Estimating Grid Reliability Using Bayesian Networks

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Ipneet Kaur
(800931009)

Under the supervision of:
Dr. Inderveer Chana
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2011

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*Estimating Grid Reliability using Bayesian Networks*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr.Inderveer Chana* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Ipneet Kaur)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

17/06/11

(Dr. Inderveer Chana)
Assistant Professor,
CSED,
Thapar University,
Patiala

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. My greatest thanks are to my parents who bestowed ability and strength in me to complete this work. I am deeply indebted to my parents and friends for their inspiration and ever encouraging moral support, which enabled me to pursue my studies.

This work would not have been possible without the encouragement and able guidance of my supervisor Dr Inderveer Chana. Her enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. Her feedback and editorial comments were also valuable for writing this thesis.

I would like to express my sincere gratitude towards all the members of the Centre of Excellence in Grid Computing at Computer Science and Engineering Department, Thapar University for their unending support.

I am also very thankful to the entire faculty and staff members of Computer Science and Engineering Department, especially Mrs. Anju Bala, S. Gurpal Singh and Ms. Rajni for their direct–indirect help and cooperation. Also I would like to thank my friends especially Sonam Chawla, Aradhana Majithia, Vaneet Kaur, Arpita Sharma, Aarti Sharma, Ravneet Kaur Grewal, Amandeep Kaur Johar and Ravneet Kaur Chawla for their support.

Ipneet

(800931009)

Abstract

Grid computing has emerged as an important field for complex systems with need of sharing of resources in a large scale environment, wide area communication and multi institutional collaboration. Grid computing is used where there is a need of large amount of computational power which cannot be provided by a single computer.

Both grid and distributed environments are different from each other in context of large scale resource sharing in grid. With the increase in the size of these systems the need to evaluate their reliability is increasing day by day. Hence various methods for estimating reliability are being developed. Bayesian Networks (BN) can also be used to estimate grid reliability.

BN is used to depict the interactions between the various components of the distributed system and show the relationship between them. The BN depicts the probabilistic associations between the system components in a very simple and easy way. The relationship is represented using directed acyclic graph where nodes represent the variables and the links between each pair of nodes represent the causal relationships between the variables. So BN is the best method that can be used to evaluate the reliability of the grid services.

This thesis discusses the use of Minimal Resource Spanning Trees (MRSTs) and BN in estimating grid service reliability. Also this thesis depicts that the Bayesian Networks can help in easy representation of grid networks, hence only historical data is required to construct them. In this thesis, the K2 algorithm has been extended to estimate the reliability of the MRST and used to build BN. Then this extended algorithm has been implemented using Java and validated.

CONTENTS

List Of Figures

List Of Tables

Chapter 1
Introduction

This chapter gives a brief introduction about grid computing, reliability in grid computing and Bayesian Networks (BN). It also presents the estimation of grid reliability using BN along with the organization of the thesis.

1.1. Grid Computing

In today's world of high speed and powerful computing, even home based desktops are powerful enough to run complex applications. But still there are numerous applications and complex scientific experiments which require huge amount of computational power and other resources. Such requirements are fulfilled by grid computing[29].

The basic idea of grid computing was conceived by Ian Foster, Carl Kesselman and Steve Tuecke. They worked to develop a toolkit which could manage computations, data movement, storage and other infrastructure to handle large grids without any restrictions regarding specific hardware and requirements. The conventional networks focus on communication among devices. Contradictory to these, grid computing (a form of networking) utilizes the unused processing cycles of all computers in a network for solving problems that are not possible for any stand-alone machine[30].

Thus, grid computing can be defined as a form of networking unlike conventional network that focus on communication among devices. It harnesses unused processing cycles of all computers in a network for solving problems too intensive for any stand alone machine[4].

The real and specific problem that underlies the Grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations. The sharing that we are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. A set of individuals and/or institutions defined by such sharing rules form what we call a Virtual Organization (VO)[2].

Grid computing appears to be a promising trend for three reasons[6]:

- its ability to make more cost-effective use of a given amount of computer resources
- as a way to solve problems that can't be approached without an enormous amount of computing power
- because it suggests that the resources of many computers can be cooperatively and perhaps synergistically harnessed and managed as a collaboration toward a common objective.

There are several reasons for programming applications on a Grid[1]:

- to exploit the inherent distributed nature of an application;
- to decrease the turnaround/response time of a huge application;
- to allow the execution of an application which is outside the capabilities of a single (sequential or parallel) architecture;
- to exploit the affinity between an application component and Grid resources with a specific functionality.

1.1.1. Comparison With Other Distributed Computing Technologies

Grid computing system is different from conventional distributed computing systems by its focus on large-scale resource sharing and open architecture for services[2].

➢ Comparison With Cluster Computing:

- Resources in a Cluster lie in the same administrative domain, where as in a Grid it’s hardly the case.
- Grids consist of heterogeneous resources, Clusters deal with homogenous resources that too only computational in nature.
- Grids are dynamic in nature, resources come and go in a Grid where as Clusters are static in nature.
- Grids are inherently distributed over a local, metropolitan, or wide-area network. Usually, clusters are physically contained in the same complex in a single location.
- Cluster interconnection technology delivers extremely low network latency, which causes problems if clusters are not close together and also makes Cluster scalability an issue. Grids are dynamic and hence more scalable.

- Comparison With CORBA:

CORBA shares surface level similarities with Grid Computing due to the strategic relationship between Grid Computing and Web Services in the Open Grid Services Architecture (OGSA). Both are based on the concept of Service-Oriented Architecture (SOA). A key distinction between CORBA and Grid Computing is that only CORBA assumes object orientation. In OGSA, there isn't a presumption of object-oriented implementation in the architecture. The architecture is message oriented; object orientation is an implementation concept.

- Comparison With P2P:

Grids usually have some form of centralized management and security which P2P systems lack, making them ideal for providing anonymity. Also, P2P systems are generally far more scalable than Grid Computing systems and P2P systems are generally more tolerant of single-point failures than Grids.

1.1.2. Classification of Grid

Grids can be classified on the basis of two parameters[2]:

- Grid Functionality

  On the basis of functionality, grid can be classified as computational grid and data grid
- Grid Topology

  On the basis of topology grid can be classified as IntraGrid, ExtraGrid and InterGrid

A Grid platform could be used for many different types of applications. Gridware applications are categorized into six main classes[1]:

- distributed supercomputing
- high-throughput
- on-demand
- data intensive
- collaborative
- service-oriented computing

1.1.3. Benefits Of Grid Computing:

Some benefits of grid computing are[3]:

- The first benefit is high performance and scalability that comes from the massive parallelism of many machines working together to finish compute workloads in parallel.
- The second benefit is ultimate reliability that comes from its managed clustering technology, in which a cluster of machines are connected and managed together to provide redundancy to each other. As long as there is one or more machines alive in the cluster, the grid should be able to continue serving for existing and new requests with its remaining capacity.
- Some other benefits like:
    - Exploitation of under-utilized resources
    - Reduces computational time
    - Provide information access
    - Reduces cost by optimising existing IT infrastructure
    - Providing access to parallel CPU capacity
    - Offers improved reliability
    - Provision of resource balancing
    - Effective management of resources
    - Access to additional resources

1.1.4. Grid Projects and Applications*:*

A well-known grid computing project is the SETI (Search for Extraterrestrial Intelligence) @Home project, in which PC users worldwide donate unused processor cycles to help the search for signs of extraterrestrial life by analyzing signals coming from outer space. The project relies on individual users to volunteer to allow the project to harness the unused processing power of the user's computer. This method saves the project both money and resources[4]. Fig 1.1 depicts the various initiatives taken by different countries in grid computing.

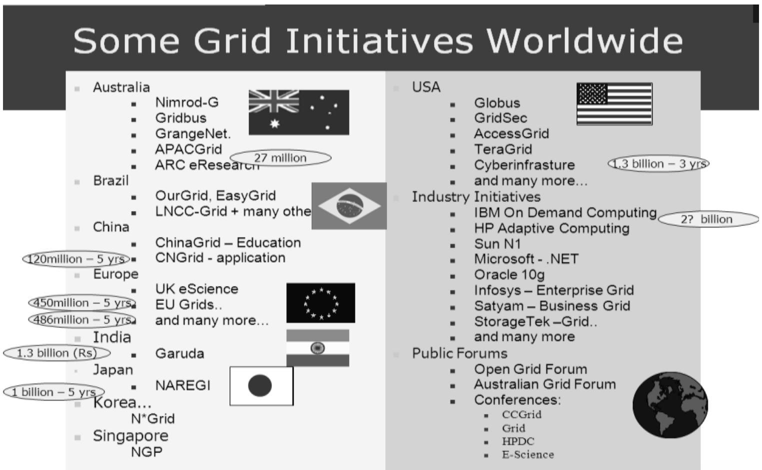

Fig 1.1: Some Grid Initiatives by different countries[4]

A typical example of a Grid Application is "weather prediction". This involves collaboration between several partners: TV stations that produce regular weather news reports, a Satellite Company that regularly provides space images of the earth, a supercomputing center that rapidly analyses the images and a visualization center that produces visual interpretations of the weather analysis. The smooth running of this project for the timely production of regular weather reports crucially depends on appropriate schemas for securely sharing, exchanging, and coordinating information between these partners.

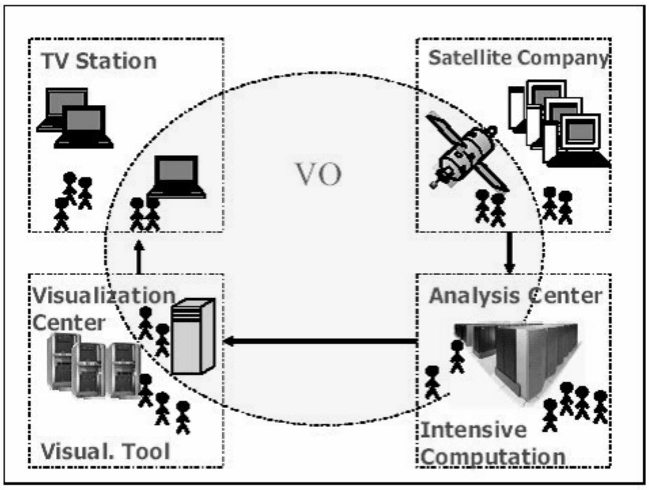

Fig 1.2: Weather Prediction Using Grid[4]

The power of Grid is particularly useful in areas involved in intensive processing such as life science research, financial modeling, industrial design and graphics rendering.

1.1.5. Issues in Grid Computing:

Grid Computing has following issues[7]:

- User Interface : Accessing information on the Grid is fairly important, and the user interface component handles this task for the user.
- Security: Computers on a Grid are networked and running applications; they can also be handling sensitive or extremely valuable data, so the security component of Grid Computing is of crucial concern. This component includes elements such as encryption, authentication, and authorization.
- Reliability: The ability to share resources among users is a fundamental concept for grid systems. Therefore, resource reliability is an important concern.
- Workload Management: Applications that a user wants to run on a Grid must be aware of the resources that are available, hence workload management service becomes an important consideration.
- Data Management: If an application is running on a system that doesn't hold the data the application needs, then data management facility takes care of moving that data to the right place across various machines, encountering various protocols.
- Scheduler: A scheduler is needed to locate the computers on which to run an application, and to assign the jobs required.

1.2.Reliability in Grid Computing

Grid Computing has gained enormous importance with rapid spread of computational environments dealing with large-scale applications in various areas. These applications involve a variety of geographically or logically distributed resources including supercomputers, storage systems and data sources owned by different organizations. In such circumstances, grid architecture offers communication among various users using an integrated system[9].

The ability to share resources among users is a fundamental concept for grid systems. Therefore, resource security, integrity and reliability are important concerns. Traditionally, the function of computer networks has been to exchange files between remote computers. In grid systems, this traditional requirement is taken a step further, so that networks provide different types of services such as computing, management, storage and so on. Thus, grid service reliability becomes an important issue for the system users due to these excessive system requirements[24].

The traditional reliability analysis perspective aims to provide accurate predictions about the system reliability using historical or test data. This approach is valid whenever the system success or failure behavior is well understood. However, for complex grid systems, understanding component interaction may prove to be a challenging problem, which usually requires intervention of a domain expert. In this respect, the Internet can be given as an example of a complex grid system. Due to the large number of internet users and the resources that are shared through it, interactions between the users and resources cannot be easily modeled. Hence estimating and evaluating reliability of grid service becomes a necessity[9].

Reliability can be defined as the measure of the likelihood of non occurrence of faults. The reliability of a system as a function of time R(t), can be defined as the conditional probability that the system has not failed in the interval [0,t], given that it was operational at time t=0. Therefore, it is essential to examine reliability to understand system fault tolerance[23].

Software reliability is a probabilistic measure and can be defined as the probability that software faults do not cause a failure during a specified exposure period in a specified use environment. The probabilistic nature of this measure is due to the uncertainty in the usage of the various software functions and the specified exposure period here may mean a single run, a number of runs, or time expressed in calendar or execution time units[5]. All the systems ought to fail, but the main point to be analyzed is when and how do they fail.

A number of analytical models have been proposed to address the problem of software reliability measurement. These approaches are based mainly on the failure history of software and can be classified according to the nature of the failure process like Times Between Failures Models, Failure Count Models, Fault Seeding Models, Input Domain Based Models, etc[5].

The most common software reliability model used is:

$R(t) = e^{-\lambda t}$

where λ is the failure rate.

Traditionally, the function of computer networks has been to exchange files between remote computers. In grid systems, this traditional requirement is taken a step further, so that networks provide different types of services such as computing, management, storage and so on. So, the major problem that underlies the Grid concept is synchronized sharing of resources and solving problems in ever changing, multi-institutional virtual organizations. The sharing that the grid computing is concerned with is not only exchanging files but in fact accessing computers, software, data, and other resources directly. Lots of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering require use of such sharing. This sharing is highly controlled, such that here resource providers and consumers define what is to be shared, who is allowed to share, and the conditions under which the sharing occurs. So this sharing needs to be highly reliable and in a synchronized and controlled way[23].

In an early stage, the grid reliability is mainly determined by the reliability of the Resource Management System (RMS), while in a later stage, the grid reliability is mostly affected by the reliability of the network for communicating or processing.

In RMS, the reliability is required because failures may occur in any of the layers in RMS like in application layer, the resource described by the program may be unclear or translated into wrong resource requests. In collective layer, the request may be matched to a wrong resource because of misunderstanding or faulty matching. In fabric layer, the virtual organization may register wrong information of their resources or remove its registered resources without notifying/updating the resource management system.

1.3.Reliability Using Bayesian Networks

The Bayesian Networks (BN) can be summarized as an approach that represents the interactions among the elements in a system from a probabilistic perspective[26]. BN can be represented using directed acyclic graph where variables are represented using nodes and the relationship between the variables is represented using links. For evaluating reliability of a system, the elements of a system are represented by the variables in BN and the interaction among the elements is depicted by the directed links between two elements (forming a child and parent relationship where child being the dependent component) leading to success or failure of the whole system[14]. So the probability of the child node being successful becomes dependent or conditional on the probability of the parent node being successful. Here conditional probability of child node can be evaluated using Baye's theorem[13]. Absence of a link depicts that there is no interaction between those nodes hence they are independent of each other or in other words we can say that their probabilities can be calculated separately. The Baye's theorem can be shown using:

$$p(c|p1, p2) = \frac{p(p1, p2|c)p(c)}{p(p1, p2)} \qquad (1)$$

From a system reliability perspective, the variables of a BN are defined as the elements in the system while the links represent the interaction of the elements leading to system "success" or "failure"[15].

In a BN the interaction between the various elements of the system is represented using a directed link between two components, forming a child and parent relationship, so that the dependent component is called as the child of the other. Therefore, the success probability of a child node is conditional on the success probabilities associated with each of its parents [14]. The conditional probabilities of the child nodes are calculated by using Bayes' theorem via the probability values assigned to the parent nodes. Also, absence of a link between any two nodes of a BN indicates that these components do not interact for system failure/success and thus, they are considered independent of each other and their probabilities are calculated separately[27].

1.4.Organization Of Thesis

The chapters in this thesis are organized as follows:

Chapter 2 gives introduction about the state of the art of estimating grid service reliability with the help of Bayesian Networks.

Chapter 3 introduces the problem that has been solved in this thesis. It tells about the existing K2 algorithm and the need to improve K2 algorithm.

Chapter 4 gives the solution of problem, design of solution and implementation of the solution of the problem.

Chapter 5 describes the conclusion and future research work possible.

Chapter 2

Review of the State of the Art

The previous chapter gave a brief introduction about estimating reliability using Bayesian Networks. This chapter tells about the state of the art of estimating grid service reliability using Bayesian Networks.

2.1. Reliability Models In Grid Computing

Although reliability estimation methods for small-scale systems has been discussed in many studies, those conventional models have some common assumptions:

1. The network topology is made up of physical links (cables) and nodes (processors) that are static without considering dynamic changes of components and logic structures.
2. The operational probabilities of nodes or links are constant without considering bandwidth and information content.

These assumptions are not appropriate for grid service reliability modeling and therefore need to be relaxed.

In grid computing, communication between two nodes can be broken logically although a physical link exists. To solve these problems, Dai and Wang have used a virtual structure [14]. Hence in a grid system, the nodes represent the sites that contain the resources, and the Resource Manager (RM) sites that can process the service requested are represented as virtualized root nodes (RN). Unlike general nodes, the RN is used to provide various services: receives the service requests from the users, analyzes the service requests, starts that service, then compiles the results returned from those resources, and finally, provides outputs to the users.

Operational probabilities of the virtualized nodes and links cannot just be set at a constant value, as done by the conventional models. Rather, the operational probabilities are affected by various conditions such as[10]:

1) for links, the available bandwidth and,
2) for nodes, the time to conduct the request.

Therefore, a model combining these conditions is much closer to the reality of a grid environment and hence can handle the unpredictability of a wide-area communication. However, Dai and Wang [10] model is based on the assumptions that failures occurring on these nodes and links satisfy a Poisson process. Furthermore, their model also assumes the failures on different elements are independent from one another, which is a good approximation to the reality found in large-scale grid systems.

Furthermore, Levitin and Dai [11] consider grid computing systems in which the resource management system (RMS) divides service task into subtasks and send the subtasks to different resources for parallel execution. In order to evaluate the quality of service its reliability and performance indices should be defined. This paper introduces the indices: service reliability (probability that the service task is accomplished within a specified time) and conditional expected system time. Also it presents the numerical algorithm for the evaluation for arbitrary subtask distribution in a given grid with star architecture. It shows the existence of a trade-off between service reliability and its expected time. It is also shown that for any given number of subtasks the reliability and performance indices depend on subtask distribution among resources.

However, grid reliability analysis is not easy because of its complexity, Dai et al.[12] presents a hierarchical model for the grid service reliability analysis and evaluation. In this study the overall architecture of the grid service system is mapped to a hierarchical modelling. They have investigated various types of failures such as blocking failures, timeout failures, matchmaking failures, network failures, program failures, and resource failures to achieve a complete picture about grid service reliability. They use Markov models, Queuing theory, and Graph theory to model, evaluate, and analyze the grid service reliability.

In another study Dai et al.[13] focuses on remote accessing and information communication in grid systems. They consider communication and processing times in grid systems, which are usually ignored by other studies while estimating reliability. Also, the new model is compared with different types of conventional models [13], through which this model is verified more suitable for grid service reliability.

Similarly in another study[14], the Bayesian networks are constructed using the historical data which has been proved to be a very efficient tool. BN have significant advantages (such as efficiency in evaluating associations and simplicity in providing a system model) over traditional frameworks, partly because they are easy to use in interaction with domain experts in the reliability field. In this there is an assumption that the BN can be built by an expert who has "adequate" knowledge about the behavior of the system. However, finding such an expert may not be possible at all times for every system under consideration.

2.2. Need For Bayesian Networks

On the basis of the analysis done in the previous section, the following comparison table, Table 2.1 has been designed.

Table 2.1: Comparison between various models on the basis of some parameters

Model	**Time to Estimate Reliability**	**Cost for Estimating Reliability**	**Estimating Reliability**	**Includes sensitivity analysis**
Hierarchical Modelling (using Markov Model, Queuing Theory and Graph theory)	no	No	Maximize	no
Genetic Algorithm based model	no	minimize	Maximize	no
Bayesian Networks based model	no	minimize	Maximize	yes

As depicted by the analysis done in Table 2.1, the hierarchical modelling just maximizes reliability but does not consider time and cost used to estimate reliability whereas genetic algorithm based modelling minimizes time and maximizes reliability but does not consider cost. Both of them do not consider sensitivity analysis too[25]. On the contrary, the Bayesian networks (BN) include sensitivity analysis. In other words, Bayesian networks is a probabilistic model which is advantageous for calculating the reliability of the grid service. So Bayesian networks are useful in identifying the probabilistic relationships and between the various elements of the system. Bayesian networks are mainly used in estimating system reliability because it becomes easy to represent systems. A Bayesian network depicts the various interactions possible among the different elements in a system from a probabilistic point of view.

2.3. Reliability Estimation Using Bayesian Networks

Estimation of systems reliability using BN dates back as early as 1988, when it was first defined by Barlow [18]. The idea of using BN in systems reliability has mainly gained acceptance because of the simplicity it allows to represent systems and the efficiency for obtaining component associations. The concept of BN has been discussed in several earlier studies. More recently, BN have found applications in, software reliability, fault finding systems, and general reliability modelling[28].

Currently, predefined BN are used for reliability estimation for specific systems. Gran and Helminen [19] provide a BN for nuclear power plants and introduce a hybrid method for estimating the reliability of the plant. They assume the existence of a BN that models the system and introduce methods for estimating prior probabilities and assessing the system reliability accordingly.

In addition to these, Amasaki et al. [20] uses BN for software quality assessment. They modeled the phases of a software system as a BN, and by using this model they observed the faults that may occur in their system. They used the actual data and created sensitivity analysis of the BN model that they constructed. Alternatively Boudali and Dugan [21] introduce a method for reliability assessment in dynamic systems by using temporal BN; where the system components change states at different time intervals. The great majority of BN assume that the prior probabilities and the structure of the BN are already known.

2.3.1. Constructing Bayesian Networks

This methodology uses the K2 algorithm[17] for constructing the BN automatically using historical data. This algorithm searches for the parent set for a node that has the maximum association with it. The K2 algorithm is composed of two main factors: a scoring function to quantify the associations and rank the parent sets according to their scores, and a heuristic to reduce the search space to find the parent set with highest degree of association [15]. This algorithm takes observational data set as an input and generates a Bayesian network as an output. Hence with K2 algorithm, we have a methodology that utilizes historical data to construct a BN model.

2.3.2. Estimating Reliability Using Bayesian Networks

Here, each element is assigned with an individual CPT (conditional probability table) within the BN. Each CPT of a given element X contains the conditional probability, p(X|S), where S is the set of X's parents. In each CPT the parents are assigned as either "Success" and "Failure"; so for x parents there will be 2^x different parent set assignments; thus 2^x entries in CPT[14]. A complete BN can be constructed using all the calculated conditional probabilities. This concept can be explained using Fig 2.1.

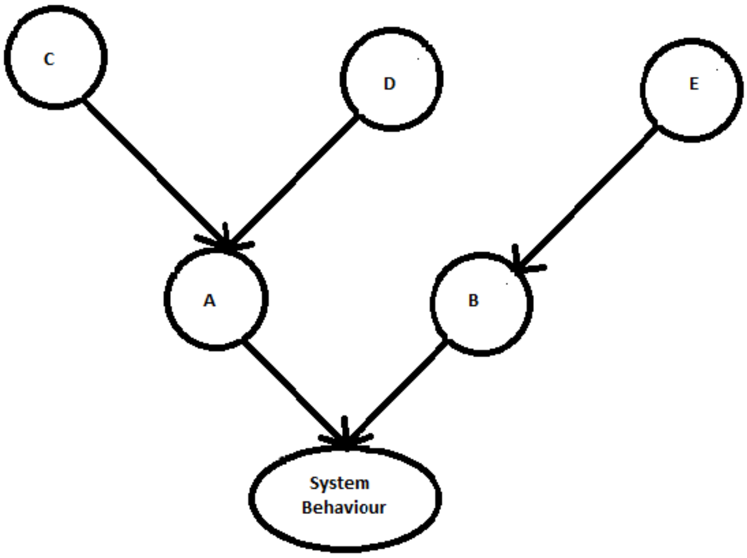

Fig 2.1: A sample Bayesian Network

Fig 2.1 shows how various components A, B, C, D and E of a system interact with each other. For this BN the child parent relationships of the components can be observed from the arrows shown. Quantitatively, the degrees of these associations are expressed as probabilities and each node is associated with a CPT.

As shown in the Fig 2.1, the topmost nodes (C, D and E) do not have any incoming edges, therefore they are independent of the rest of the elements in the system whereas the nodes A and B are dependent upon C, D and E. The prior probabilities that are assigned to these nodes should be known using historical data about the system[14]. Using these prior probabilities, the success probability of the dependent nodes, like A, can be calculated using (1). So, the probability of the node A is dependent on its parents, C and D as shown in Fig 2.1. Overall system reliability can also be calculated by using 1.

2.4.Estimating Grid Reliability Using Bayesian Networks

The first step is to find the Minimal Resource Spanning Trees (MRSTs). These are formed from a network when a path is required to fulfil a resource request keeping into consideration the usage of minimum number of resources. The K2 algorithm, as explained in next section, is used to construct the BN model for each MRST. Here historical data is used for estimating node and link reliabilities with the help of the BN model[13]. Here the focus is on reliabilities of MRSTs in a grid system to evaluate the overall grid service reliability. So, each MRST acts as a smaller size grid system [9] and therefore can be modeled and evaluated separately. Once a historical system dataset is available, the K2 algorithm can be used to construct BN for each MRST. Each row in historical data shows the state of the components at some specific point of time. For simplifying our methodology, it is assumed that the component failure data follows a binary behavior. That is, for each component in the MRST the value of 0 represents failure while the value of 1 represents full functionality. Also, the last column depicts the information about the overall MRST Behavior (0 depicts failure and 1 depicts availability of the grid service through MRST).The K2 algorithm uses this historical data to find associations between the various elements of the system and finally produces the BN as an output[13].

Step 1: The first step picks up the first element and evaluates its possible parents according to the ordered set of nodes i.e. K2 algorithm iterates on the elements according to their ordering in the dataset. For this the K2 algorithm computes the function f (scoring function) using the possible parent sets and the results are compared such that the set with highest f value will be chosen as the parent set.

As and when the iterations of K2 algorithm follow, the number of possible parent sets and hence f score calculations increases[13]. After certain iterations when the algorithm finishes with the last column, the BN structure is displayed as an output.

Step 2: The next step of the methodology uses the BN constructed in the first step to estimate the system reliability[14]. In this step all the conditional probabilities are calculated and stored in a table called CPT (conditional probability table). Here the components that have one or more parent sets are associated with conditional probabilities and the others with no parent set are associated with prior probabilities. If n is the number of parents of that component then each CPT has 2^n entries. Here, the probability values are calculated using the historical data[13].

Now as it can be seen that the accuracy of the reliability of this MRST increases as the number of the observations increases, hence for getting a more accurate result we need to have more number of observations in historical dataset[14]. So we can construct BN of other MRSTs and calculate their reliabilities using same method. Then using Baye's rule, the overall grid service reliability can be calculated.

2.5.K2 Algorithm

Although there are various algorithms for learning Bayesian networks from data, they can be subdivided into two general approaches: methods based on conditional independence tests, and methods based on a scoring function and a search procedure. There are also hybrid algorithms that use a combination of independence-based and scoring-based methods[22].

The algorithms based on independence tests perform a qualitative study of the dependence and independence relationships between the variables in the domain, and attempt to find a network that represents these relationships as far as possible. The examples of algorithms based on independent tests are Power Constructor and Bayesian Network Power Constructor. The algorithms based on a scoring function (also called a metric) attempt to find a graph that maximizes the selected score; the scoring function is usually defined as a measure of fit between the graph and the data[22].

K2 is a scoring function and a search procedure based algorithm. This kind of method starts from an initial DAG and, at each step, performs the local change (operator) yielding the maximal gain, until a local maximum of the scoring function is reached. The algorithm is as follows[17]:

i. procedure K2;
ii. {Input: A set of n nodes, an ordering on the nodes, an upper bound u on the number of parents a node may have, and a database D containing m cases.}
iii. {Output: For each node, a printout of the parents of the node and reliability of MRST}
iv. for i:= 1 to n do
v. $\Pi_i := \phi$;
vi. $P_{old} := f(i, \Pi_i)$; {This function is computed using 2.}
vii. OKToProceed := true;
viii. While OKToProceed and $| \Pi_i | < u$ do
ix. let z be the node in $Pred(x_i) - \Pi_i$ that maximizes $f(i, \Pi_i \cup \{z\})$;

```
            P_new := f(i, Π_i ∪ {z});
            if P_new > P_old then
                  P_old := P_new;
                  Π_i := Π_i ∪ {z};
            else OKToProceed := false;
      end {while};
      write('Node: ', x_i, ' Parent of x_i: ',Π_i);
   end {for};
   end {K2};
```

$$f(i, \Pi_i)=\prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \qquad (2)$$

where:

Π_i: set of parents of node x_i

$q_i = |\phi_i|$

ϕ_i: list of all possible instantiations of the parents of x_i in database D. That is, if $p_1,\ldots, p_s$ are the parents of x_i then ϕ_i is the Cartesian product $\{v_1^{p_1},\ldots, v_{r_{p_1}}^{p_1}\}$ x ... x$\{v_1^{p_s},\ldots, v_{r_{p_s}}^{p_s}\}$ of all the possible values of attributes p_1 through p_s.

$r_i = |V_i|$

V_i: list of all possible values of the attribute x_i

α_{ijk}: number of cases (i.e. instances) in D in which the attribute x_i is instantiated with its k^{th} value, and the parents of x_i in Π_i are instantiated with the j^{th} instantiation in ϕ_i.

$N_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, That is, the number of instances in the database in which the parents of x_i in Π_i are instantiated with the j^{th} instantiation in ϕ_i.

The informal intuition here is that $f(i, \Pi_i)$ is the probability of the database D given that the parents of x_i are Π_i.

Chapter 3
Problem Statement

This chapter introduces the problem that has been solved in this thesis. It tells about the existing K2 algorithm and the need to improve K2 algorithm.

3.1. Existing K2 algorithm

K2 is an algorithm used for constructing a Bayesian Network (BN) from a database of records. This algorithm heuristically searches for the most probable belief–network structure given a database of cases. The main advantage of using K2 algorithm for constructing a BN is that it is a score function based algorithm and hence it does not require use of experts' advice to construct networks.

3.2. Need For Extending K2 Algorithm

The main need to improve this algorithm is that it does not include reliability calculation. Also this algorithm has not been implemented in any of the programming environments. Hence an easy to use interface is required for this algorithm and its enhanced version needs to include calculation of reliability of an MRST.

3.3. Objectives of Thesis

- To analyze and compare the available reliability models in grid computing.
- To assess the probabilistic relationships and identifying probabilistic mappings between system components using BN.
- To implement the improved version of K2 algorithm using Java.

Chapter 4

Design and Implementation

The previous chapter discusses about the problem which is being solved in this thesis. This chapter discusses about design and implementation of the solution of the problem.

4.1.Extended K2 Algorithm

The K2[17] is an algorithm used for constructing a Bayesian Network from a database of records. This algorithm heuristically searches for the most probable belief–network structure given a database of cases. The extended K2 algorithm, after finding the probabilistic relationships, finds the reliability of the MRST. The extended algorithm is as follows:

i. procedure XK2;

ii. {Input: A set of n nodes, an ordering on the nodes, an upper bound u on the number of parents a node may have, and a database D containing m cases.}

iii. {Output: For each node, a printout of the parents of the node and reliability of MRST}

iv. for i:= 1 to n do

v. $\Pi_i := \phi$;

vi. $P_{old} := f(i, \Pi_i)$; {This function is computed using 2.}

vii. OKToProceed := true;

viii. While OKToProceed and $| \Pi_i | < u$ do

ix. let z be the node in Pred(x_i) - Π_i that maximizes $f(i, \Pi_i \cup \{z\})$;

x. $P_{new} := f(i, \Pi_i \cup \{z\})$;

xi. if $P_{new} > P_{old}$ then

xii. $P_{old} := P_{new}$;

xiii. $\Pi_i := \Pi_i \cup \{z\}$;

xiv. else OKToProceed := false;

xv. end {while};

xvi. write('Node: ', x_i, ' Parent of x_i: ',i);

xvii. $Prob_i$=p(c|p1,p2); {This function is computed using 3 and this is an addition to the original algorithm}

xviii. end {for};

xix. Reliability=Prob$_{MRST}${This is also an addition to the original algorithm}

xx. end {K2};

$$f(i, \Pi_i)=\prod_{j=1}^{q_i} \frac{(ri-1)!}{(Nij+ri-1)!} \prod_{k=1}^{r_i} \alpha_{ijk}! \quad (2)$$

where:

Π_i: set of parents of node x_i

$q_i = |\phi_i|$

ϕ_i: list of all possible instantiations of the parents of x_i in database D. That is, if $p_1, ..., p_s$ are the parents of x_i then ϕ_i is the Cartesian product $\{v_1^{p1}, ..., v_{r_{p_1}}^{p_1}\}$x ... x$\{v_1^{p_s}, ..., v_{r_{p_s}}^{p_s}\}$ of all the possible values of attributes p_1 through p_s.

$r_i = |V_i|$

V_i: list of all possible values of the attribute x_i

α_{ijk}: number of cases (i.e. instances) in D in which the attribute x_i is instantiated with its k^{th} value, and the parents of x_i in Π_i are instantiated with the j^{th} instantiation in ϕ_i.

$N_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$, That is, the number of instances in the database in which the parents of x_i in Π_i are instantiated with the j^{th} instantiation in ϕ_i.

$$p(c|p1, p2) = \frac{p(p1,p2|c)p(c)}{p(p1,p2)} \quad (3)$$

where

c is variable whose conditional probability is required

p1 and p2 are the variables upon which c depends

In this extended algorithm, the main extensions that have been done are to include conditional probabilities using Baye's theorem as depicted in lines xvii and xix.

4.2.Flowchart for Extended K2 Algorithm

The following flowchart depicts the extended K2 algorithm. It shows the main steps that have been performed in the implementation. All the processes are shown in order in which they are implemented. The Fig 4.1 depicts the flowchart of the extended K2 algorithm:

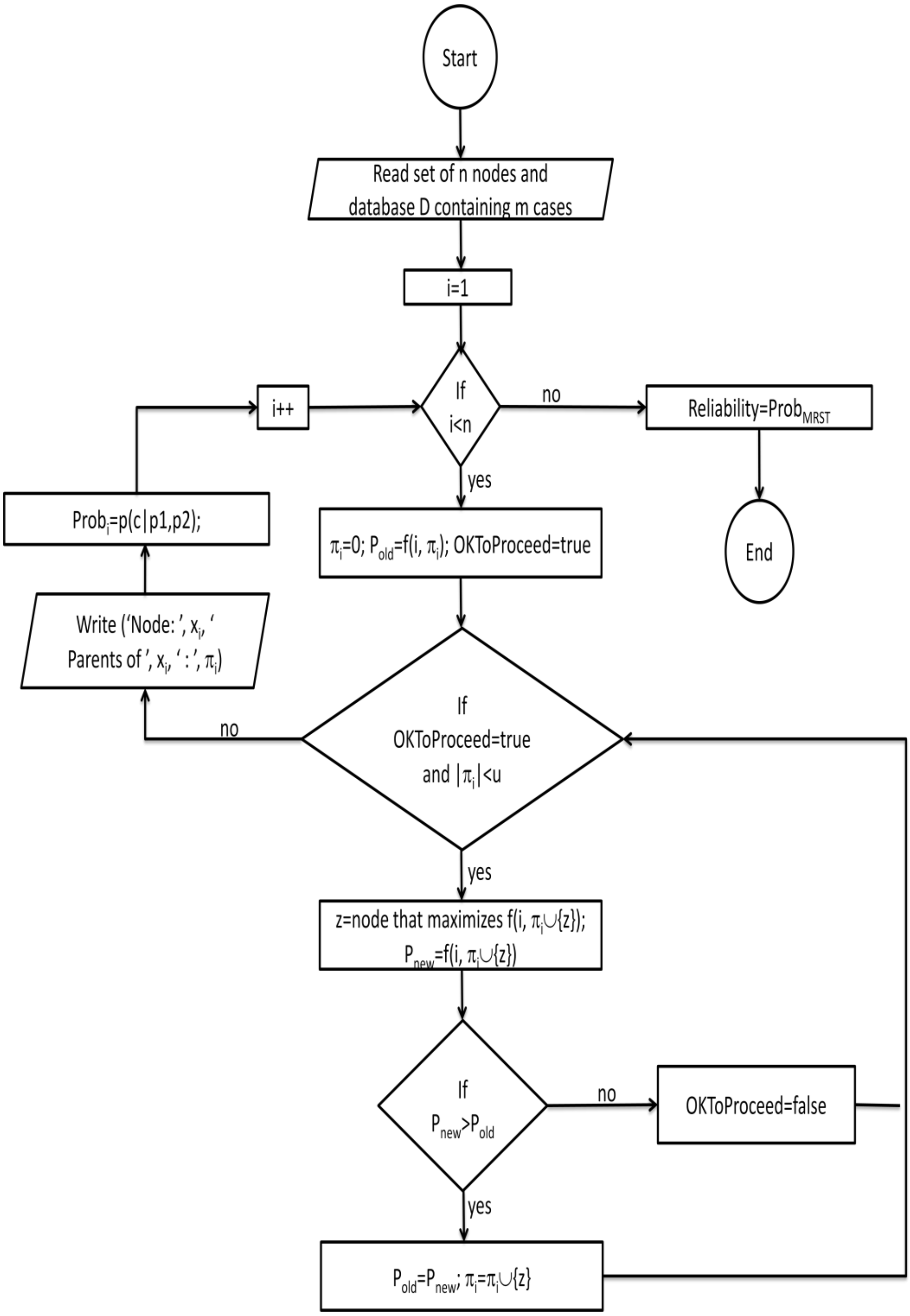

Fig 4.1: Flowchart for extended K2 algorithm

4.3. Implementation

Grid computing utilizes unused processing cycles of all computers in a network for solving problems that are too intensive for any stand-alone machine[2]. To evaluate the reliability for any given grid service, the links between all the nodes and various resources need to be identified. These sets of nodes and links utilized in running a service form a spanning tree. By calculating the reliability of these spanning trees, the grid service reliability can easily be determined[6]. A spanning tree that connects a root node or resource manager to other nodes using links, such that the nodes contain all the resources utilized by the grid service, is known as resource spanning tree(RST). The number of RSTs can be very large but minimum resource spanning tree(MRST) is an RST that includes each resource only once i.e. it does not contain any redundant resource. So while selecting MRSTs, all the RSTs having redundant resources need to be removed. The reliability of a service in a grid system can be evaluated by using the reliabilities of services through MRST[8].

Case 1: Three Node System

In this system, there are three nodes each with grid networks. Each network has specific number of resources. Fig 4.2 depicts a sample three-node grid system which we will be using in our first case.

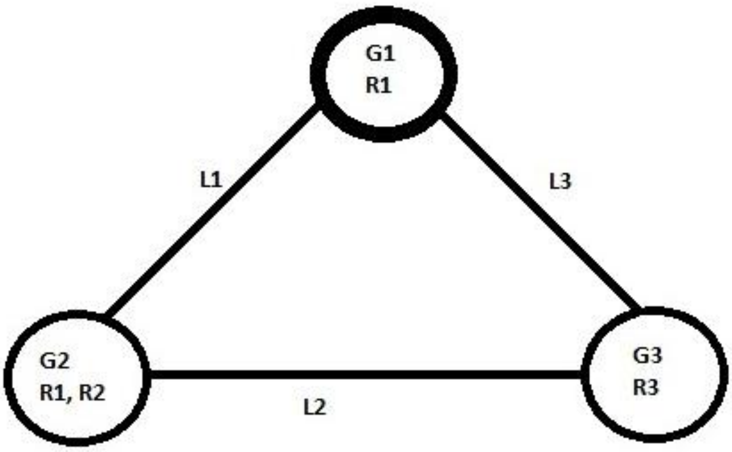

Fig 4.2: Grid Network for Case 1

As can be seen in the Fig 4.2, when the root node G1 requests R1 and R3, there are several paths to fulfil these requests. Some of the RSTs that include the requested resources are {G1, G2, G3} and {G1, G3}. Here, {G1, G3} is an MRST but {G1, G2, G3} is not, because {G1, G3} already covers all the requested resources.

Now the MRST {G1,G3} has been considered. It forms one of the MRSTs with two nodes (G1 and G3) and one link (L3). Thus it can be considered as a system with three components and a BN model for this MRST can be created. So, each element in the MRST is represented using a separate node in this BN model. The historical dataset for this MRST is depicted in Table 4.1. It shows all the three components of the example MRST represented in different columns (Gs represent nodes and Ls represent links).

Table 4.1: Historical dataset for case 1

Observation	G1	L3	G3	MRST Behaviour
1.	0	1	1	0
2.	0	1	0	0
3.	1	0	0	0
4.	1	0	1	0
5.	0	0	0	0
6.	0	0	1	0
7.	1	1	1	1
8.	1	1	0	0
9.	0	1	1	0
10.	1	1	1	1

After performing step 1 in the methodology mentioned in chapter 2, a BN structure is developed which is depicted in Fig 4.3.

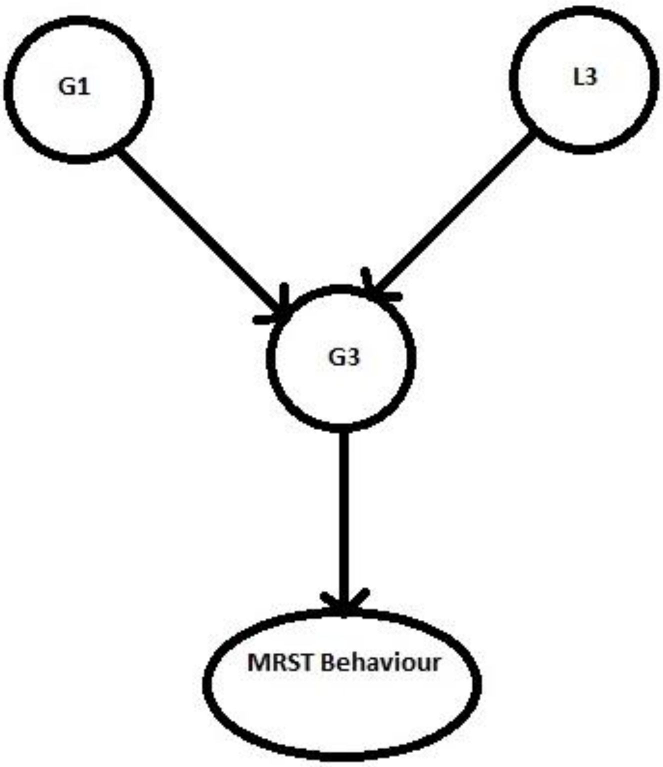

Fig 4.3: BN Structure for case 1

Here it can be seen from the BN in Fig 4.3 that G3 is dependent on G1 and L3, and from Table 4.1, p(G3=1|G1=1,L3=1)=0.63. Hence the success probability for MRST Behaviour node can be calculated as 0.63 or 63%. So the reliability of this MRST is 63%.

Case 2: Four Node Grid System

In this system, there are four nodes each with grid networks. Each network has specific number of resources. Fig 4.4 depicts a sample four-node grid system which we will be using in our second case.

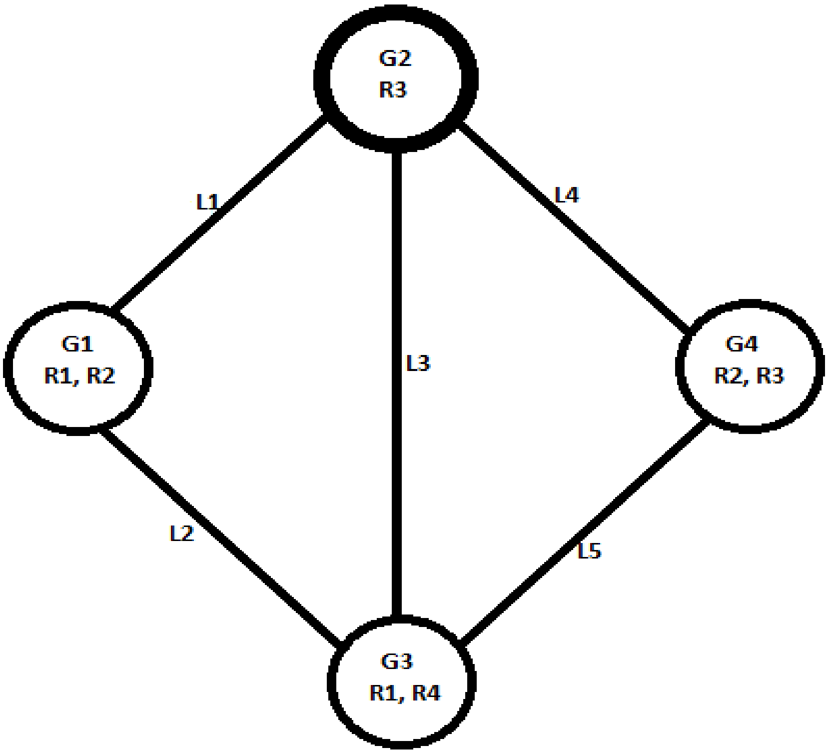

Fig 4.4: Grid Network for Case 2

As can be seen in the Fig 4.4, when the root node G1 requests R1 and R3, there are several paths to fulfil these requests. Some of the RSTs that include the requested resources are {G1, G2}, {G1, G3, G4}, {G1, G2, G4} and {G1, G2, G3, G4}. Here {G1, G2} is an MRST but {G1, G2, G4} is not, because {G1, G2} already covers all the requested resources.

Now here the MRST {G1,G2} has been considered. It forms one of the MRSTs with two nodes (G1 and G2) and one link (L1). Thus it can be considered as a system with three components and a BN model for this MRST can be created. So, each element in the MRST is represented using a separate node in this BN model. The historical dataset for this MRST is depicted in Table 4.2. It shows all the three components of the example MRST represented in different columns (Gs represent nodes and Ls represent links).

Table 4.2: Historical dataset for case 2

Observation	G1	L1	G2	MRST Behaviour
1.	0	1	1	0
2.	0	1	0	0
3.	1	0	0	0
4.	1	0	1	0
5.	0	0	0	0
6.	0	0	1	0
7.	1	1	1	1
8.	1	1	0	0
9.	0	1	1	0
10.	1	0	0	0

After performing step 1 in the methodology mentioned in chapter 2, a BN structure is developed which is depicted in Fig 4.5.

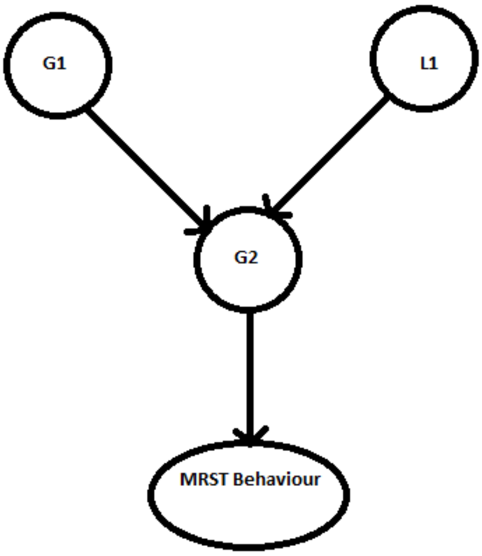

Fig 4.5: BN Structure for case 2

Here it can be seen from the BN in Fig 4.5 that G2 is dependent on G1 and L1, and from Table 4.2, p(G2=1|G1=1,L1=1)=0.67. Hence the success probability for MRST Behaviour node can be calculated as 0.67 or 67%. So the reliability of this MRST is 67%.

Case 3: Five Node Grid System

In this system, there are five nodes each with grid networks. Each network has specific number of resources. Fig 4.6 depicts a sample five-node grid system which we will be using in our second case.

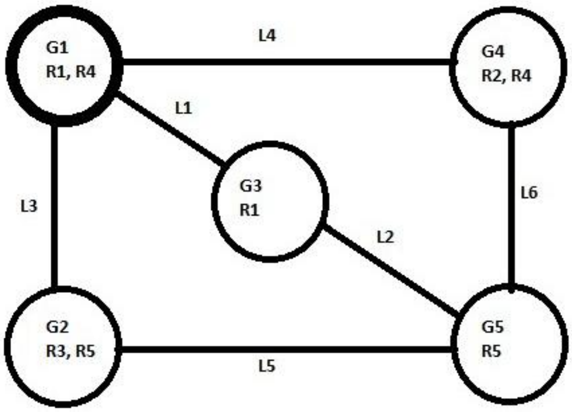

Fig 4.6: Grid Network for Case 3

As can be seen in the Fig 4.3, when the root node G1 requests R1 and R5, there are several paths to fulfil these requests. Some of the RSTs that include the requested resources are {G1, G2}, {G1, G3, G5} and {G1, G4, G5}. Here {G1, G2} is an MRST but {G1, G2, G5} is not, because {G1, G2} already covers all the requested resources.

Now the MRST {G1,G3, G5} has been considered here. It forms one of the MRSTs with three nodes (G1, G3 and G5) and two links (L1 and L2). Thus it can be considered as a system with five components and a BN model for this MRST can be created. So, each element in the MRST is represented using a separate node in this BN model. The historical dataset for this MRST is depicted in Table 4.3. It shows all the three components of the example MRST represented in different columns (Gs represent nodes and Ls represent links).

Table 4.3: Historical dataset for case 3

Observation	G1	L1	G3	L2	G5	MRST Behaviour
1.	0	0	1	1	0	0
2.	1	0	0	0	0	0
3.	1	0	0	1	1	0
4.	1	1	1	1	1	1
5.	1	1	1	1	1	1
6.	0	1	0	0	1	0
7.	0	0	0	0	0	0
8.	1	1	1	1	1	1
9.	0	1	0	1	0	0
10.	1	0	0	1	1	0

After performing step 1 in the methodology mentioned in chapter 2, a BN structure is developed which is depicted in Fig 4.7.

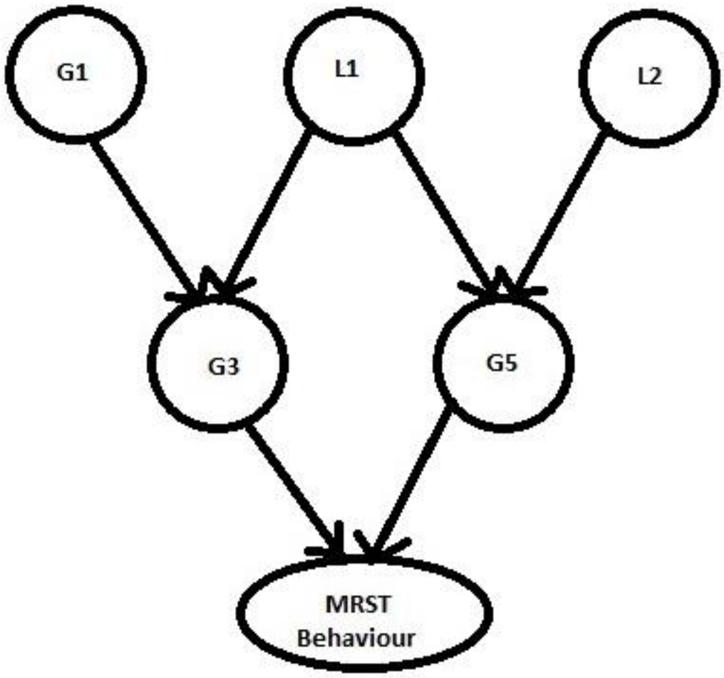

Fig 4.7: BN Structure for case 3

From the Table 4.3, G3 being 0 given the parent instantiations as G1=0 and L1=0 is 0.5, as two out of ten observations parents are instantiated as 0 and 0; and for one of these cases G3 is instantiated as 0. In the next step, with the help of CPT and the prior probabilities that G1 and L1 have, the success probability value for G3 can be calculated. According to the BN structure (in Fig 4.7) components G1 and L1 are independent of others; therefore their success probabilities can be directly inferred from the observations dataset in Table 4.3. Hence it can be evaluated that p(G1=1)=0.6 and p(L1=1)=0.5. When we continue the computations for the other components in the network, success probabilities for the rest of the components in the sample MRST can be evaluated; such that p(L2=1)=0.6 and p(G5=1)=0.75. In the last step, the MRST reliability can be calculated by using these probability values and the CPT of the MRST Behavior node in the BN structure given in Fig 4.7. The success probability of the MRST Behavior node can be calculated as 0.35 or 35%; which is the reliability of the MRST.

4.4. Installation Of Pre-requisites and Necessary Components

4.4.1. Eclipse IDE

Eclipse is an Open Source community whose projects are focused on providing a vendor-neutral Open development platform and application frameworks for building software. The Eclipse Foundation is a not-for-profit corporation formed to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an Open Source community and an ecosystem of complementary products, capabilities, and services. The Eclipse [19] IDE can be downloaded from the homepage of Eclipse Foundation [19].

The IDE can be installed easily following the on screen messages. The next step is to configure the Eclipse IDE for the implementation of the algorithm. The steps to be followed are:

- Make sure JDK 1.5 is installed on the system.
- Change the runtime environment to JRE 1.5 from that of default Eclipse.

4.5.Experimental Results

This thesis depicts the implementation of the extended K2 algorithm which takes the MRST and the historical grid data as an input and tells the probabilistic relationship (child parent relationship) between those nodes which will help to form Bayesian Networks. After telling the relationship between the nodes this algorithm tells the reliability of the MRST. This algorithm has been implemented using Eclipse IDE for Java. The implementation flows as follows:

- The input is taken from the user in the form of number of nodes and number of observations.
- The historical data is then entered by the user. In case the user tries to enter the data which is greater than the number of observations, a prompt box appears which warns the user.
- When the user clicks on Show button, the relationship between the nodes appear which is required to form a Bayesian Network. This relationship is the optimal relationship which is formed after analyzing the grid historical data.
- When the reliability button is clicked the reliability of the MRST is calculated and returned to the user.

Case 1: Three Node Grid System

Fig 4.8 depicts the page of the implementation of the algorithm before the user enters the number of nodes and number of observations. This is the first page which appears when the implementation is executed.

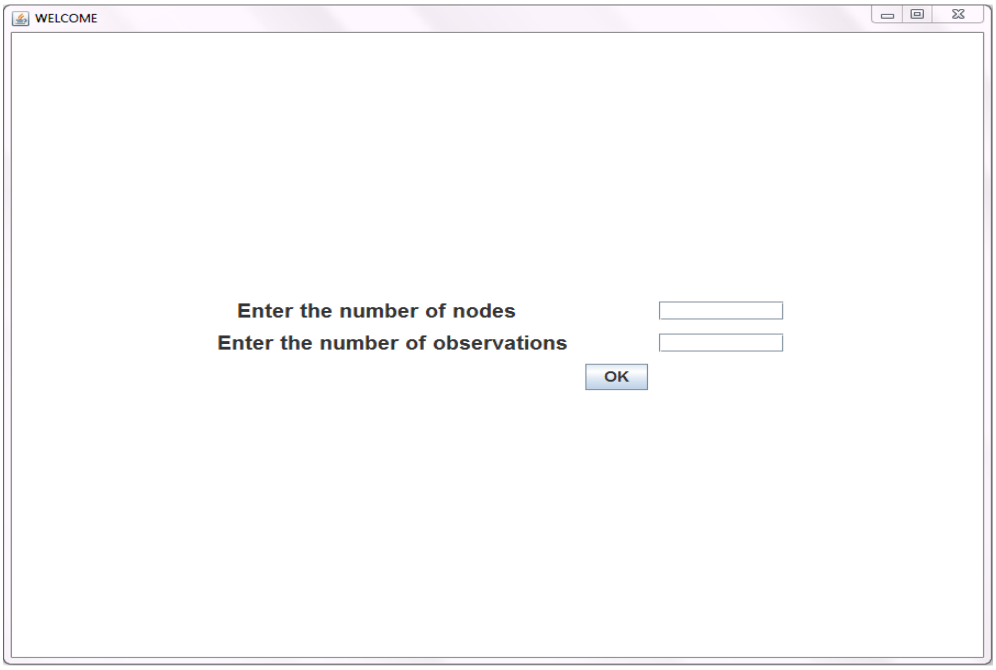

Fig 4.8: First page before entering no. of nodes and no. of observations

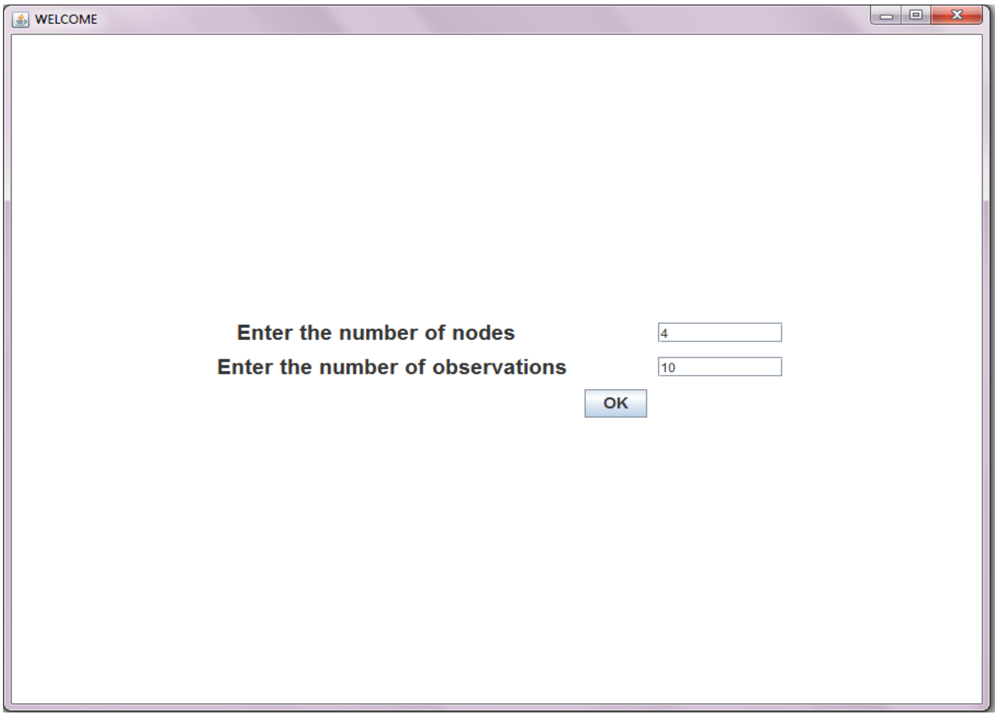

Fig 4.9: First page for case 1 after entering no. of nodes and no. of observations

When the user enters the required information i.e. the number of nodes and number of observations, the page for case 1 is as depicted in Fig 4.9. After this the user clicks the OK button and reaches to the next page as shown in Fig 4.10.

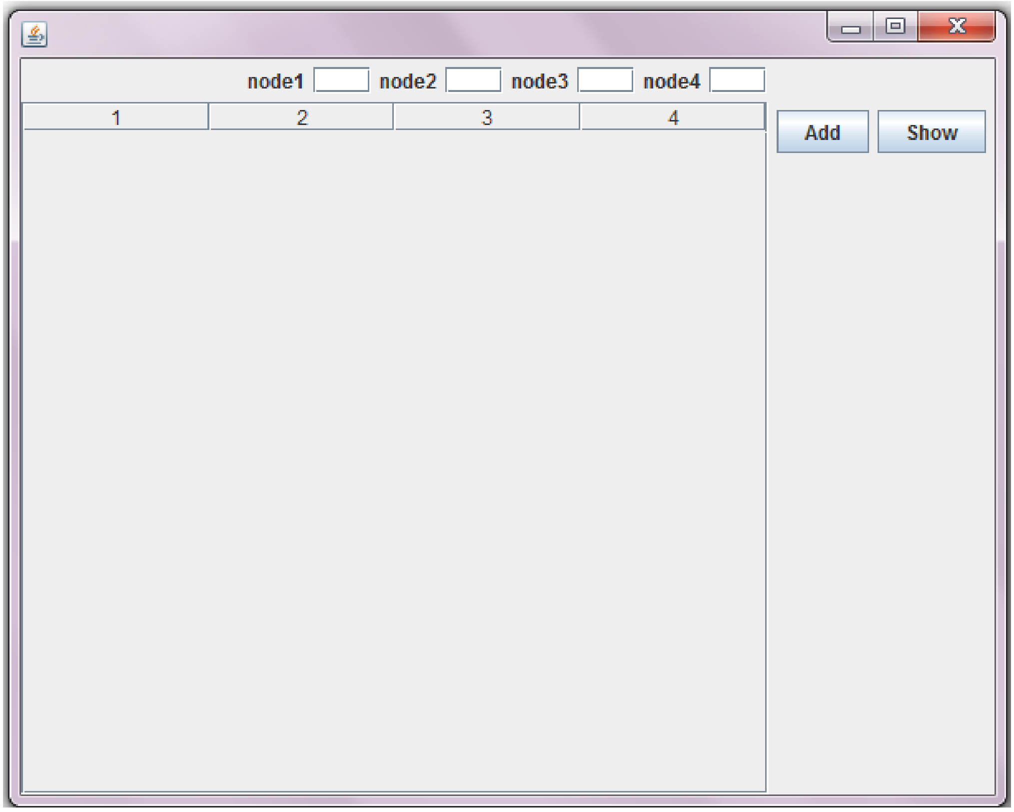

Fig 4.10: Page after clicking OK button

Fig 4.10 depicts the page after clicking the OK button in the previous page. Now the user is required to add the grid historical data observation by observation. Here the user is allowed to enter only 10 observations as mentioned in the previous page.

Fig 4.11 depicts the page when the user has entered data by entering the information in above text boxes. As the user clicks on Add button, the observation appears in the row of the table as depicted Fig 4.11.

When the user tries to enter the data which is greater than the number of observations as entered by user on first page, the prompt box appears, as depicted in Fig 4.12, which warns the user.

After taking the input the child parent relationship between the various nodes of an MRST appear as shown in Fig 4.13. This relationship is actually required to form a Bayesian Network. When the user clicks on Show BN button the next page appears which is depicted in Fig 4.14.

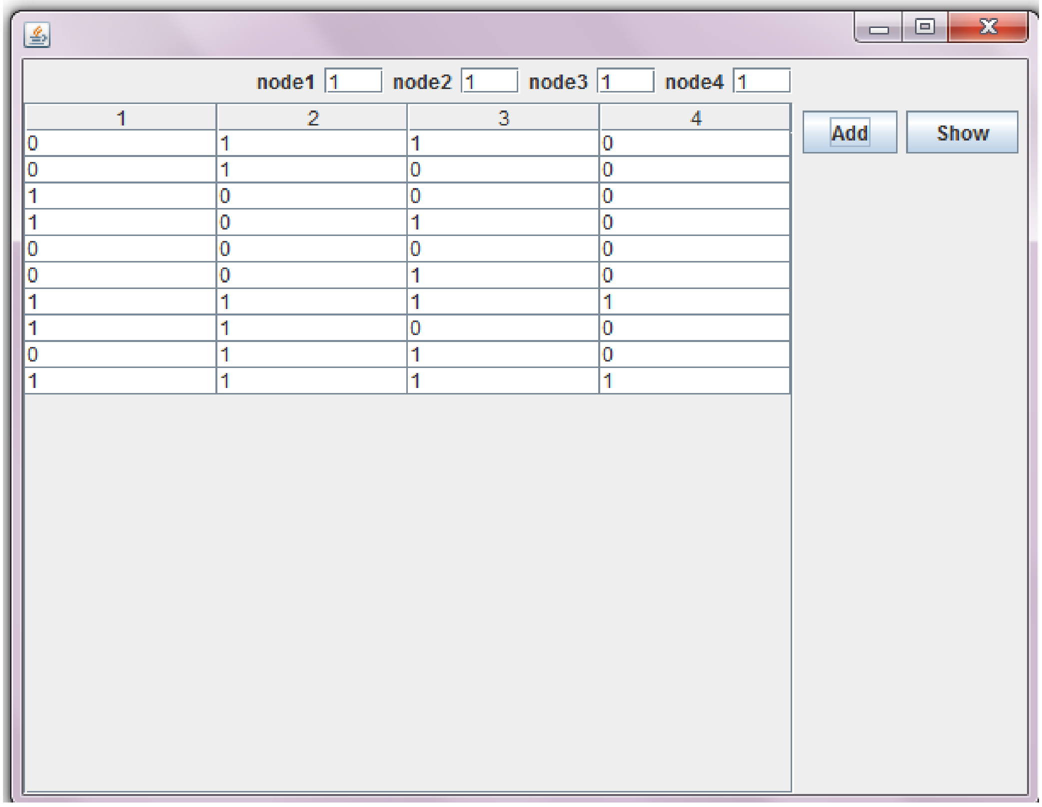

Fig 4.11: Page after entering the grid historical data for case 1

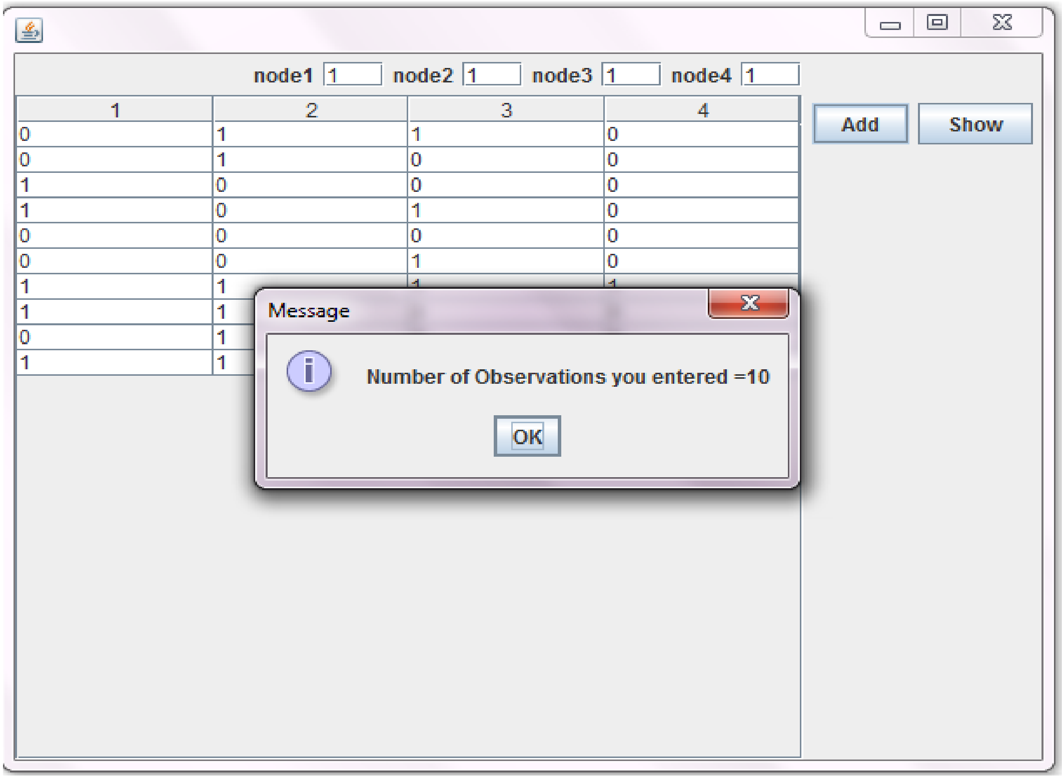

Fig 4.12: Prompt box that appears when the user tries to enter more number of observations than as mentioned earlier

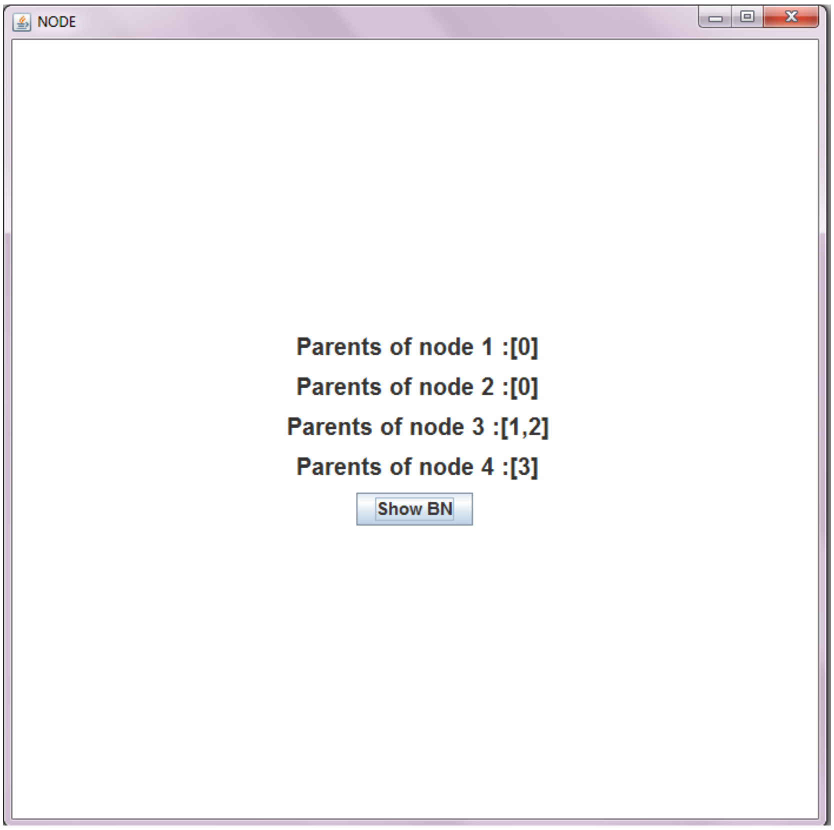

Fig 4.13: Page showing relationship between the nodes of case 1

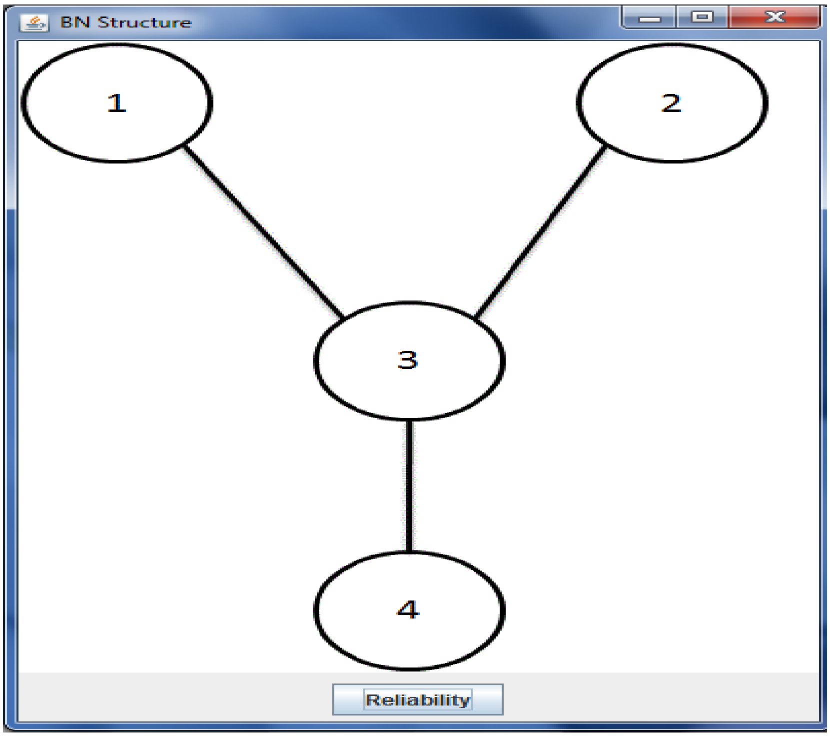

Fig 4.14: BN structure for case 1

Fig 4.14 depicts the snapshot of the BN structure for case 1.

Fig 4.15 depicts the page that depicts the reliability of the MRST. This page appears when the user clicks on Reliability button on previous page.

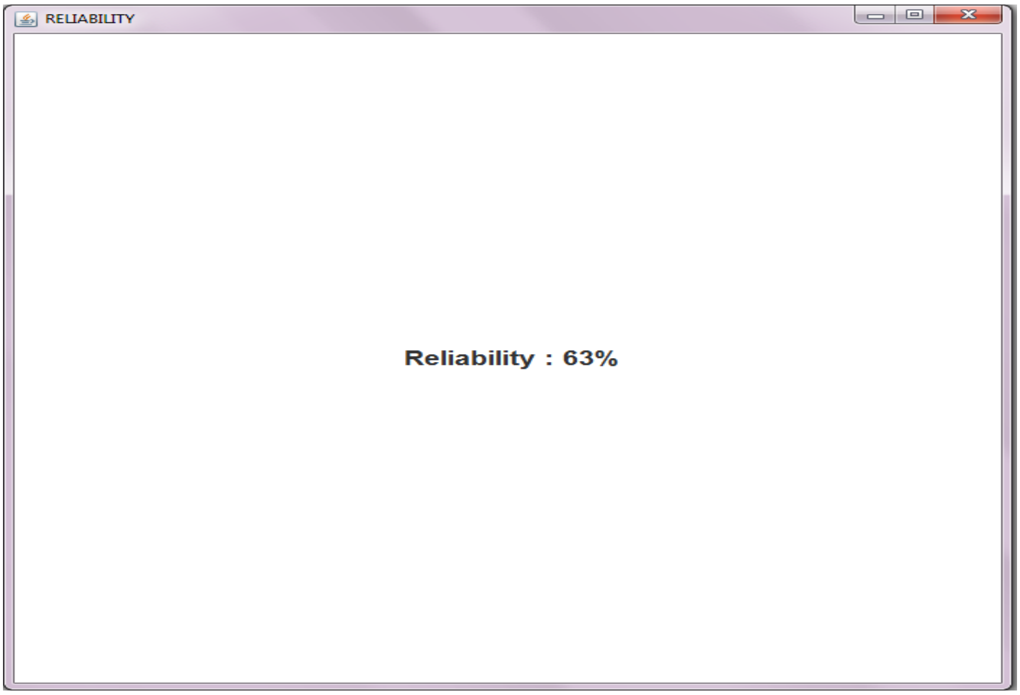

Fig 4.15: Page showing the calculated reliability of MRST of case 1

Case 2: Four Node Grid System

Fig 4.16 depicts the page of the implementation of the algorithm before the user enters the number of nodes and number of observations. This is the first page which appears when the implementation is executed.

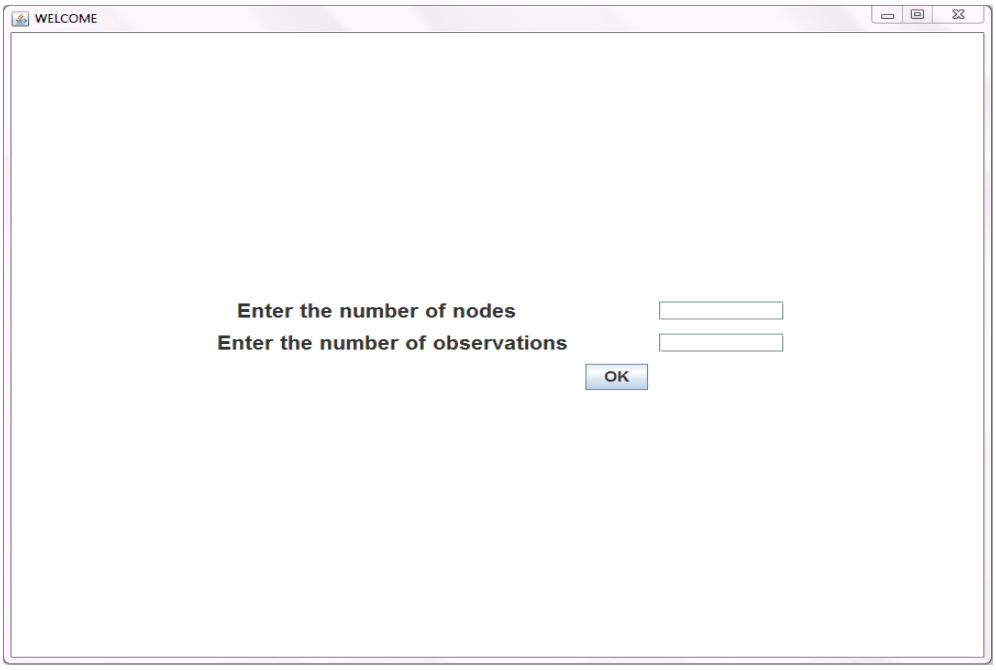

Fig 4.16: First page before entering no. of nodes and no. of observations

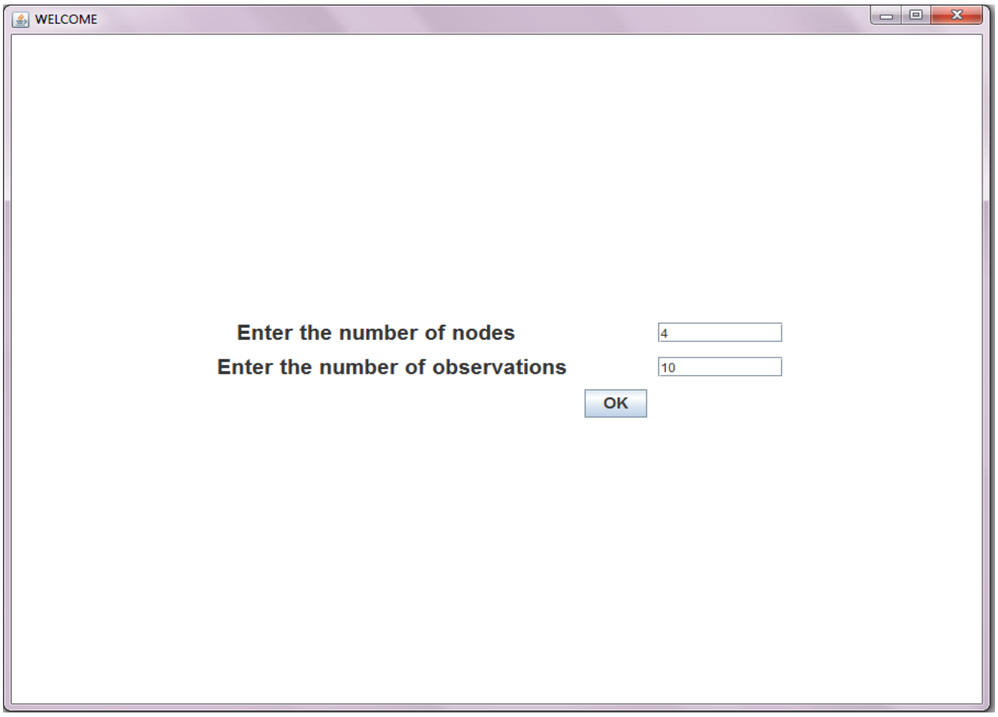

Fig 4.17: First page for case 2 after entering no. of nodes and no. of observations

When the user enters the required information i.e. the number of nodes and number of observations, the page for case 2 is as depicted in Fig 4.17. After this the user clicks the OK button and reaches to the next page as shown in Fig 4.18.

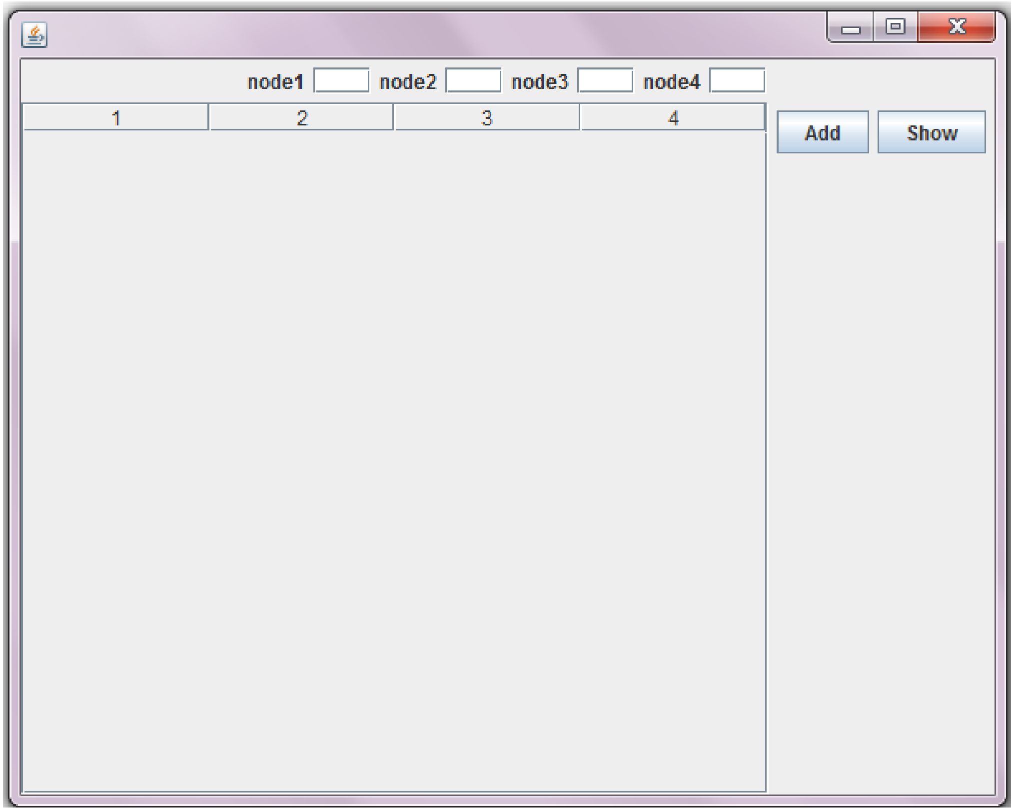

Fig 4.18: Page after clicking OK button

Fig 4.18 depicts the page after clicking the OK button in the previous page. Now the user is required to add the grid historical data observation by observation. Here the user is allowed to enter only 10 observations as mentioned in the previous page.

Fig 4.19 depicts the snapshot of the page when the user has entered data by entering the information in above text boxes. As the user clicks on Add button, the observation appears in the row of the table as depicted in above figure.

When the user tries to enter the data which is greater than the number of observations as entered by user on first page, the prompt box appears, as depicted in Fig 4.20, which warns the user.

After taking the input the child parent relationship between the various nodes of an MRST appear as shown in Fig 4.21. When the user clicks on Show BN button the next page appears which is depicted in Fig 4.22.

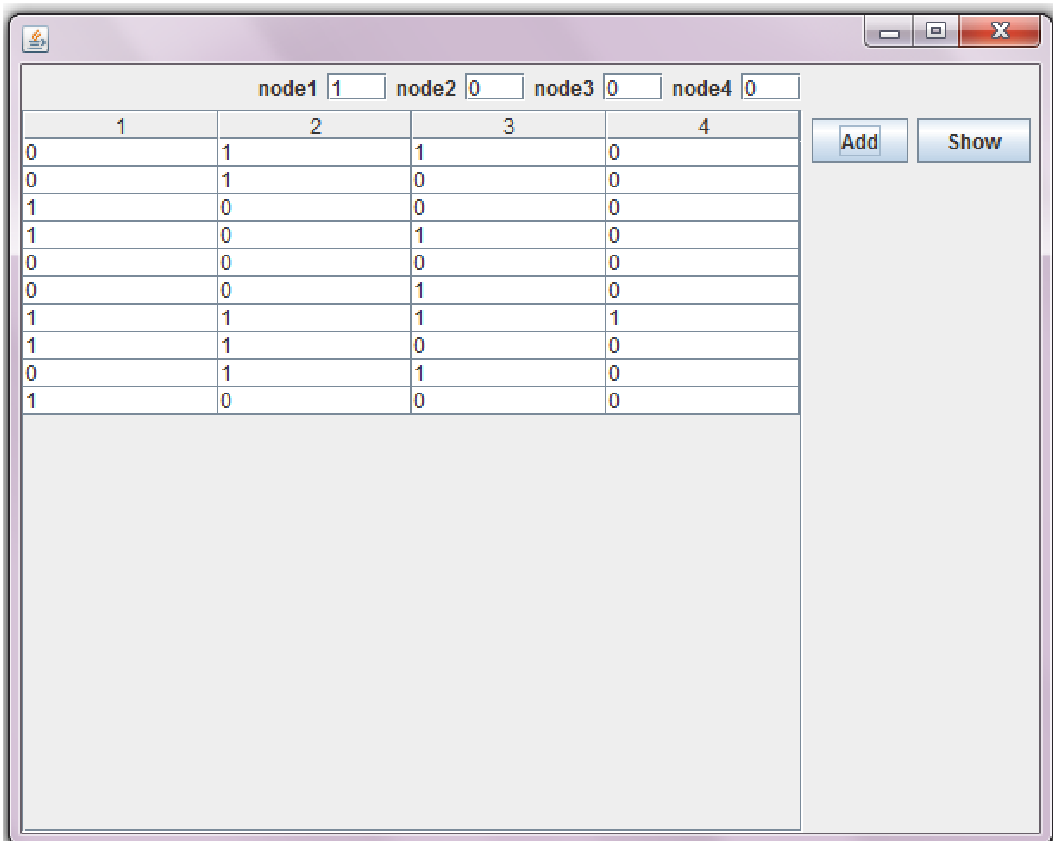

Fig 4.19: Page after entering the grid historical data for case 2

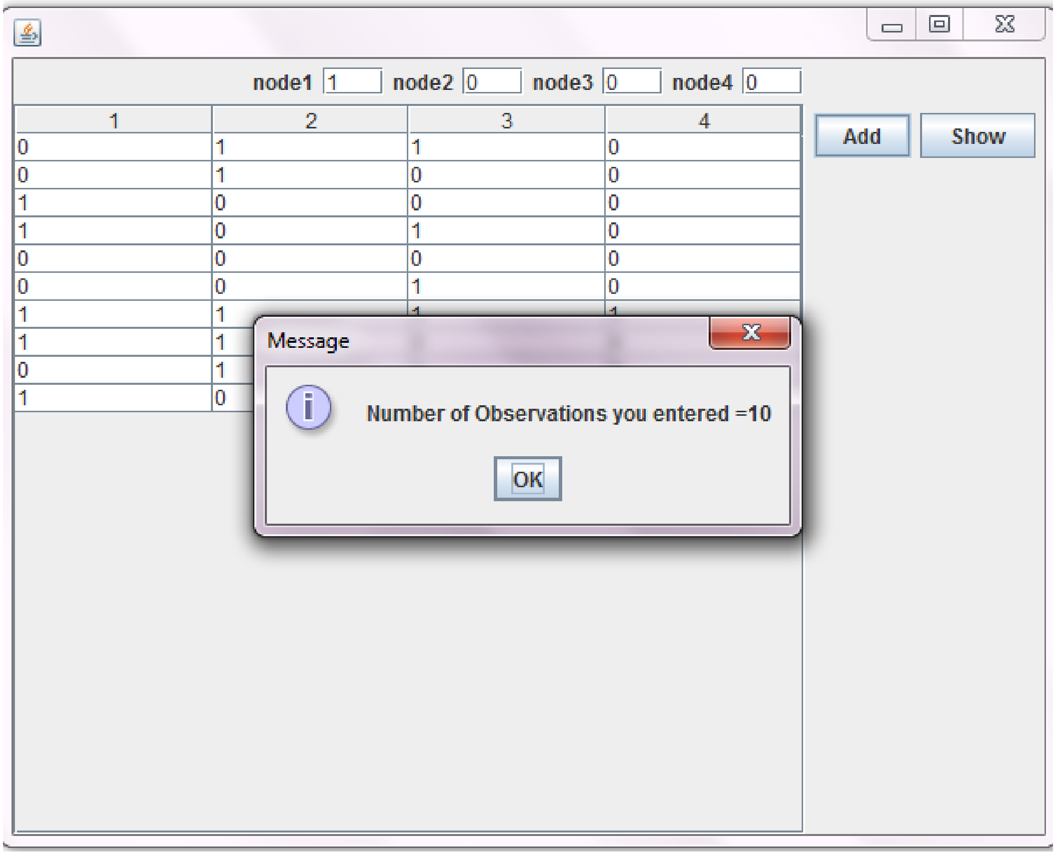

Fig 4.20: Prompt box that appears when the user tries to enter more number of observations than as mentioned earlier

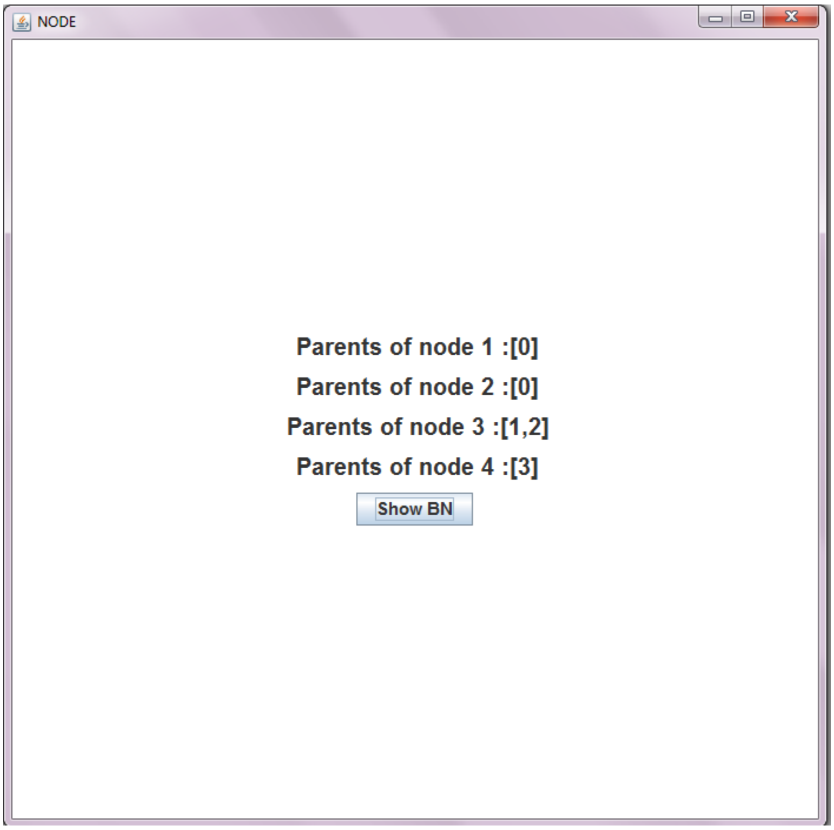

Fig 4.21: Page showing relationship between the nodes of case 2

Fig 4.22 depicts the BN structure for case 2.

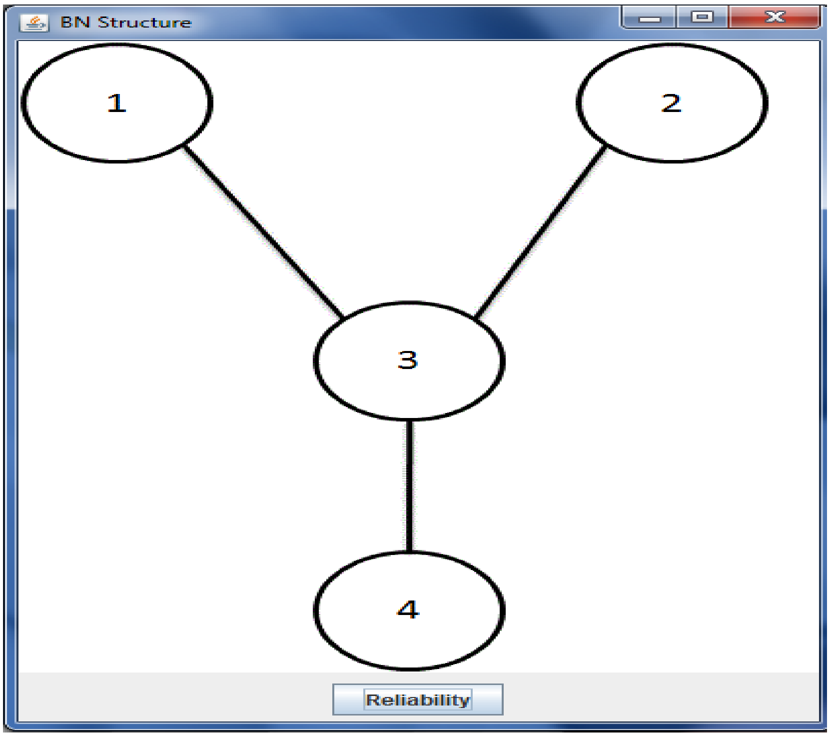

Fig 4.22: BN structure for case 2

Fig 4.23 depicts the page that depicts the reliability of the MRST. This page appears when the user clicks on Reliability button on previous page.

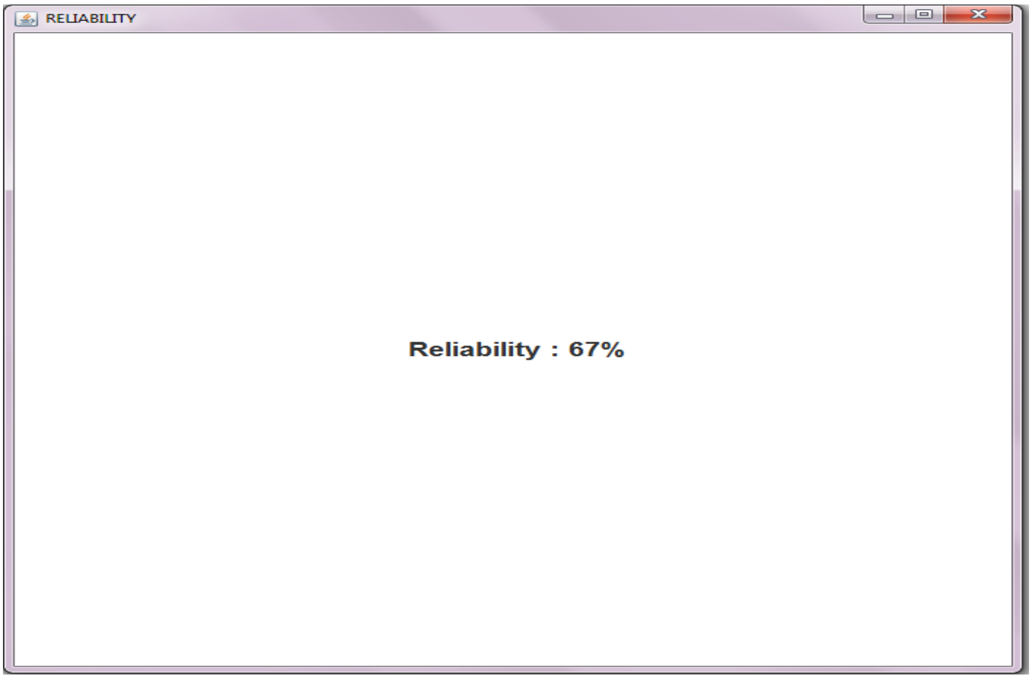

Fig 4.23: Page showing the calculated reliability of MRST of case 2

Case 3: Five Node Grid System

Fig 4.24 depicts the page of the implementation of the algorithm before the user enters the number of nodes and number of observations. This is the first page which appears when the implementation is executed.

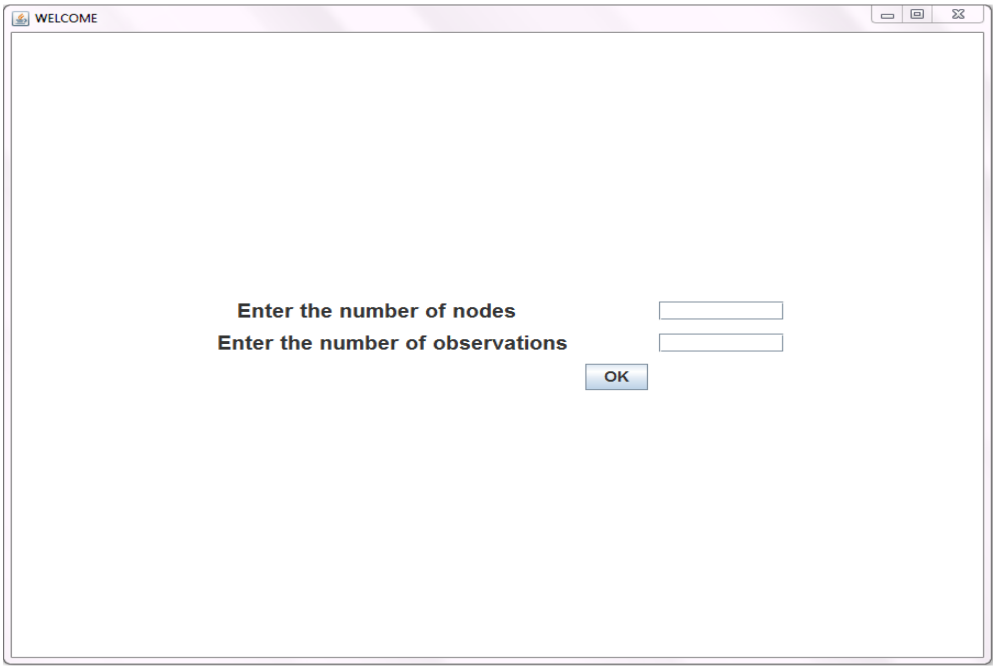

Fig 4.24: First page before entering no. of nodes and no. of observations

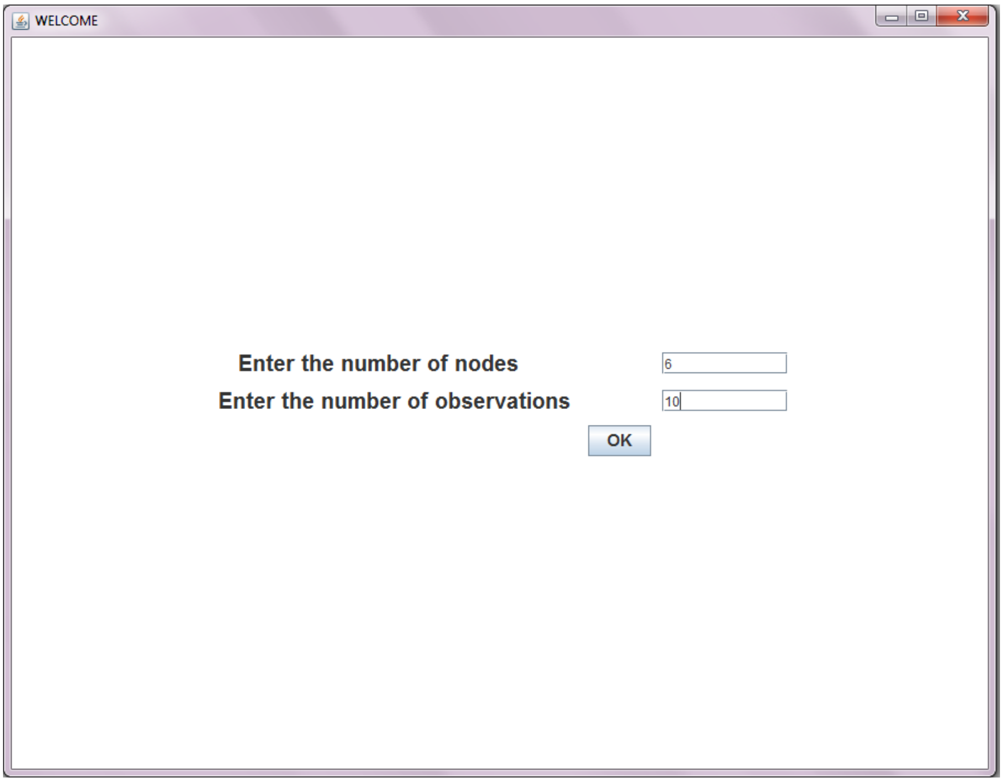

Fig 4.25: First page for case 3 after entering no. of nodes and no. of observations

When the user enters the required information i.e. the number of nodes and number of observations, the page for case 3 is as depicted in the Fig 4.25. After this the user clicks the OK button and reaches to the next page as shown in Fig 4.26.

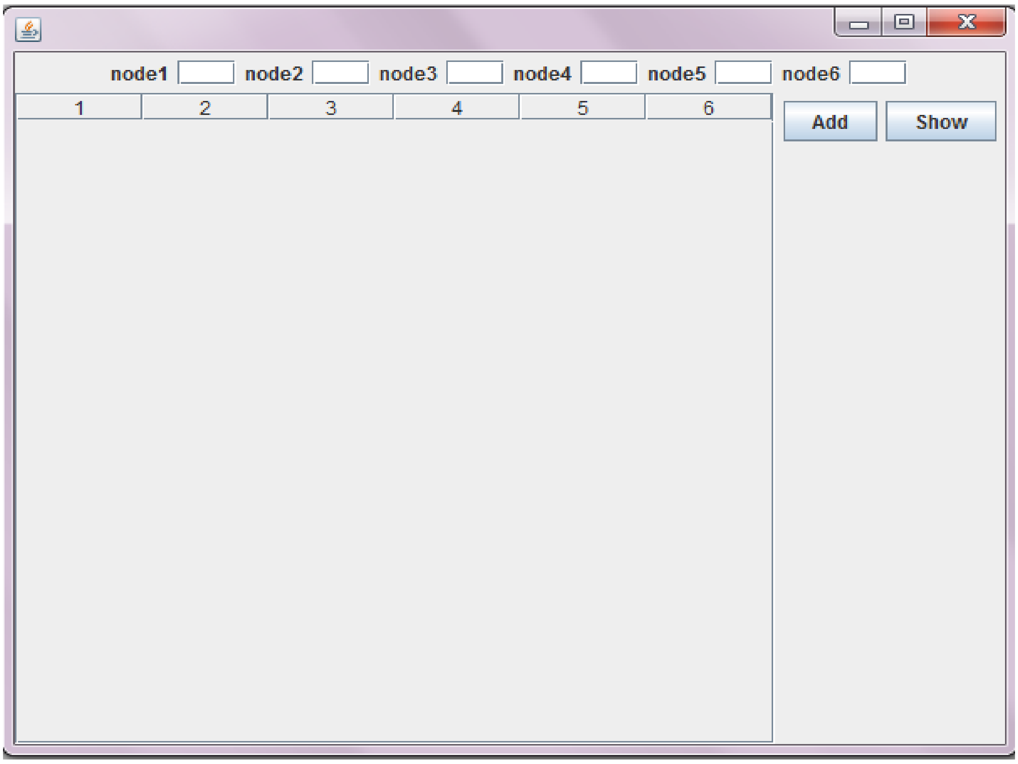

Fig 4.26: Page after clicking OK button

Fig 4.26 depicts the page after clicking the OK button in the previous page. Now the user is required to add the grid historical data observation by observation. Here the user is allowed to enter only 10 observations as mentioned in the previous page.

Fig 4.27 depicts the page when the user has entered data by entering the information in above text boxes. As the user clicks on Add button, the observation appears in the row of the table as depicted in Fig 4.27.

When the user tries to enter the data which is greater than the number of observations as entered by user on first page, the prompt box appears, as depicted in Fig 4.28, which warns the user.

After taking the input the child parent relationship between the various nodes of an MRST appear as shown in Fig 4.29. When the user clicks on Show BN button the next page appears which is depicted in Fig 4.30.

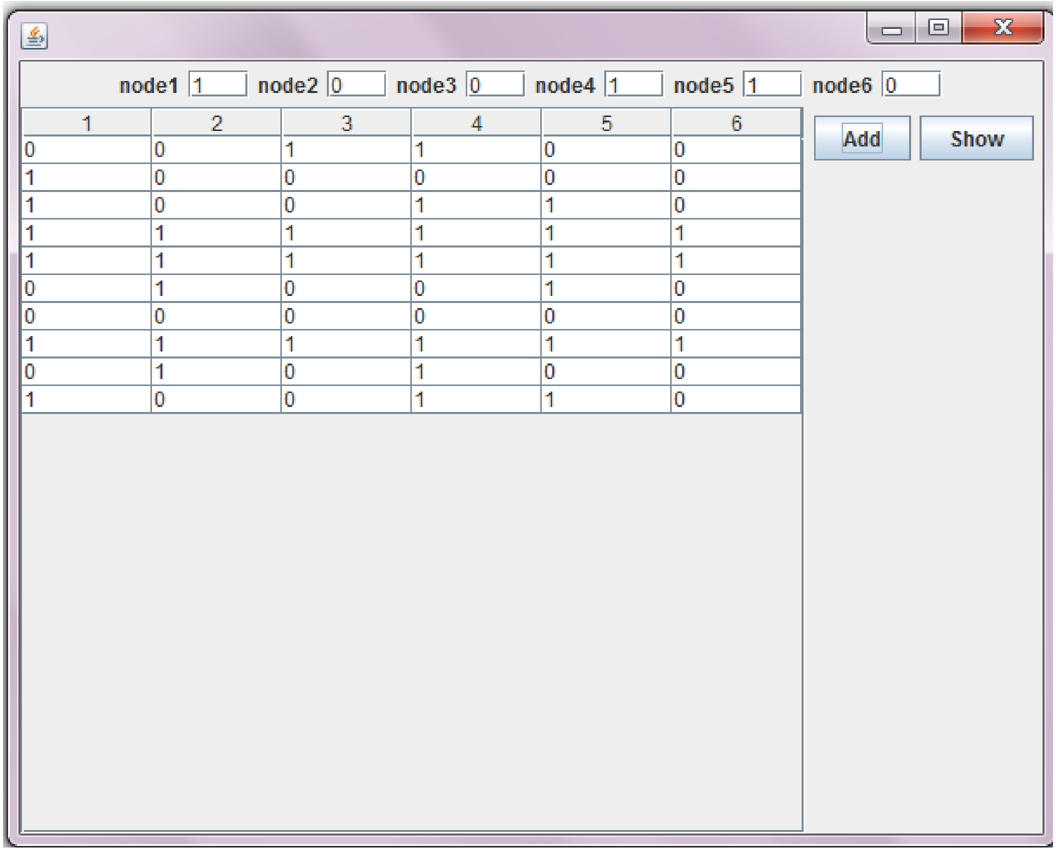

Fig 4.27: Page after entering the grid historical data for case 3

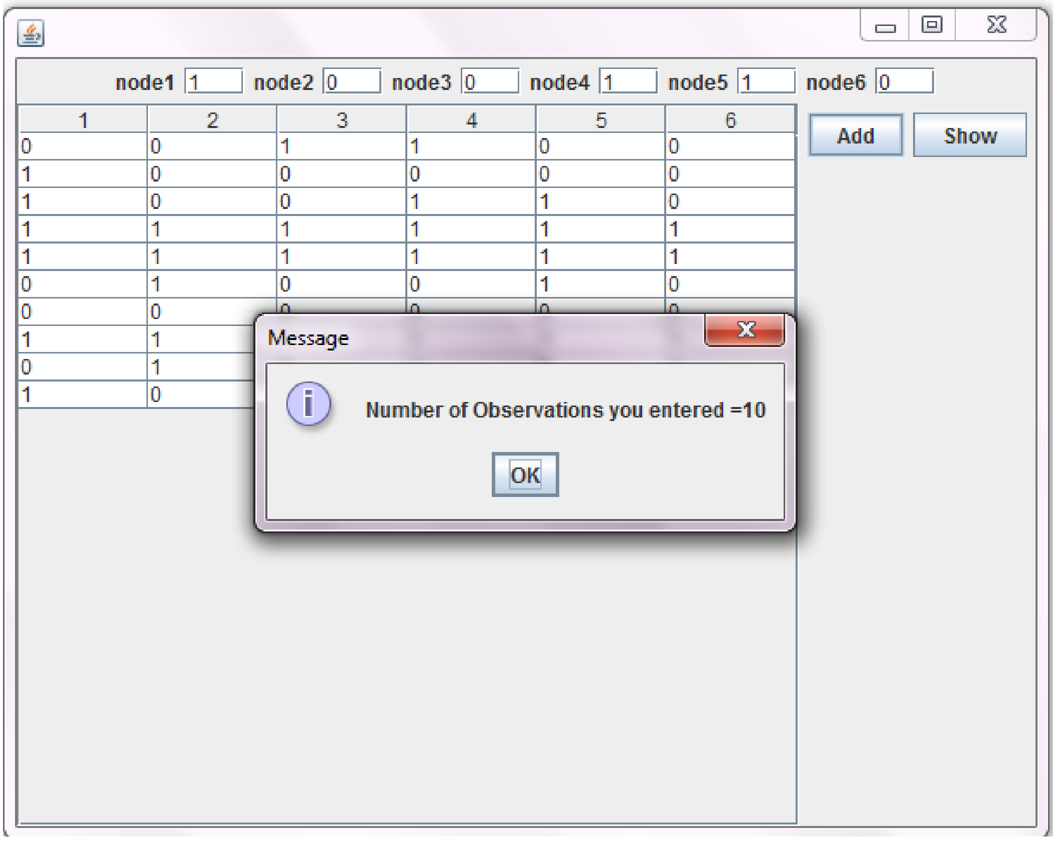

Fig 4.28: Prompt box that appears when the user tries to enter more number of observations than as mentioned earlier

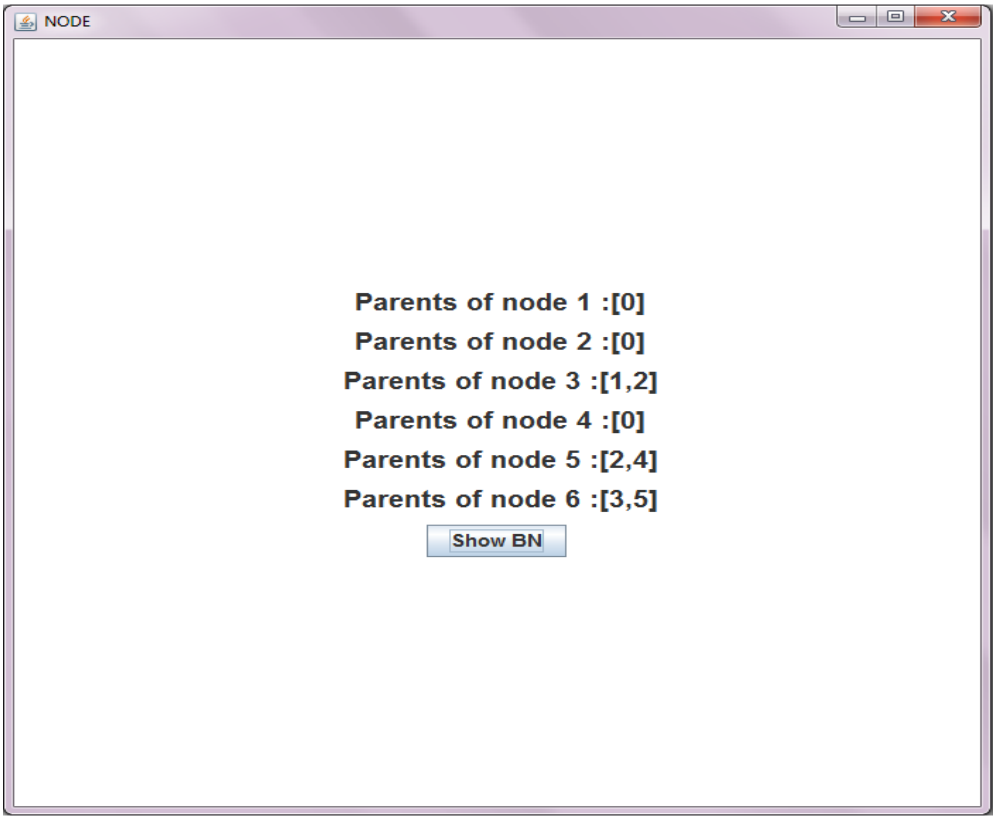

Fig 4.29: Page showing relationship between the nodes of case 3

Fig 4.30 depicts the BN structure for case 3.

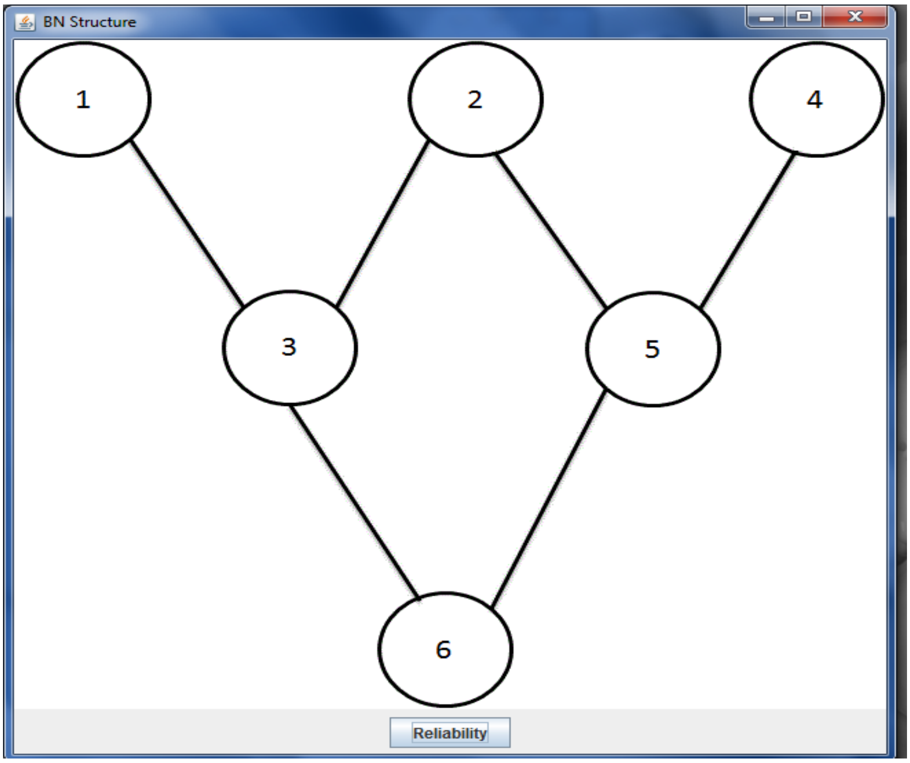

Fig 4.30: BN structure for case 3

Fig 4.31 depicts the page that depicts the reliability of the MRST. This page appears when the user clicks on Reliability button on previous page.

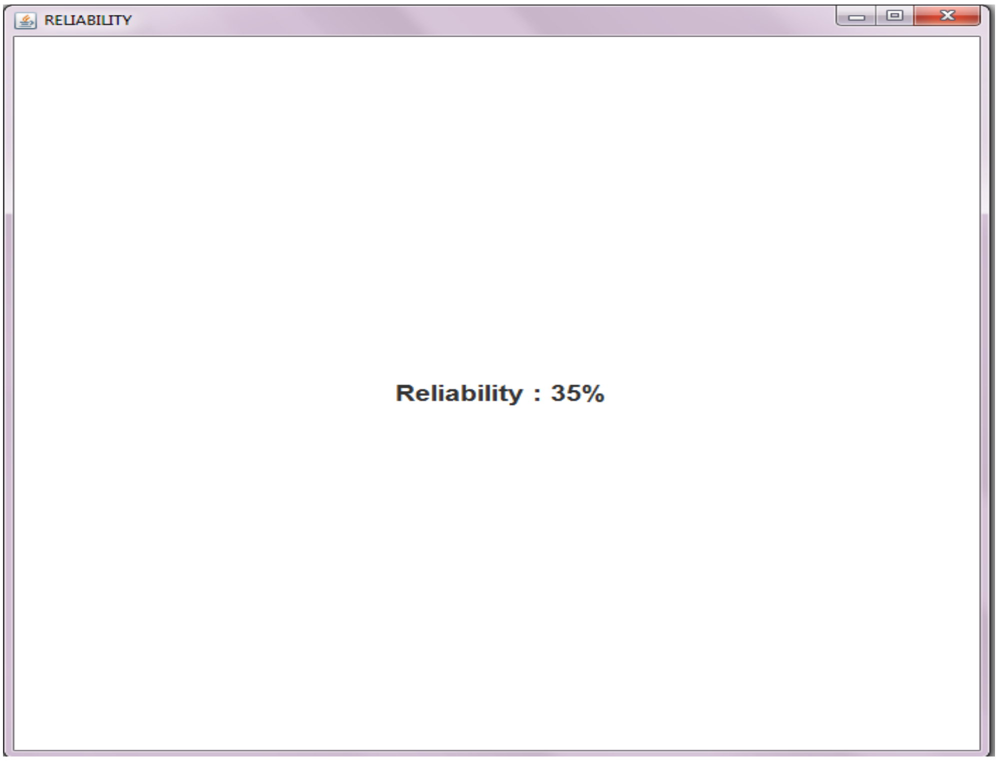

Fig 4.31: Page showing the calculated reliability of MRST of case 3

In this thesis work, the user requirement is on the basis of minimal resource spanning trees. Thus if the user is not satisfied with the reliability level, then he must consider other alternatives of his system requirements like optimizing the system on the basis of cost and reliability or both.

Chapter 5
Conclusions and Future Scope

5.1. Conclusions

The work presented in this thesis provides an insight into the world of grid computing and the current state of the art in the techniques available to estimate grid reliability.

This thesis work primarily focuses on how Bayesian Network based model can be used to estimate reliability in a distributed environment, specifically grid reliability and tries to extend the K2 algorithm. This extended K2 algorithm is implemented in Java and used to estimate the reliability of an MRST .

The MRSTs are basically constructed to estimate reliability of a network and the Bayesian Networks (constructed using K2 algorithm) to know the relationships between various nodes in a network. So this thesis uses both MRSTs and Bayesian Networks to know the associations between various nodes and to analyze grid service reliability. The success probabilities of various nodes are calculated using historical data, hence there is no need of experts.

5.2. Thesis Contributions

The main contributions of the work are:

- Analyzing the need of estimating grid reliability.
- Comparing the available reliability models for grid computing
- Assessing the probabilistic relationships and identifying probabilistic mappings between system components.
- Analyzing the need of using Bayesian Networks to estimate the grid reliability.
- Extending the K2 algorithm to estimate the grid reliability and implementing it using Java.

5.3. Future Scope

- This thesis provides implementation for estimating the reliability of an MRST only. In future this implementation can be extended so that it calculates the overall grid service reliability using the methodology discussed in this thesis. This can be done by calculating the reliability of each MRST and then combining them using the Baye's rule.

- This thesis work considers only reliability but for more practical considerations other parameters like cost, time etc. can be optimized simultaneously.

References

[1] Mark B., Buyya R., and Laforenza D., “Grids and grid technologies for wide-area distributed computing”, Software-Practice and Experience, Volume 32, No. 15, pp.1437–1466. Wiley Publishing, Hoboken, NJ, Dec 2002.

[2] Foster I., Kesselman C., Tuecke S., “The Anatomy of the Grid: Enabling scalable virtual organizations”, International Journal of High Performance Computing Applications, Volume 15, No. 2, pp.200–222, 2001.

[3] Asagba, Oghenekaro P., “Qualities of Grid Computing that can last for Ages”, Journal of Applied Sciences and Environmental Management, Volume 12, No. 4, pp.47 – 52, Dec 2008.

[4] Lai C.L. and Yang C.T., “Construct a Grid Computing Environment on Multiple Linux PC Clusters”, Tunghai Science, Volume 5, pp.107−124, Jul 2003.

[5] Amrit L.G., “Software Reliability Models: Assumptions, Limitations, and Applicability”, IEEE Transactions On Software Engineering, Volume SE-11, No. 12,pp.1411-1423, Dec 1985.

[6] Gerami M., “Some Basic Concepts of Grid Computing”, Journal Of Telecommunications, Volume 4, Issue 1, pp.14-17, Aug 2010.

[7] Foster I, Kesselman C., “The Grid: Blueprint for a Future Computing Infrastructure” Morgan Kaufmann: San Francisco, CA, 1999.

[8] Chetty M., Buyya R., “Weaving computational Grids: How analogous are they with electrical Grids?”, Journal of Computing in Science and Engineering (CiSE), Volume 4, Issue 4, pp.61-71, Jul-Aug 2002.

[9] Dabrowski C., “Reliability in Grid Computing Systems”, Concurrency and Computations: Practice and Experience, published by John Wiley & Sons, Ltd., Volume 21, Issue 8, pp.927-959, Jun 2009.

[10] Dai Y., Wang X., “Optimal resource allocation on grid systems for maximizing service reliability using a genetic algorithm”, Reliability Engineering & System Safety, Volume 91, Issue 9, pp. 1071-1082, 2006.

[11] Levitin G., Dai Y.S., “Service reliability and performance in grid system with star topology”, Reliability Engineering and System Safety, Volume 92, Issue 1, pp. 40–46, Jan 2007.

[12] Dai Y. S., Pan Y., Zou X., “A Hierarchical Modeling and Analysis for Grid Service Reliability”. IEEE Transactions on Computers, Volume 56, No. 5, pp.681-691, May 2007.

[13] Dai Y. S., Xie M., Poh K. L., & Liu G. Q., “A study of service reliability and availability for distributed systems”, Reliability Engineering and System Safety, Volume 79, Issue 1, pp.103-112, Jan 2003.

[14] Doguc O., Ramirez-Marquez J. E., “Estimating Reliability of Grid Systems using Bayesian Networks”, GCA, available at http://personal.stevens.edu/~jmarquez/Ph.D._Students_files/Ozge_White_Paper_short_3.pdf

[15] Cooper G. F., Herskovits E., "A Bayesian Method for the Induction of Probabilistic Networks from Data", Machine Learning, Volume 9, No. 4, pp.309-347, 1992.

[16] Dai, Y. S., Levitin, G., "Optimal Resource Allocation for Maximizing Performance and Reliability in Tree-Structured Grid Services", IEEE Transactions on Reliability, Volume 56, Issue 3, pp.444-453, Sept 2007.

[17] Carolina R., “Illustration of the K2 Algorithm for Learning Bayes Net Structures”, available at http://web.cs.wpi.edu/~cs539/s05/Projects/k2_algorithm.pdf.

[18] Fenton N., Krause P., Neil M., “Software Measurement: Uncertainty and Causal Modeling”, IEEE Software, Volume 19, No. 4, pp.116-122, Jul-Aug 2002.

[19] Gran B.A. and Helminen A., (2001), “A Bayesian Belief Network for Reliability Assessment”, SAFECOMP '01 Proceedings of the 20th International Conference on Computer Safety, Reliability and Security, pp.35–45, 2001.

[20] Amasaki S., Takagi Y., Mizuno O., and Kikuno T., “A Bayesian Belief Network for Assessing the Likelihood of Fault Content”, Proceedings of the 14th International Symposium on Software Reliability Engineering (ISSRE’03), pp.215-226, Dec 2003.

[21] Boudali H., and Dugan J. B., “A Continuous-Time Bayesian Network Reliability Modeling, and Analysis Framework”, IEEE Transaction on Reliability, Volume 55, Issue 1, pp.86-97, Mar 2006.

[22] Acid S., Luis M., Juan M., Rodriguez S., Rodriguez J.M., Salcedo J.L., “A comparison of learning algorithms for Bayesian networks: a case study based on data from an emergency medical service”, Artificial Intelligence in Medicine, Volume 30, Issue 3, pp.215-232, Mar 2004.

[23] Kumar V.K. P., Hariri S., Raghavendra C.S., “Distributed program reliability analysis”, IEEE Transactions on Software Engineering, Volume SE-12, pp.42–50, 1986.

[24] Ke W.J., and Wang S. D., “Reliability evaluation for distributed computing networks with imperfect nodes.” IEEE Transactions on Reliability, Volume 46, Issue 3, pp.342-349, 1997.

[25] Laskey K.B., “Sensitivity Analysis for Probability Assessments in Bayesian networks”. IEEE Transactions on Systems, Man, and Cybernetics, Volume 25, pp. 901-909, 1995.

[26] Neil M., Marquez D. and Fenton N., "Improved Reliability Modeling using Bayesian Networks and Dynamic Discretization." Reliability Engineering & System Safety, Volume 95, Issue 4, pp. 412-425, April 2010.

[27] Fenton N., Neil M., Marquez D., "Using Bayesian Networks to Predict Software Defects and Reliability", 5th International Mathematical Methods in Reliability Conference (MMR 07), Glasgow, 1-4 July 2007.

[28] Langseth H. and Portinale L., “Bayesian networks in reliability”, Reliability Engineering and System Safety, Volume 92, Issue 1, pp. 92–108, January,2007.

[29] Foster I., Zhao Y., Raicu I., Lu S., “Cloud Computing and Grid Computing 360-Degree Compared”, Grid Computing Environments Workshop (GCE '08), pp. 1-10, Nov. 2008.

[30] I. Foster. What is the Grid? A Three Point Checklist, July 2002 available at http://dlib.cs.odu.edu/WhatIsTheGrid.pdf

Paper Communicated

Ipneet Kaur, Inderveer Chana, “Estimating Grid Service Reliability Using Bayesian Networks” communicated in International Conference On Meta Computing (ICoMeC 2011), Goa, India, 15 – 16 December, 2011.