INTRODUCTION

1.1. INTRODUCTION TO OPTIMIZATION TECHNIQUES

1.1.1. An Overview

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. Mathematical Programming is one of the most powerful tools for optimization of communication parameters and it is still used today by the researchers, to describe the structuring of mathematical symbols into a model or program. These 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. It 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, [TURB 1972], 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 modelling, than ever before observed in the past.

Markland and Newett [MARK 1972] 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 [RADN 1973, GRAY 1973, FABO 1976] sought specific information on the use of mathematical programming techniques. One of these surveys conducted by Fabozzi and Valaente [FABO 1976] examined the use of mathematical programming methodology and where in the organization it was used.

A similar study was conducted by Ledbetter and Cox [LEDB 1977]. They found that many organizations were using the mathematical programming techniques as reported by Fabozzi and Valaente as well as other management science methodologies. An important area of development of mathematical programming concerns with the use of computer in operation research. 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 communication systems [CHU

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]:

$$Maximize \ (or \ Minimize) \ Z = \sum_{j=1}^{n} c_j x_j$$

subject to $\sum_{j=1}^{n} a_{ij} x_j \ <= \ b_i \ (for \ i = 1, 2, \ldots, m)$

and $x_j \ geq 0 \ (forj = 1, 2, \ldots, n)$
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. The decision maker's goal is to find the values of the decision variables within limits and to optimize the value of the objective function.

1.2. INTRODUCTION TO DATA COMMUNICATIONS

1.2.1. The Need of Study

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.

Picture phone 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 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 [ISDN] began serving
the public in 1986. These networks allow the simultaneous transmission of voice, data, and video images [FITZ 1988].

During the 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.

1.2.3. The Indian Scenario

India is a resource rich fast developing country where telecommunication has passed from the stage of convenience to essentiality. Computers have 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. The idea of computer networking has been widely accepted in government organizations. The National Informatics Center [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 [AGAR 1995].

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 [TRC] 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 secure 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. To meet the telematics challenge in the country, the IITs and some university departments have begun setting up facilities to train the scientists and the technicians, engaged in futuristic research and develop new systems in communication technology.

1.3. COMMUNICATION SYSTEM

1.3.1. Meaning of Communication System

The on-set of the microprocessor technology has made the Communication System [CS] economically viable and attractive for many applications of computer. However, many problem areas in communication system are still in their primitive development stages. CS is increasingly drawing attention, yet has a meaning that is not understood.
In a CS 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. A user-oriented definition [BHUT 1994, SITA 1995] of distributed computing is "Multiple Computers, utilized cooperatively to solve problems".

1.3.2. Main Aspects of Communication Systems

While addressing task allocation issues in a CS, the following are important aspects need to be 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 centralize system. That is when a particular application is run on 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 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 user can trust on the system and rely on it.

1.3.3. **Machine Size Vs Instruction Execution**

Communication Systems 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 [VLSI] circuits, which can be mass-produced economically. Their development cycle is much shorter than that of large machines.

1.3.4. **The Logical Vs Physical Design**

The implementation of communication systems 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 too 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 CS is motivated by the need for cost reduction in tasks executing.

1.3.5. 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 [FORT 1985, STON 1987, BOKH 1988] and divided in to 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.
1.3.6. Types of Communication Systems

1.3.6.1. Horizontal Vs Vertical Distribution

By Vertical distribution we mean that there is a hierarchy of processors, shown in Fig.1.1 [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.

---

Figure 1.1 Vertical 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 Fig.1.2 [MART 1988].
1.3.6.2. 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.

1.3.7. Feature of Communication System

There are a variety of reasons for building any communication System, 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.

(iii) **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.

(iv) **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.

1.3.8. **Application area of Communication Systems**

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 Communication Systems. To name a few such type of application
are signal processing, meteorology, image analysis, cryptography, nuclear reactor control, solar & radar surveillance, simulation of VLSI circuits, and industrial process monitoring.

1.4 TASK ALLOCATION

The task allocation in a communication 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 such system under the constraints as:

(i) Memory capacity available at each processor and

(ii) Time constraints in executing task.

The advent of VLSI technology resulting in low cost microprocessor has made CS an economic reality in today's computing environment. The modularity, flexibility and reliability of CS make it attractive to much type of users, and several CSs 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 (i) how the Control Message [CM] and Data Message [DM] are exchanged in the CS and (ii) how to control the communication process along with the protocol definition.

The format of the CMs, the headers and the trailers of DMs 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 CSs are still in their primitive development stage. Some major problems present the following widespread uses CS are:

(i) The degradation in system throughput caused by the saturation effect,

(ii) The difficulty in evenly utilizing each processor in the CS,

(iii) A large gap between the engineering application requirements and existing distributed network architecture and

(iv) The difficulty in verifying task allocation resulted from any allocating model.

Communication 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 CS. 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 processing time 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 is shown in Fig.1.3.

![Diagram of General Structure of Tasks Allocation](image)

**Figure 1.3: 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 1980a, 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 \( P = \{p_1, p_2, p_3, \ldots, p_n\} \), of "n" processors interconnected by communication links and a set \( T = \{t_1, t_2, t_3, \ldots, t_m\} \) of "m" executable tasks. 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 \( P \), \( f : T \rightarrow P \). Then the total cost or Time of an allocation \( x \) can be expressed as follows:

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

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 \( p_l \)
Researchers have identified several approaches to the task allocation models. The following of them are concern with our work:

1.5.1 Graph Theoretical Approaches

The graph theoretical approaches suggest graphical methods to represent and allocate application tasks to various processors in a communication system. In most of the graph theoretical approaches, the solution processes begins with the abstraction of tasks and inter task communication cost through a graph model in which tasks are represented by nodes and inter task communications as weights on the non-directed edges connecting these nodes. By performing the min-cut algorithm the graph minimizes the total computation cost. These approaches do consider load balancing, resource limitation, and generalization issues without giving much consideration to the computing time and complexity. Graph theoretical algorithms become NP-hard for CS consisting of reasonable number of processors.

1.5.2 The Integer Programming Approaches

The integer programming approaches are based on the direct enumeration algorithms subject to the additional constraints. This allows real time embarrassments, and rather complicates it to be incorporated into the allocation model to meet various application requirements. These approaches are limited and do consider the system
resources limitation, the amount of real time computation, and memory needed to obtain an optimal solution, as these grow exponential functions of the order of the problem.

1.5.3 The Heuristic Approaches

The heuristic approaches provide approximate solutions to task allocation and load balancing problems. These approaches require less computation time as compared to the integer programming methods and are applicable in the situations where an optimum solution is not achievable within the reasonable time limit.

1.6 INTRODUCTION TO RELIABILITY

Reliability theory has grown into an engineering science in its own right following the World War II. Much of the initial theory, engineering, and management techniques centered about hardware, however, human and procedural elements of a system were often included. Since the late 1960s the term software reliability has become popular, and now reliability theory refers to both software and hardware reliability [SHOO 1983].

1.6.1 Combinational Reliability

In performing the reliability analysis of a complex system, it is almost impossible to treat the system in its entirety. The logical approach is to decompose the system into functional entities composed of units, sub systems, or components. Each entity is assumed to have two states, one operational and one failed. The sub division generates a block-diagram or fault-tree description of system operation. Models are then formulated
to fit this logical structure, and the calculus of probability is used to compute the system reliability in terms of the sub division reliabilities.

### 1.6.2 Series Configuration

The simplest and perhaps the most common structure in reliability analysis is the series configuration. In the series case the functional operation of the system depends on the proper operation of all system components. A series reliability configuration is portrayed by the block-diagram representation shown in Figure 1.4(a) and the reliability graph shown in Figure 1.4(b). In either case, a single path from cause to effect is created. Failure of any component is represented by removal of the component, which interrupts the path thereby causes the system to fail.

![Block Diagram of Series with n-components](image)

**Figure 1.4(a)**

![Reliability Graph of a Series system with n-components](image)

**Figure 1.4 (b)**
1.6.3 Parallel Configuration

In many systems several signal paths perform the same operation. If the system configuration is such that failure of one or more paths still allows the remaining path or paths to perform properly, the system can be represented by a parallel model. Block diagram and reliability graphs for a parallel system are shown in Figures 1.5(a) and 1.5(b) respectively. There are "n" paths connecting input to output, and all units fail; in order to interrupt all the paths. This is sometimes called a redundant configuration.
Reliability Graph of a Parallel System with n-Components
Figure 1.5 (b)

1.6.4 An-r-Out-of-n Configuration

In many problems the system operates if “r” out of “n” units function e.g., a bridge supported by “n” cables, “r” of which are necessary to support the maximum load. One can draw a reliability graph as an aid. Each parallel path will contain “r” different elements, corresponding to the combinations of “n” things “r” at a time.

1.7 SCHEDULING POLICIES IN A CS

The scheduling policies in a CS 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 CS as shown in Fig. 1.6. 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 [BUCK 1979, CASA 1988]. The global scheduling relates to the problem of deciding where jobs or tasks [ROTI 1994]
run. 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: centralized and decentralized. 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 de-centralized.

Figure 1.6: Taxonomy of Scheduling
1.8. DEFINITIONS, ASSUMPTIONS AND NOTATIONS

1.8.1 Definitions

(i) **Inter Task communication Time**: The Inter Task Communication Time $t_{ik}$ of the interacting task's $t_i$ and $t_k$ depends on the period in which data units exchanged between them during the time execution.

(ii) **Execution Time**: Each task $t_i$ has an Execution Time $e_{ij}$ (where $1 \leq i \leq m$ and $1 \leq j \leq n$) that depends on the processor $P_j$ to which it is assigned and the duration of work to be formed by the task.

(iii) **Load Balancing**: Spreading the total load of a program / application on the available processing elements as evenly as possible is referred to as load balancing.

1.8.2. Assumptions

To cope up with the real life problems and to keep the algorithms reasonable in size the following assumptions are made:

(i) The number of tasks to be allocated is more than the number of processors, as normally is the case, in the real life CSs.

(ii) Whenever two or more tasks are assigned to the same processor the Inter Tasks Communication Time between them is assumed to be zero.

(iii) Whenever two or more tasks are assigned to the Inter Processor Distance between them is assumed to be zero.

(iv) If a task is not executable on a certain processor due to absence of some resources, the Execution Time of the task is taken to be infinite ($\infty$).
### 1.8.3 Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CT</td>
<td>Communication Time</td>
</tr>
<tr>
<td>CTM(_)</td>
<td>Communication Time Matrix of Order m x m</td>
</tr>
<tr>
<td>et&lt;sub&gt;ij&lt;/sub&gt;</td>
<td>Time of execution of the i&lt;sup&gt;th&lt;/sup&gt; task on the j&lt;sup&gt;th&lt;/sup&gt; processor</td>
</tr>
<tr>
<td>CRM(_)</td>
<td>Communication Reliability matrix of order m x m</td>
</tr>
<tr>
<td>CR</td>
<td>Communication Reliability</td>
</tr>
<tr>
<td>CUR</td>
<td>Communication Unreliability</td>
</tr>
<tr>
<td>CURM(_)</td>
<td>Communication unreliability Matrix of order m x m</td>
</tr>
<tr>
<td>ET</td>
<td>Execution Time</td>
</tr>
<tr>
<td>ETM(_)</td>
<td>Execution Time Matrix of order m x n</td>
</tr>
<tr>
<td>EUR</td>
<td>Execution Unreliability</td>
</tr>
<tr>
<td>EURM(_)</td>
<td>Execution Unreliability Matrix of order m x m</td>
</tr>
<tr>
<td>IPROD</td>
<td>Total number of assignments (Combinations)</td>
</tr>
<tr>
<td>m</td>
<td>Number of Tasks</td>
</tr>
<tr>
<td>n</td>
<td>Number of Processors</td>
</tr>
<tr>
<td>MCT(_)</td>
<td>An array to store Maximum Communication Time</td>
</tr>
<tr>
<td>nar</td>
<td>Counter for Allocation</td>
</tr>
<tr>
<td>NETM(_)</td>
<td>Modified Execution Time Matrix of order n x n</td>
</tr>
<tr>
<td>NERM(_)</td>
<td>Modified Execution Reliability matrix of order n x n</td>
</tr>
<tr>
<td>NEURM(_)</td>
<td>Modified Execution Unreliability Matrix of order n x n</td>
</tr>
<tr>
<td>P</td>
<td>Set of Processors {p&lt;sub&gt;1&lt;/sub&gt;, p&lt;sub&gt;2&lt;/sub&gt;, p&lt;sub&gt;3&lt;/sub&gt;,.............p&lt;sub&gt;n&lt;/sub&gt;}</td>
</tr>
<tr>
<td>PCOMB</td>
<td>An Array to store processors combinations</td>
</tr>
<tr>
<td>POS[.]</td>
<td>A matrix to store the positions of communication time on the bases of MCT(_)</td>
</tr>
<tr>
<td>T</td>
<td>Set of task {t&lt;sub&gt;1&lt;/sub&gt;, t&lt;sub&gt;2&lt;/sub&gt;,.............t&lt;sub&gt;m&lt;/sub&gt;}</td>
</tr>
<tr>
<td>T&lt;sub&gt;ass&lt;/sub&gt;</td>
<td>Assigned Tasks</td>
</tr>
<tr>
<td>TCOMB()</td>
<td>An Array to store tasks combinations</td>
</tr>
<tr>
<td>T&lt;sub&gt;non-ass&lt;/sub&gt;</td>
<td>Non-Assigned Tasks</td>
</tr>
<tr>
<td>NCTM(_)</td>
<td>Modified Communication Time Matrix of order n x n</td>
</tr>
<tr>
<td>pet()</td>
<td>An array to store processor wise execution Time</td>
</tr>
<tr>
<td>itctm(_)</td>
<td>Inter tasks Communication Time Matrix of order m x n</td>
</tr>
</tbody>
</table>
1.9 ORGANIZATION OF THE THESIS

The First chapter of this thesis entitled “TO STUDY AND ANALYSIS OF OPTIMIZATION TECHNIQUES AND THEIR APPLICATION TO EVOLVE SOME ALGORITHM TO PERFORMANCE ENHANCEMENT OF COMMUNICATION SYSTEM” is devoted to the introductory background of the topic in optimization techniques, communication systems, task allocation and other issues directly related to the work. A sequential historical introduction to the optimization has been described in details. Various aspects of the optimization have been mentioned. This chapter contains the different techniques of the optimization such as, Graph theoretic approaches, Integer programming approaches and many more. A brief overview of the communication system has been given to understand the basics of it. Assignment problem or let us say task assignment problem in these systems, is discussed and also some of the related concept has been included. In order to complete this piece of research we use number of definitions, assumptions and notations, all such definitions, assumptions and notations, which are used throughout the thesis has been put together and listed in this chapter.
Finally, the organization of the thesis, include the chapter wise brief summary of the present work has been mentioned in this chapter.

Task assignment of any communication system is a most computing 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 known 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 correlate to our work. We have categories them in two groups such as commonly adopted approaches and consideration of a particular issue. In commonly adopted approaches we have considered the Branch and Bound methods, Dynamic Programming, Decomposition approach, Genetic algorithmic approach, Heuristic approaches, Integer Programming approaches, Network Partitioning algorithm, Problem Reduction method, Petri Net modeling & Reliability evaluation and Shortest Path algorithmic method. On the other hand for consideration of a particular issue, we have chosen the precedence constraints, particular architecture consideration and reliability evaluation.

The Third Chapter of the thesis is the important and main, as it include all five research problems. The general Motivation for the present research has been written briefly. The Objective, Technique, Computational Algorithm, Implementation and
Conclusion of each problem are mentioned in this chapter. The Details of each problem is as mentioned below:

In the Problem – I, the assignment problems in communication systems are one of major factor to determine the performance of such systems. The present piece of research problem suggests an exhaustive search approach to obtain the optimal time for optimally assigning the tasks to the processors of communication system. Here we define an index that is based on the time for the execution of task to various processors and also communication time amongst the tasks. The computational algorithm of the approach and its implementation are mentioned. The results are shown in tabular form as well as graphical 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 coded into C++ and several sets of data have been tested to verify the effectiveness of the algorithm.

The title of the Problem – II is Optimizing the time of CS: Matrix Partitioning Approach. The performance analysis of any communication system has an important area of research these days. The assignment problems in communication systems are one of the major problems to determine the performance of these systems. The present piece of research problem – II suggests a matrix partitioning approach to assign all the tasks to the available processors of the system. Here we have chosen such a problem in which the tasks are more as compared to the processors of the system. It contains the computational algorithm of the technique and its implementation. This piece of research is capable to deal all such real life situations, where the tasks, which are to be executed on the
processors of the communication systems, are more than the number of processors of that system. This technique does not require adding dummy processors. The developed algorithm is coded into C++. The several sets of input data have been tested to verify the effectiveness of the algorithm.

In the Problem - III, a new approach for assignment in communication system based on tasks weight matrix partitioning has been considered. This suggests an efficient algorithm for communication system. The problem chosen, in which the number of tasks is more than the number of processors in the system, as it is most of the time the case of real life situation. The tasks are allocated to the processor in such a way that to suit the desired execution requirement of the tasks. The model addressed here is based on the consideration of task execution time and task weights. The method is presented in algorithmic form and implemented on the several sets of input data to test the performance and effectiveness of the algorithm. The developed algorithm is coded into C++.

In the Problem - IV, we have used the Matrix partitioning technique to improve the reliability of communication system. It is almost impossible that the communication systems has to execute only as many tasks as the number of processors available in the system. That means the number of tasks, which are to be executed on the communication systems, shall be the more as compared to the number of processors in the systems. The problem of execution of "m" tasks to "n" processors (m > n) in a communication systems is addressed here through an efficient algorithm for the task execution in a
communication system. The execution reliabilities of the tasks on different processors have taken into consideration while preparing the algorithm to such a case. The model discussed here is based on the consideration of execution unreliability's of the tasks to the processors. Keeping in view we suggested an efficient method to assign all the tasks as per the required availability of processors so that none of the tasks get remains unexecuted and the present approach does not require to adding dummy processors. The several sets of input data are considered to test the effectiveness and efficiency of the algorithm. It is found that the algorithm is suitable for arbitrary number of processor with the random program structure and workable in all the cases.

Our day-to-day experience reveals that processor distance influences the overall task completion cost in any communication system, for instance, in the case of telecommunication network. It is well known fact that a person has to pay more for talking to person sitting at a long distant place. The case of Communication System is similar to that of a telecommunication network in this aspect. In a system, a task is allocated to a processor in such a way that extensive Inter Task Communication is avoided and the capabilities of the processor suit to the execution requirements of the task.

The model discussed in the Problem - V provide an optimal solution for assigning a set of “m” tasks of a program to a set of “n” processors where m > n in a Communication System with the goal to maximize the overall throughput of the system and allocated load on all the processors should be balanced. The present allocation policy
involves stepwise modification of execution and communication matrices by making the clusters tasks. Some of the tasks may not involve in any cluster, those tasks treated as independent tasks. The mathematical programming approach has been used to determine the optimal allocation of tasks. The several sets of input data are considered to test the complexity and efficiency of the algorithm. It is found that the algorithm is suitable for arbitrary number of processor with the random program structure and workable in all the cases.

In the Fourth Chapter, as we all know that research is an on going process and there is no last limit so that we have coin some of the new research problems, each are to be solved by the future researcher. Keeping these things in mind, the last chapter of this thesis contains the future scope of this research area and new research problem along with the suggestions for the researchers.

During the period of this research work some of the research papers have been presented in various seminar and conferences. Two research papers are published in Journal (Copies of both papers are appended at the end of the thesis). And also some of research papers have been sent for the publication in various 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 of in alphabetic order of such research material has been attached.