There are millions of HTML pages in the Net and most of them contain links. These links have a direction. Some pages receive much attention and have many pages referring to them: software repositories, international organizations, pages written by acknowledged experts in various fields, etc. Also there are pages with many links, such as resource lists. This is a circular definition but it works. Iterative algorithms can take usual text-search results, sort out things and produce the "authorities". This is the idea behind the IBM CLEVER project, click here for details. Google also does something similar, read the press release. Interestingly, these ideas can be extended to find concepts and applications of Graph Theory in Web mining. A directed graph is a collection of nodes (vertices) together with arcs joining some of these vertices. The Web is a huge directed graph \( G = (V,E) \), where the set \( V \) of vertices is the set of all pages and the set \( E \) of arcs corresponds to all pointers.

The Web is certainly not a complete graph. In fact, it is a very sparse graph. The pointers are much less than the maximum possible. The Web is not a connected graph. Although every page is accessible to anybody by "definition of the Web", this does not mean that everything is reachable from everywhere just by following links. There are isolated groups of pages with no link from/to their outside world. In fact, there are also isolated vertices, not pointing to or being pointed by anything, for example, personal pages with just a CV, without pointers, accessible only from their known URL.

From now on, let's stay within a subgraph \( G' \) of the Web, that is, some subset of the Web pages together with the arcs that have end points in this subset. This subset could be a "community", say all pages on a particular topic, all commercial pages in some domain etc. There are complete subgraphs in the Web.

A Minimal Dominating Set \( G' \) is a minimal subset of vertices from which every other vertex in \( G' \) can be reached immediately. This would be an economic collection of bookmarks. Each page would be in the bookmarks or would be pointed by a bookmarked page. The number of vertices in the minimal dominating set is called the dominance number of \( G' \).
The Set Covering Problem (SCP) is exactly this problem. If in addition, we don’t want overlaps in the coverage, we are facing a version called the Set Partitioning Problem (SPP). SPP would obviously lead to a larger bookmarks list because it has more constraints. More precisely, some of the SPP constraints are relaxed in SCP, so the latter gives better results.

What is the Shortest Path between two pages A and B in the same community which are totally unknown to each other? In a specialized community with “extravert” members, it is highly probable that there are various sequences of pages, each pointing to the next one, starting at A and ending at B. If we take the “length” of arcs to be 1, the Shortest Path would be the sequence with the minimum number of steps. In some cases this might be surprisingly short. Arcs can also have varying lengths, say the “distance” between the hosting servers (geographical, average connection time etc). In that case, the Shortest Path solution would approximate the fastest way to locate page B starting from page A.

The Shortest Spanning Tree: From all arcs in G’ select some so as to cover all vertices tree-like, at minimum total “length”. There is now a unique way from every page to another and there is also a root. The root is not necessarily a page of particular importance; it is just the best location to start building the optimal tree. It is not necessarily a resource list. Notice that you cannot do cycles in this collection of arcs, i.e. you cannot go back to the starting page.

Another way to build a bookmarks list is to take a time of fixed length, say p. Every non-bookmarked page has some “distance” from the list, the one from the nearest bookmarked one. If we want again to cover all the community in a way that minimizes the sum of all these “distances”, we are solving a minimum problem, the p-Median Problem. This problem is better understood as an approach to install facilities in an optimal way for customers. A practical application would be the location of mirror sites.

The minimax version is the p-Centres Problem where we seek to minimize the maximum “distance”. Again it is closely related to facilities location problems, but the emphasis is more in avoiding extreme delays in serving individual customers rather than the community as a whole.
Graphs can have flows. An obvious analogue is network traffic between servers/vertices with the connecting lines having fixed capacities. The Maximum Flow Problem would find the maximum possible traffic between any two servers. There are also special algorithms to find the maximum flow between every pair of servers/vertices. Also, if the traffic between two servers is set, there are various flow patterns to achieve it. Given also costs per unit flow on the lines, we can find the Minimum Cost Flow between the two servers. These are techniques actually used in the design of networks.

The Graph Partitioning Problem would divide pages in a community into clusters of minimum interaction. There are many variations of this problem, depending on the constraints we impose: maximum cluster size, fixed number of clusters, fixed number and sizes of clusters, equal-size clusters (equipartitions) etc. The simplest case is partition into two clusters. If the two clusters are not necessarily of equal size, this would clearly divide certain communities into “opposite opinion” groups.

For the commercial sites in a specific industry in some country, meaningful clusters would be numerous and would correspond to groups of affiliated or otherwise related companies - nobody points to competitors!

Maintenance recommendation has been an important topic in e-commerce. Subjective classification is a potentially useful approach for both better understanding customer Graph based Web logs and identifying information actionable to customer maintenance. Subjective classification seems attractive because obtaining a large set of objective data, with labeling for training and testing, is often difficult. In particular, building a classifier when a training data set is small and possibly inaccurate is important. That's because decision makers find that identifying user purchase patterns from a Graph based Web log is difficult. There's no direct relationship between Graph based Web log data and purchase patterns. It's also difficult because the information in the small training data set is insufficient. A proposed method to build a classifier further selects a small subset of the training data set to build a classifier that possibly leads to high accuracy. This approach can help identify whether customers have purchase interest. The result of such classification provides actionable patterns and helps companies gain high customer maintenance.
Thus, organizations must understand their customer's behaviour, preferences and future needs. This imperative leads many companies to develop a great many e-service systems for data collection and analysis. Web mining is a popular technique for analyzing visitor activities in e-service systems. It mainly includes:

- Web text mining—extracting knowledge from the content of documents or their descriptions
- Web structure mining—inferring knowledge from Internet links and organization
- Graph based Web log mining—extracting interesting patterns in Web access logs and other Web usage information.

The ultimate goal of Web mining is to identify actionable information that can help, for example, acquire new customers, retain old customers, and grow customer's profitability. However, there's no direct relationship between Graph based Web log data and the visitor's purchase patterns. When the training data set is small and possibly inaccurate, the task is even more difficult because a small set of training data is insufficient. We've developed a subjective classification approach that provides companies with actionable patterns, thereby gaining high customer retention. A typical Graph based Web log contains information including requesting time, IP addresses, access methods, the URLs of requested files, HTTP protocol version numbers, the status of server responses, numbers of transfer bytes, URLs of referrer's pages, and user agents. Our Graph based Web log mining approach classifies a particular site's visitors into different groups on the basis of their purchase interest.

The best potential customers are ready to click and buy. Some are prospecting for information and familiarizing themselves with different brands; they might become important, profitable customers in the future. Others enjoy free rides and never purchase anything. We can subjectively identify visitors who have some purchase interest and those who don't buy their access patterns. Visitors with purchase interest exhibit certain access patterns:
• They access certain pages for a rather long time because they need time to read the contents. So, the ratio between the amount of time they need to read contents and the amount of time they navigate from one page to another is large.

• They navigate down to low-level pages because they need to access specific topics.

• They often use the HTTP POST mode (which sends data to a Web server and retrieves a response), because they’re interested in registering with Web sites and are willing to fill out forms with their own information.

• They often access images and graphic files.

The aim of the present thesis is to investigate "Some-real time analysis of applications of graph theory concepts in web mining with respect to online trading." The thesis consists of seven chapters.

CHAPTER-1

GRAPH-BASED WEB MINING

Graph-based Web mining represents a collection of techniques for mining the relational aspects of Web represented as a graph. Two major approaches to graph-based Web mining are frequent subgraph mining and graph-based relational learning. This chapter focuses on one particular approach embodied in the Subdue system, along with recent advances in graph-based supervised learning, graph-based hierarchical conceptual clustering, and graph-grammar induction.

Logic-based approaches fall under the area of inductive logic programming (ILP). ILP embodies a number of techniques for inducing a logical theory to describe the Web, and many techniques have been adapted to multi-relational Web mining. Graph-based approaches differ from logic-based approaches to relational mining in several ways, the most obvious of which is the underlying representation. Furthermore, a logic-based approach rely on the prior identification of the predicate or predicates to be mined, while graph-based approaches are more Web-driven, identifying any portion of the graph that has high support. However, logic-based
approaches allow the expression of more complicated patterns involving, for example, recursion, variables, and constraints among variables. These representational limitations of graphs can be overcome, but at a computational cost.

CHAPTER-2

CONSISTENT BIPARTITE GRAPH CO-PARTITIONING IN WEB MINING

Collection of Graphs attracted more and more attention in recent years due to its high impact on various applications in web mining. While the Graph algorithms are two types of data, such as documents and terms, have been well studied in the literature, the work on more types of data is still very limited. As an attempt in this direction, in this chapter, we worked on a specific case of high-order Graph in which there is a central type of objects that connects the other types so as to form a star structure of the inter-relationships. Actually, this case could be a very good abstract for many real-world applications, such as the Graph of categories, documents and terms in text mining. In our philosophy, we treated such kind of problems as the fusion of multiple pair-wise Graphs sub-problems with the constraint of the star structure. Accordingly, we proposed the concept of consistent bipartite graph co-partitioning and developed an algorithm based on semi-definite programming (SDP) for efficient computation of the Graph Clustering results.

CHAPTER-3

INFLUENCE OF GRAPH THEORY IN WEB MINING

From its very beginning, the potential of extracting valuable knowledge from the Web has been quite evident. Influence of Graph theory in Web mining - i.e. the application of data mining techniques to extract knowledge from Web content, structure, and usage - is the collection of technologies to fulfill this potential. Interest in Influence of Graph theory in Web mining has grown rapidly in its short existence, both in the research and practitioner communities. This chapter provides a brief overview of the accomplishments of the field - both in terms of technologies and applications - and outlines key future research directions.
CHAPTER-4

ANALYSIS OF THE PERFORMANCE OF MINING WITH ANT COLONY OPTIMIZATION

In this chapter, a review of existing ACO algorithms is offered and an identification of common features used in the development of a Mining ACO framework. An empirical analysis of these novel implementations is also presented using a variety of single and multiple objective continuous function and combinatorial optimization problems based visitor entries. These optimization problems have been chosen since they demonstrate the advantages and disadvantages of adding MACO algorithm. Here, these two new MACO algorithms are applied to a real-world optimization problem.

CHAPTER-5

SYSTEM STUDY

In this chapter, we have been discussed the goal of the feasibility studies to evaluate alternative systems and to propose the most feasible and desirable systems for the development. The detailed study is carried out to check the work ability of proposed system. Feasibility study is a test of system proposal regarding to its ability, impact on the organization ability to meet user needs, an effective use of resources. Thus when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The goal of the feasibility studies is to evaluate alternative systems and to propose the most feasible and desirable systems for development. Feasibility studies typically involve cost/benefit analysis. The feasibility of a proposed system can be evaluated in terms of three major categories.

- Organizational Feasibility
- Technical Feasibility
- Economic Feasibility
CHAPTER-6

SYSTEM CONFIGURATION

In this chapter, we have discussed the requirements of hardware, software and the details of the software about the proposed system.

CHAPTER-7

DESIGN AND DEVELOPMENT

This chapter deals with the design and development of the proposed system. It is observed that the developed system “Graph based Web log classification” overcomes from all the problems that existing system one has. The system effectively maintains the register records in the database. As the system developed based on a framework, it is easy to enhance the system in future. The framework separates the business logic from the presentation logic. So, that one can easily modify or update the presentation without known of business tier and vice-versa. The GUI of the system gives the user friendliness to operate the system. No need of training to operate the system. Reports and searching for records is error prone, tedious as well as time consuming environment. The system makes report generation process from tedious to easy one. Unauthorized persons could not able to access the system. This makes the system security one. It becomes quite unreliable on keeping accumulated large number of records in the manual system. But the developed system overcomes from this problem. The data in the database is maintained with integrity and security.