INTRODUCTION

1.1. INTRODUCTION TO MATHEMATICAL PROGRAMMING

1.1.1. An Overview

Mathematical Programming gained impotence with the advent of "Operation Research" with comprises of applications of mathematical analysis to managerial, Scientific and computational problems. The operation research or management science approach to solving a problem is based on the scientific method. Sir Francis Bacon first started it in 1960. Bacon felt that problem inquiry should consist of the following four steps:

- Observation and problem description
- Hypothesis statement
- Model development and test
- Model analysis

During the last 1930s in England and early 1940s in United States, the use of scientific methods was extensively made to analyze optimization problems. World War II challenged both countries at this time to develop optimal solution for allocation, transportation and multi-variable type problems. The modeling techniques studied by
operations researchers in 1940s and 1950s usually required algebra or calculus for solution purposes. The term “Mathematical programming” was used then, and is still used today, to describe the structuring of mathematical symbols into a model or program. Mathematical programming problems first arose in the field of Economics where allocation problems had been a subject of deep interest.

During World War II, a group of researchers sought to solve allocation type problems for the United States Air Force. One of the members of this group formulated and devised a solution procedure in 1947 for linear programming type problems. This solution procedure, called the simplex method, marked the beginning of the field of study called Mathematical Programming. Mathematical Programming is useful for finding an optimal solution to a complex problem involving many interrelated variables. It also simplifies exposition of many network problems and helps in finding optimal results which management must implement to realize its objectives.

In 1972, [TURB72], surveyed a large number of United States Corporations on their use of operation research activities. One of his conclusions was that individuals within organization were more often viewing decisions-making situations as a management science type problem requiring some types of mathematical modeling, than ever before observed in the past. Markland and Newett [MARK1972] voiced concern in their study on the misuse of mathematical programming and future problems that can be caused by the ineffective use of management science practitioners who fail to consider implementation as a part of every study undertaken in management science. Some
surveys [RADN1973, Gray1973, FABO 1976] sought specific information on the use of mathematical programming techniques. One of these surveys conducted by Fabozzi and Valaente [FABO1976] examined the use of mathematical programming methodology and where in the organization it was used. A similar study was conducted by Ledbetter and Cox [LEDB77]. They found that many organizations were using the mathematical programming techniques as reported by Fabozzi and Valaente as well as other management science methodologies. Mathematical Programming techniques have increased in number over the year. An important area of development of mathematical programming concerns with the use of computer in management science. More and more computer software are being developed every day making the computational aspects of mathematical programming less complex. Time-sharing and interactive computer systems are also influencing a de-emphasis on computational experience and more on problem formulation. During the last decade, mathematical programming approaches have gained immense importance for solving task allocation and load balancing problems in Distributed Processing Environment (DPE) [CHUL980, MA1982, SINCl1987, and REID1994].

1.1.2. Mathematical Programming Problems

During the last decade, mathematical programming techniques have been recognize as the most powerful methods for modelling and analyzing several kind of problems. Many research problems are formulated in the form of mathematical models, which describe the quantitative features of all types of problems. Mathematical programming techniques are used for the formulation and solution of research problems
by systematic planning of various activities. The basic problem in mathematical programming is to find the unknown values of some variables, which will optimize the value of the objective function subject to a set of constraints. Most of the mathematical programming problems can be formulated in the following general form [KWAK 1987]:

\[
\text{Maximize (or Minimize)} \quad Z = \sum_{j=1}^{n} c_j x_j \\
\text{subject to } \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (\text{for } i = 1, 2, \ldots, m) \\
\text{and } x_j \geq 0 \quad (\text{for } i = 1, 2, \ldots, n)
\]

Where,

\[
Z = \text{value of the objective function which measures the effectiveness of the decision choice,}
\]

\[
x_j = \text{unknown variables that are subject to the control of the decision maker,}
\]

\[
c_j = \text{unit profit contribution of an output or cost of an input which is known,}
\]

\[
a_{ij} = \text{production (or technical) coefficients that are known, and}
\]

\[
b_i = \text{available resources in limited supply.}
\]

Production of goods, in any organization, requires productive resources, which are in short supply in the real world, and they are, therefore, restricted resources. These restricted resources are expressed as equalities or inequalities in mathematical programming models.
1.2. **INTRODUCTION TO COMMUNICATION**

1.2.1. **An Overview**

The reasons for studying data communications can be summed up in the occupational history of the United States. In the 1800s they were an agricultural society dominated by farmers. By the 1900s they had moved into an industrial society dominated by labor and management. Now, as they approach the twenty-first century, they clearly have moved into the information society, which do computers, data communications, and highly skilled individuals who use brainpower instead of physical power dominate. The industrial society has reached its zenith, and the communication/computer era, started in mid 1950s, which is dubbed the information society, is advancing rapidly.

In an information society dominated by computers and communications, value is increased by knowledge as well as by the speed of movement of that knowledge. This new information economy will completely destroy Ricardo's labor theory of value, because, in such a society, what increases value is not the labor of individuals, but information. The main stream of the information age is a communication network. The value of a high-speed data communication network that transmits knowledge/information is that it brings the message sender and the message receiver closer together in time. For example, in the 1800s it might have taken several weeks for specific information to reach the United States from England. By the 1900s it could be transmitted within an hour. Today, with modern data communication systems, it can be transmitted within seconds. Finally, the transition from an industrial to an information society means that we have to
learn many new technologically based skills. The study of data communications has become a basic tool that can be used throughout our lifetime. We incorporate our knowledge of data communications into several careers such as circuit designer, programmer, business system application developer, communication specialist, and business manager.

1.2.2. Evolution

Today we take data communications for granted, but it was early pioneers like Samuel Morse, Alexander Graham Bell, and Thomas Edison who developed the basic electrical and electronic systems that ultimately became today's voice and data communication networks. When the telephone arrived, it became the accepted communication device that everyone wanted. Several technological enhancements were made in telephonic technology from 1837 to 1951 and it was in 1951 that the first direct long distance customer dialing began. The first international satellite telephone call was sent over the Telstar satellite in 1962. In 1963, touch-tone telephones began to be marketed. Their push buttons were easier to use than rotary dials, and they became quite popular. By 1965, there was widespread introduction of commercial international telephone service by satellite. Picturephone service, which allows users to see as well as talk with one another, began operating in 1969. All through the 1970s there were many arguments and court cases regarding the monopolistic position that A T & T held over other companies that wanted to offer communication services. The litigation led to the litigation led to the divestiture of AT&T on January 1, 1984. During 1983-84 the newer cellular telephone networks supplanted traditional radio telephone-type calls. Integrated
Services Digital Networks began serving the public in 1986. These networks allow the simultaneous transmission of voice, data, and video images [FITZ1988]. By 1987 there was considerable competition in both the voice and data communication markets as a number of independent companies began to sell communication services in a manner similar to that of automobile marketing. And now we have smaller and less expensive portable telephones to carry around everywhere.

India is a resource rich fast developing country where telecommunication has passed from the stage of convenience to essentiality. Computers have recently entered in a big way in Indian society. As on March 31 1986, it was estimated that the country had about 3050 super, mainframes and minis and 7000 micros and personal computers. The number of microcomputers and personal computers is growing fast. Today it has crossed the limit of one million. It was thus thought to have the computer facilities. The idea of computer networking has been widely accepted in government organizations. The NIC has been doing this for planning purposes to Govt. of India. NICNET computer and communication facilities have provided nation-wide links to make the computing resources available at the places from where the information emanates [AGAR1995]. The Indian society has now clearly realized the importance of communications, which may be in its various forms including telephones, memorandums, telex, mail, reports etc. Attempts have been made by the organizations like telecommunications research center to produce prototype model of several important communication systems. Electronic mail, facsimile equipment, modems, teletext, voice mail etc. have been developed, integration of computer communication with special digital switches, ISDN, has also
been tried. This has put India in the forefront of telecommunication services. The Govt. of India has realized that without a better and secures telecom system the fruits of development can never be fairly distributed, be it in education, economics, rural development or international trade and banking. The development of efficient communication systems has opened new vistas in electronic industries. The OSI technology is coming up fast and Fiber Optics technology has also entered into the scene. Optical fiber computer communication networks [Fibre LANs] are now being developed in India.

### 1.3. DISTRIBUTED SYSTEM

Distributed processing plays an important role in large data base installations where processing load is distributed for organizational efficiencies. Banking system, travel agency systems, and power control systems are few examples of distributed processing environment. The application that exhibits parallelism, involving enormous repetitive processing, or requiring extremely fast processing in a real-time environment demand distributed processing systems. To name a few such type of application are signal processing, meteorology, image analysis, cryptography, nuclear reactor control, sonar & radar surveillance, simulation of VLSI circuits, and industrial process monitoring. The on-set of the microprocessor technology has made the Distributed Processing System [DPS] economically viable and attractive for many applications of computer. However, many problem areas in DPS are still in their primitive development stages. DPS are increasingly drawing attention, yet have a meaning that is not understood. The term "DPS" is used to describe whenever there are
several computers interconnected in some fashion so that a program or procedure running system terms with multiple processors. However, the term has different meanings to different systems because processors can be interconnected in many ways for various reasons. In the most general form, the word distribution implies that the processors are in geographically separate locations. Occasionally, the term is also applied to an operation using multiple mini-computers, which are not hardware, connected with each other and are connected through satellite. A user-oriented definition [BHUT1994, SITA1995] of distributed computing is "Multiple Computers, utilized cooperatively to solve problems". While addressing task allocation issues in a DPS, the three important aspects need consideration:

(i) **Multiplicity of resources:**

There are a number of resources, in particular, processors. Homogeneity of physical resources is not essential. A system may have processors with identical characteristics and capabilities.

(ii) **Dispersion:**

The resources in the system are physically or logically distributed. AU the processors are independent and tied together by communication links. The communication links provide means for transferring information between the processors.
(iii) **Control:**

All the processors in the system are autonomous and there is no master and slave relation among the processors. For the user, the collection of processors should be invisible, the multiple processor system should appear as virtual uniprocessor, and users need not know on which machine their programs are running and where their files are stored.

(iv) **Transparency:**

All the processors in the system are autonomous and there is no master-slave relationship among them. The fact there are several co-operating processors in the system must be invisible (transparent) to the user and a single image id presented to the user. The system should appear as a uniprocessor system and users need not know on which machine their programs are executing and where the data or files stored.

(v) **Flexibility:**

Flexibility is an important aspect of a distributed system. It means that should be easy to incorporate changes in a user transparent manner or with minimum interruption to the user. Further, in every system, new functionalities are to be added from time to time to make it more powerful and easy to use. Therefore, It should be easy to add new services to the system.
(vi) **Performance:**

The performance of the distributed system must be at least as good as a centralized system. That is when a particular application is run on a distributed system, its overall performance should be better than or at least equal to that of running the same application on a single processors system.

(vii) **Scalability:**

Scalability refers to the capability of a system to adapt to increased service load. A distributed system must be able to cope with the growth of nodes and users in the system.

(viii) **Security:**

Security must be provided in the system to prevent destruction and unauthorized access so that users can trust on the system and rely on it.

1.3.1. **Distributed Real Time Systems**

The on-set of the "Microprocessor Technology" has made the Distributed Real-Time Processing System (DTRS) economically viable and attractive for many applications of computer. However, many problem areas in Distributed Real-Time Processing System are still in their primitive development stages. Distributed Real-Time Processing System are increasingly drawing attention, yet have a meaning that is not understood. The term "Distributed Real-Time Processing System" is described whenever several computers interconnected in some fashion such that a program or procedure
utilizes this distributed but combined power and gets executed in real time. The term has different meanings with regard to different systems, because processors can be interconnected in many ways for various reasons. In its most general form, the word distribution implies that the processors are fixed in geographically separated locations. Occasionally, the term is also applied to an operating environment using multiple mini-computers not connected with each other with the help of physical communication lines but are connected through satellite.

1.3.2. Machine Size Vs Instruction Execution

Distributed system provides much faster execution by facilitating parallel execution of tasks. A major driving force towards distributed processing is the cost of small processors. Until the spread of mini computers in the early 1970s, a commonly accepted rule was Grosch's Law, which said, "The cost per machine instruction executed is inversely proportional to the square of the size of the machine ". Grosch's Law became questionable in the 1970s. Even some people suggested that it had been reversed because the cost per instruction on some mini-computers was lower than on large computers and on microprocessors was lower than mini-computers. The reason is related to the use of Very Large Scale Integrated circuits, which can be mass-produced economically. Their development cycle is much shorter than that of large machines.

1.3.3. The Logical Vs Physical Design

The implementation of a distributed system depends both on logical and physical premises. The logical functions center on the procedural design for channeling the flow
of data and controlling the physical facility throughput of the projected system configuration. The object of the physical functions is the design of the hardware and hard-software [firmware] devices to provide a specific level of capability. Functions will be distributed both to machines in the computer center and to many machines at user locations. This trend of the distribution of processing is continuing because software usage of machine instructions per second is growing much faster than the development of higher speed machines. The distributed system is motivated by the need for cost reduction in tasks executing. The chronological order of development of the distributed system requiring the logical and the physical design phase is depicted in following figure:
1.3.4. Distributed Vs Parallel Computing

In some computer networks, the control mechanism is mostly centralized. In others, they are mostly distributed. Where purely centralized control exists, loss of the center puts the entire network out of action. With the distributed control any portion of network can be destroyed and the rest will continue to function. Although computation speed has increased several folds over the past three decades of computing, the user demand for faster speeds is growing at a much faster rate. One of the approaches to meet the growing demand of faster computation is to use parallel processing. Parallel computers, which emphasize parallel processing, may be employed. Parallel computers are available with different architectures [FORT1985, STON1987, BOKH1988] and divided into three classes: array computers, pipeline computers and multiprocessor systems. To name a few are IBM 3081, Denelcor HEP, Cray x-MP and PARAM systems. All the parallel computers described above are centralized computing items and all hardware and software resources are housed at the same place.

1.3.5. Types of Distributed Processing Systems

1.3.5.1. Horizontal Vs Vertical Distribution

By Vertical distribution we mean that there is a hierarchy of processors, shown in the following figure [MART 1988]. The transaction may enter and leave the computer system at the lowest level. The lowest level may be able to process the transaction or may execute certain functions and pass it up to the next level. Some, or all, transactions eventually reach the highest level, which will probably have access to on-line files or...
databases. The computers at the lowest level can be networked together, if data sharing is required.

![Diagram of horizontal distribution]

**Horizontal Distributions**

A horizontal distribution implies that the distributed processors do not differ in rank. They are of equal status-peers and referred as peer-coupled systems. A transaction may use only one processor, although there are multiple processors available. On some peer-coupled systems a transaction may pass from one system to another causing different set of files to be updated as depicted in the following figure [MART 1988].

![Diagram of horizontal distribution]

**Horizontal Distributions**
1.3.5.2. Functional Distribution Vs System Distribution

In some distributed systems, usually vertical systems, functions are distributed, but not the capability to fully process entire transactions. The lower-level machines may be intelligent terminals or intelligent controllers in which processors are used for functions such as message editing, screen formatting, data collection dialogue with terminal operators, security, or message compaction concentration. They do not complete the processing of entire transactions. This distribution is referred as functional distribution and is contrasted with system distribution in which the lower-level machines are systems their own right, processing their own transactions, and occasionally passing transactions or data up the hierarchy to higher level machines. In a system distribution environment the lower machines may be entirely different from, and incompatible with, the higher machines. In a function distribution environment, close cooperation between the lower-level and higher-level machines is vital. Overall system standards are necessary to govern what functions are distributed and exactly how the lower and higher machines form part common system architecture with appropriately integrated control mechanisms and software.

1.3.6. Features of Distributed System

There are a variety of reasons for building of any distributed systems, some of them are as follows:

(i) Computational Speedup:

Computational speedup can be achieved if a particular computation can be partitioned into a number of sub-computations that can run concurrently. A distributed
system may allow user to distribute the computation among the various sites – to run that computation concurrently. In addition, if a particular site is currently overloaded with the jobs, some of them may be moved to other, lightly loaded, sites.

(ii) **Fault Tolerance:**

One of the real attractions of distributed processing is the resulting high fault tolerance. If a communication link or a site fails in a distributed system, the remaining sites can potentially continue operating. However, this may reduce the overall throughput of the system.

(iii) **Increased Throughput:**

The throughput of the system is expected to increase by distributing the total workload to the various service stations. As in a distributed system there are a number of processing elements, one would hope to get more work done in shorter period of time.

(iv) **Communication:**

In the distributed system it is quite often the programs running at different sites need to exchange data with one another. When a number of site are connected to one another by a communication network, the processes at different sites have the opportunity to exchange the information.

(v) **Resource Sharing:**

If a number of sites are connected to one another, then a user at one site may be able to use the resources available at another. In general, resource sharing in a distributed
system provides mechanisms for sharing files at remote sites, processing information in a
distributed database, printing files at remote sites, utilizing specialized hardware devices,
and performing other operations. Distributed processing plays an important role in large
data base installations where processing load is distributed for organizational efficiencies.
Banking system, travel agency systems, and power control systems are few examples of
distributed processing environment. The application that exhibits parallelism, involving
enormous repetitive processing, and/or requiring extremely fast processing in a real-time
environment demand distributed systems. To name a few such type of application are
signal processing, meteorology, image analysis, cryptography, nuclear reactor control,
sonar & radar surveillance, simulation of VLSI circuits, and industrial process
monitoring.

1.4 TASK ALLOCATION

The task allocation in a distributed system finds extensive applications in the
faculties where large amount of data is to be processed in relatively short period of time,
or where real-time computations are required. Meteorology, Cryptography, Image
Analysis, Signal Processing, Solar and Radar Surveillance, Simulation of, VLSI circuits
and Industrial process monitoring are areas of such applications. These applications
require not only very fast computation speeds but also different strategies involving
distributed task allocation systems. In such applications the quality of the output is
proportional to the amount of real-time computations. To meet such challenging
computing requirements at electrifying speeds some efficient task allocation strategies are
required for proper utilization of distributed system under the constraints of memory
capacity available at each processor and time constraints in executing task. The advent of VLSI technology resulting in low cost microprocessor has made distributed system an economic reality in today's computing environment. The modularity, flexibility and reliability of distributed system make it attractive to much type of users, and several distributed systems have been designed and implemented in recent years. The first step in the distributed software engineering is to partition application program into a set of smaller independent tasks and allocates them to different processors. To enable the increasing variety of computers to communicate with one another, there must be rigorously defined protocols as to (a) how the control message and data message are exchanged in the distributed system and (b) how to control the communication process along with the protocol definition.

The format of the control messages, the headers and the trailers of data messages are likewise rigorously defined. Protocol becomes quite complex, as it is desirable that there should be a widely accepted standard so that all types of machines can inter communicate. However, many problematic areas in the distributed systems are still in their Primitive development stage. Some major problems that present the widespread use distributed systems are: (a) the degradation in system throughput caused by the saturation effect, (b) the difficulty in evenly utilizing each processor in the distributed systems, (c) a large gap between the engineering application requirements and existing distributed network architecture and (d) the difficulty in verifying task allocation resulted from any allocating model.
Distributed systems have been so complex that intuition alone is not sufficient to predict their performance. Therefore, mathematical modeling plays an important role for predicting the performance of the distributed systems. Mathematical models of system performance range from relatively simple ones, whose solution can be obtained analytically, to the complex ones, which require simulation. Assigning tasks to processors is called task allocation, which involves the allocation of tasks to processors in such a way that some effectiveness measures are optimized. If the effectiveness measure can be represented as a linear function of several variables subjected to a number of linear constraints involving these variables, then the task allocation is classified as a Linear Programming Problem. Likewise, for the processor, which can perform anyone the several tasks, possibly the difference of execution, and the effectiveness measure is the total execution cost to perform all tasks when one and only one task is allocated to each processor. In such cases, task allocation is classified as an assignment problem. Assigning \( m \) tasks to \( n \) processors, through exhaustive enumeration, results in \( n^m \) possible ways. A general structure of task allocation in a distributed system is shown in following figure:
General Structure of Tasks Allocation

Shatz and Wang [SHAT 1987] studied that the problem of choosing an optimal allocation from all assignments is exponentially complex. An efficient task Allocation policy should avoid excessive Inter Processor Communication [IPC] and exploit the specific efficiencies of the processors and in case of a system having similar processor, the tasks or modules should be distributed as evenly as possible. The bottleneck in IPC is to provide linear speed up solutions with the increase in number of processors as
suggested by Chu et al and Lint et al. [CHU 1980, LINT 1981]. The strategies of task allocation on a parallel and distributed system may be done in any of the following ways:

(i) **Static Allocation:**

In static allocation when a task is assigned to processor, it remains there while the characteristic of the computation change, a new assignment must be computed. The phrase "characteristics of the computation" means the ratios of the times that a program spends in different parts of the program. Thus in a static allocation, one is interested in finding the assignment pattern that holds for the life time of a program, and result in the optimum value of the measure of effectiveness.

(ii) **Dynamic Allocation:**

In order to make the best use of resources in a distributed system, it is essential to reassign modules or tasks dynamically during program execution, so as to the advantage of changes in the local reference patterns of the program [MANI 1998]. Although the dynamic allocation has potential performance advantages, Static allocation is easier to realize and less complex to operate.

1.4.1 **Task Allocation Problem**

Consider a set of “n” processors \( P = \{p_1, p_2, p_3, \ldots, p_n\} \), interconnected by communication links and a set of “m” executable tasks \( T = \{t_1, t_2, t_3, \ldots, t_m\} \). The allocation of each “m” tasks to “n” available processors such that an objective cost function is minimized subject to the certain resource limitations and constraints imposed
by the application or environment. An assignment of tasks to processors is defined by a function \( f \), from the set of tasks \( T \) to the set of processors \( P \), \( f: T \to P \). Then the total cost of an allocation \( x \) can be expressed as follows:

\[
\text{Cost}(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[ e_{ij} x_{ij} + \sum_{k \neq j} \sum_{l \neq j} c_{ik} d_{jl} x_{ij} x_{kl} \right]
\]

where,

\[
x_{ij} = \begin{cases} 
1, & \text{if } i^{th} \text{ task is assigned to } j^{th} \text{ processor} \\
0, & \text{otherwise}
\end{cases}
\]

\( e_{ij} = \) cost of executing task \( t_i \) on processor \( p_j \)

\( c_{ik} = \) Communication between tasks \( t_i \) and \( t_k \)

\( d_{jl} = \) distance between processors \( p_j \) and \( p_l \)
1.5. SCHEDULING POLICIES

The scheduling policies in distributed systems have two categories such as job scheduling and task scheduling. Job scheduling allocates independent jobs to different processors to optimize system performance. Task allocation scheduling requires assignment of multiple interdependent tasks or modules of a single allocation program to minimize job completion time. The decision of scheduling polices is required at several levels within distributed system as shown in figure mentioned below.

Taxonomy of Scheduling
At higher level the decision about local or global scheduling might be desired to make. The local scheduling policies decide about the allocation of a task or a job to the time slots for a single processor [BUCK1979, CASA1988]. The global scheduling relates to the problem of deciding where a job or a task [ROTI1994] runs. In the next lower level is the choice between static and dynamic in a global scheduling. Static scheduling is prior assignments tasks to the processors where the allocation does not change during the lifetime of tasks. While in dynamic scheduling the allocation decision is made during execution of tasks. Further there are two type of dynamic scheduling: decentralized and centralized. If there is a single decision point, the dynamic scheduling is known as centralized and if the decision-making is spread throughout the system it is known as decentralized.

1.6. RELIABILITY

The reliability theory has grown into an engineering science in its own right. Much of the initial theory, engineering, and management techniques centered about hardware, however, human and procedural elements of a system were often included. Since its beginning, following World War II, and also then in the late 1960s the term software reliability has become popular, and now reliability theory refers to both software and hardware reliability [SHOO 1983, AMAR 1998, DENS 1998]. Human reliability analysis is a method by which the probability of a system required human action, task, or job will be completed successfully within the required time period and that no extraneous human actions detrimental to system performance will be performed. Results of human reliability analysis’s are often used as inputs to probabilistic risk assessments, which
analyze the reliability of entire systems by decomposing the system into its constituent components, including hardware, software, and human operators. Reliability allocation is the process of specifying a level of reliability for each subsystem or module in a system so as to achieve a system reliability objective. Reliability optimization has attracted many researchers since 1960 due to reliability critical importance in various kinds of systems. To maximize system reliability, the following options can be considered:

- Enhancement of component reliability,
- Provision of redundant components in parallel, and,
- Reassignment of interchangeable components.

The diversity of system structures, resource constraints, and options for reliability improvement has led to the construction and analysis of several optimization models.

1.7. ORGANIZATION OF THE THESIS

The First chapter of this thesis entitled “APPLICATIONS OF MATHEMATICAL PROGRAMMING IN THE DEVELOPMENT OF SOME MODELS FOR PERFORMANCE EVALUATION OF DISTRIBUTED SYSTEMS” is devoted to the introductory background of the topic in Mathematical programming, communication, distributed systems, task allocation and other issues directly related to the work. A brief overview of the communication and distributed system has been given to understand the basics of it. Task allocation problem in these systems, is discussed and also some of the related concept has been included. Finally, the organization of the thesis,
include the chapter wise brief summary of the present work, has been mentioned in this chapter.

Task assignment for any distributed system is a most interesting and demandable research problem. Various methodologies and techniques are available in the literature to solve such problems. As it is the primary for researchers to know about related methodology and techniques. The Second Chapter of this thesis is a collection of all such research work, which is available in the literature, and directly-indirectly correlates to our work. We have considered the Decomposition approach, Genetic algorithmic approach, Integer Programming approaches, Dynamic Programming, Heuristic approaches, Network flow & Partitioning algorithm, Problem Reduction method, Petri Net modeling & Reliability evaluation and Shortest Path algorithmic method. And also precedence constraints, particular architecture consideration, reliability evaluation has mentioned.

The Third Chapter of the thesis is the important and main, as it include all five research models. The general motivation for the present research has been written briefly. The Objective, Technique, Algorithm, Implementation and Conclusion of each model are mentioned separately. The details of each model is as mentioned below:

The title of the Model – I is the ‘reliability based algorithm for performance enhancement of the distributed systems’. The model is developed with the objective of maximizing the overall processing reliability. Here we have considered the processing reliability and communication amongst the tasks. The results shows that the optimal assignments with optimal reliability. The present model has also compared with Richard
et al [RICH 1982] and found that the developed model is better than the existing one at front of time complexity. Results are shown graphically as well as in tabular form. This study is capable to deal all such real life situations, where the tasks are more than the number of processors. The developed algorithm is programmed and several sets of data have been tested to verify the effectiveness of the algorithm.

The title of the Model – II is the ‘task allocation through reliability index for distributed system’. In this model a set of m tasks are assigned to a set of n processor. Inter task communication cost of the task have been considered along with the processor reliability and processing cost. Total processing reliability and cost have been obtained and then reliability function is defined. Model has also been compared with Zahedi et al [ZAHE 19991] and Kumar et al [KUMA 1999]. Results are shown in tabular form and also graphically. The developed algorithm is programmed and several sets of data have been tested to verify the effectiveness of the algorithm.

The title of the Model – III is ‘an efficient algorithm for tasks allocation to the distributed systems’, with the objective to obtain the optimal assignments in such a way that the execution cost, execution reliability, communication cost and communication reliability has to be optimized. A technique has been devised to fulfill the said objective. The results are shown graphically and also mentioned in the tabular form. To test the performance of the algorithm this model is compared with the existing ones and found that it is a good approach. This study is capable to deal all such real life situations, where the tasks are more than the number of processors. The developed algorithm is
programmed in C++ and several sets of data have been tested to verify the effectiveness of the algorithm.

The title of the Model – IV is the ‘communication reduction based algorithm for optimal assignment in distributed systems’. Here we have suggested a way to assigning tasks which have maximum communication. The developed algorithm provides the optimal cost of assignments. The developed model has been compared with Yadav et al [YADA1995] algorithm. This study is capable to deal all such real life situations, where the tasks are more than the number of processors. The developed algorithm is programmed and several sets of data have been tested to verify the effectiveness of the algorithm.

The title of the Model – V is the ‘optimizing the reliability of distributed systems’, it defines the executing reliability and communication reliability function to obtain the optimal reliability of distributed system. Here we have obtained the processor wise execution reliability. The communication reliability and total reliability of the distributed system has mentioned in the conclusion part of the model. This study is capable to deal all such real life situations, where the tasks are more than the number of processors. The developed algorithm is programmed and several sets of data have been tested to verify the effectiveness of the algorithm.

The future scope of the research has been given in the chapter four of this thesis. During the period of this research work some of the research papers have been presented in seminar and conferences and some of research papers have been sent for the
publication in journals. A combined list of such papers has been mentioned in this chapter. During this research work we have gone through large number of research papers, books, monographs, Ph. D. thesis etc. so that in the last of the thesis a list in alphabetic order of such research material has been attached.