we could be helping to decrease the additional cost incurred to search for the dangling links in the web structure before the implementation of the PageRank.
• Iterations –

The PageRank Algorithm is an iterative process. This implies that the more the number of iterations the more accurate the calculated PageRank for a web page is. The number of iterations can be user defined. These iterations chosen have an impact on the complexity of the PageRank Algorithm. Each iteration contributes towards a better PageRank calculation and helps achieve convergence. The MMSA PageRank implementation helps us to achieve this goal in less number of iterations. Thus, as seen in the comparison of the data outputs generated in figure 3.4 and figure 4.13, we can imagine the amount of decrease in the complexity when the number of iteration decreases from 20 to 5.

For a clear comparison and study of the actual improvement, data was collected for different range of web pages and iterations for both the applications. Random numbers were used as PageRank values. A collection of such data ranges are summarized in the table below:
The table 2 above shows some statistics to determine the use of the tool developed. Two critical problems faced by Google are the high number of iterations required to stabilize a PageRank and dangling links for computation. The number of iterations required by the MMSA application

<table>
<thead>
<tr>
<th>Number of web pages</th>
<th>PageRank Iteration</th>
<th>PageRank Value at Iteration</th>
<th>MMSA Iteration</th>
<th>MMSA Value at Iteration</th>
<th>Stable?</th>
<th>Improvement in PageRank</th>
<th>Time Required in ms for Max Flow</th>
<th>User Link structure = MMSA Max Flow Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>20</td>
<td>1.124</td>
<td>21</td>
<td>3.9</td>
<td>Yes</td>
<td>Yes</td>
<td>28</td>
<td>No</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>3.3</td>
<td>29</td>
<td>6.1</td>
<td>Yes</td>
<td>Yes</td>
<td>51</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>2.7</td>
<td>10</td>
<td>5.0</td>
<td>Yes</td>
<td>Yes</td>
<td>44</td>
<td>No</td>
</tr>
<tr>
<td>12</td>
<td>16</td>
<td>5.3</td>
<td>16</td>
<td>5.3</td>
<td>Yes</td>
<td>No</td>
<td>79</td>
<td>Yes</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>5.0</td>
<td>12</td>
<td>7.0</td>
<td>Yes</td>
<td>Yes</td>
<td>44</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.75</td>
<td>2</td>
<td>0.75</td>
<td>Yes</td>
<td>No</td>
<td>22</td>
<td>No</td>
</tr>
<tr>
<td>15</td>
<td>40</td>
<td>5.46</td>
<td>17</td>
<td>14.11</td>
<td>Yes</td>
<td>Yes</td>
<td>134</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 2: Statistics for convergence obtained using the Iterative approach
assures a better PageRank than that obtained using the original PageRank Application. Further analysis can be done on the factor that stabilizes the PageRank vector.

- **Convergence** –

  In 1998, Brin and Page claimed that a web of approximately 322 million pages converged in about 52 iterations and a web half that size converged in about 45 iterations. These are large numbers. We are however dealing with smaller graphs. Thus, the improvement rate of convergence from the PageRank application to the MMSA PageRank application is minute and sometimes even un-noticeable. We should however, keep in mind of the escalation requirements of this tool that is currently built for small scale structures. Let us look at some data analysis done to achieve the statistics for Convergence trends in the MMSA Algorithm.

  In table 2 above the column header that shows the statistics for obtaining a stable value is called convergence. In most of the cases we have obtained a converging value. This value is either a stable number or a number that changes the second digit after the decimal. In any case, it stabilizes to a whole number being the same.

- **Accessibility** –

  Accessibility is one factor that is now vital for educational web sites. However, we should also start implementing accessibility in our non-academic web sites. Making accessible web sites will help create a common platform. Moreover, maybe Google’s Search Engine starts counting accessibility in web pages as one of the criteria for ranking web pages. You never know – what Google does next.
We have designed our web site to meet accessibility standards such as ADA and 508 compliance. Also the use of the WAVE toolbar helps to recognize the issues that the can help serve people with disabilities. This helps to assist the screen readers and other assistive technologies read the screen text right. The assistive technology has been improving and so is the web technology. Thus, both should support each other.

The WAVE toolbar gives us no error on our web page. This is included in the figure 5.1 that shows a screenshot of the results obtained. Initially some errors were detected on the web page. However, by using the WAVE toolbar and FAE, I worked on the errors, so the application is Accessible. You can download the WAVE Toolbar here: http://wave.webaim.org/toolbar. Also you can use this link to WAVE your page without downloading this toolbar.

![Figure 5.1: The WAVE Toolbar showing the results of the MMSA PageRank Application](image)
Figure 5.2: The WAVE Toolbar showing the results of the Mark Horrell’s PageRank Application
FUTURISTIC APPROACH

The MMSA model is a robust newly developed algorithm that has been implemented on the existing PageRank algorithm to test the objectives claimed by the abstract process. An in depth study of what kinds of agents are available and the services they offer should be the direction to start with. The knowledge about the agent types could help in selecting or deciding about the agent integration in the MMSA model.

Also, a new version of the Max Flow algorithm could be proposed which has a better complexity than Edmonds-Karp algorithm. When you take the PageRank application in consideration, the user interface could be a challenging aspect as the number of links has to be entered manually. That means the user has to keep checking boxes off whenever there exists a solution. This could be a tedious task when handling large data sets.

There could also be variations in the implementing methods for the MMSA model. The new variation suggestions should work both horizontally and vertically. It could also be implemented as an iterative process for a specific portion of the application or could be many MMSA’s integrated at different stages and levels. Giving a clear picture would be an implementation where MMSA occurs at one level, produces a solution set and this solution set is then passed to the next level. It will create a hierarchy suitable for application in which it needs connected levels to transmit values on to the next level.
CONCLUSION

The MSSA model represents a fundamentally new model that supports both, robustness and dimensionality. It is the next step forward in the world of automation. The MMSA model has integrated in itself technologies like software agents which provide the automation, Max flow algorithm which provides the dimensionality. The purpose of this whole new level of integration seems fulfilled as and when the solutions returned are optimized. For small data sets, a second saved in computation may have no affect but when considering large amounts of data sets every second counts. The research required for the implementation let alone the research for the idea was not a stroke of luck. Every phase had to be planned and then implemented. During the implementation of this project, I have faced hurdles and options that sometimes even took me to the wrong direction and delayed the process of development. However, the alternatives, all of them have helped me learn more about why and why not it can be implemented. Thus, nothing at all during my project development phase has gone to waste. The MMSA model was developed keeping in mind the agile software development life cycle. It has witnessed changing requirements, changing implementations, changing logics and even changing interfaces. Thus, I could put into practice what I have learned as the ABC of software development.

There are a lot of important aspects for the successful implementation of the MMSA model. These have to be in synchronization with each other, as majority of them are dependent. The abstract description of the MMSA model had to be tweaked a little as and when required to achieve the realistic goals. The algorithm and the flowchart are also compared with the real world application that was selected for the basis of the implementation and comparison aspects. The PageRank application had variations in its implementation procedures. This was also dealt with by chalking down the advantages and disadvantages for each variation. Then the
modification of the original application into the MMSA was studied and the most favorable application variation was selected. This was further analyzed to meet the specific requirements to meet our goals. The order and flow of events, the criteria, parameters, graph layouts and their respective validations and optimizations are all vital components to meet the MMSA model implementation. The negligence on one or more elements or even a combination could require reconsideration. All the above elements are the basis the MMSA achieves the desired optimization. Of all the above, the criteria, parameters and events are application dependent. An event is a real world phenomenon that starts or triggers the MMSA process. The criterion specifies the objective for the model to satisfy. The negotiation process selects the relevant parameters based on the criterion to generate graph layouts. The set-up is of a directed graph having vertices, edges with weights, and modeling movement of flow. The dynamic vertices are vertices in a directed graph that are modeled as software agents to offer a powerful functionality. This functionality resulting from the software agents pertains to behavior cooperation and mainly negotiation done by interaction between agents in a coordinated way to accomplish objectives. The negotiation process is a vital agent function carried on while interacting with vertices for generating graph layouts. By having one or more applicable parameters it generates one or more graphs for each parameter. This graph generation is carried out in a parallel approach. Using the parallel method for processing the graphs improves the execution time. In the implementation 5 graphs are generated - 4 random graphs and 1 user entered. However, this can be scaled to a larger number. The difference in the execution time will not be clear in small scale systems. However, monitor the execution time for large scale systems and you will see the importance of using parallel approach here. This adds to the dimensionality of the model, giving the multidimensional effect. The valid graphs generated are optimized using the Max Flow algorithm. Thus, an optimized
PageRank is obtained after implementing the MMSA process. Also when the PageRank’s convergence factor is studied, the MMSA has achieved an improvement to obtain the convergence factor to be smaller than the factor obtained by the Original PageRank application. The execution time for the Max Flow algorithm calculation and the graph generation has been displayed with the help of timer. This execution time is in milliseconds and is directly proportional to the number of web pages in a website. Reducing the number of iterations can help to decrease the computation time for each individual web page. This has also been achieved by the MMSA implementation. The Multidimensional Max Flow Software Agent Model has been implemented on a small scale, however it has the ability to be expanded to a larger scale and work integrated with Google’s PageRank application. Talking about the other areas of improvement, MMSA has removed the dangling links in the web sites. This improvement will have a huge impact on the execution time required by Google every month to calculate the PageRank’s for individual pages. Before the PageRank calculation starts the crawler searches for all the dangling links and removes from the calculation matrix. These removed dangling links are stored elsewhere and then returned back to the structure. Hence, a lot of time and memory could be saved.
REFERENCES


