SURVEY OF THE LITERATURE

2.1 AN OVERVIEW

Over the past few years a number of studies in optimization techniques and their applications to Communication System (CS) have led to the identification of several challenging problems. One of these problems is optimally assigning the tasks to the processors in the CS. In such system, it is essential to assign each task to the processor whose characteristic are more suitable for the execution of the task. Availability of information at geographically dispersed locations and the need of its transmission from one source to another have also led to the development of CS.

The task allocation problem in a CS may either be static or dynamic. If the allocation is static the information regarding over all costs of a hardware configuration is known before hand. When a task is assign to a particular processor, it stays on that processor during its life time. The problem of assigning m tasks to n processors, where m > n as normally is the case, in the real life is a complex one. The complexity may be reduced by an efficient task partitioning strategy [BERG 1987]. In case of dynamic allocation the tasks or modules can reassign 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. Recently, Zhou, Zhibo, Zhou, Tong and Jinxiang Wang [ZHOU 2007] have discussed the bit error rate performance of an Invariant Delay Multiple Access DCSK (IDMA-DCSK) Communication System over multi path fading channels. They have established the close relation between theoretical and the simulation results. Ha, C and Kuo, W. [HA 2006] has also presented the multi-path heuristic for redundancy allocation. Onishi, J., Kimura, S., James, R.J.W., Nakagawa, Y., [ONIS 2007] has solved the redundancy allocation problem by using surrogate constraint method. A through survey of the related literature reveals that the earlier researches can be classified in the following two major categories

- Commonly Adopted Approaches
- Consideration of a Particular Issue

### 2.2 COMMONLY ADOPTED APPROACHES

#### 2.2.1 Branch –and – Bound Methods

Branch – and – Bound technique is very popular and widely adopted technique among the researchers. There exist several algorithms based on this technique for task allocation problem [AROR 1980, MA 1984, SINC 1987]. An optimal task allocation strategy with main concern of maximizing reliability and considering communication cost as the constraint function for a distributed database management system has been reported in [VERM 1997]. The model is converted into a state space search tree and Branch- and –Bound technique is used to achieve the optimum results. This approach has
been adopted for optimizing the distributed execution of join queries by Reid [REID 1994].

2.2.2 Dynamic Programming

Dynamic programming is an algorithm design method that can be used when the solution to a problem may be viewed as the result of a sequence of decisions. The main feature of this technique is that it often drastically reduces the amount of enumeration by avoiding the enumeration of some decision sequences that can not possibly be optimal. This approach aims at finding the optimal sequence of decision employing the principal of optimality. Dynamic programming based solution procedure can be regarded as recursive optimization because the decisions, leading to an optimal solution, are made one at a time. A number of research papers based on dynamic programming approach are available in the literature of task allocation problem in CS. Employing this approach, Bokhari devised a shortest tree algorithm that runs in \(O(mn^2)\) time for the assignment of \(m\) modules to \(n\) processors [BOKH 1981a]. A relationship between module allocation and non–serial dynamic programming has been established by Rosenthal [ROSE 1982]. Considering cost and reliability aspects, Ashrafi et al [ASHR 1992] developed optimization models for selection of programs. Dynamic programming technique is applied by Ferman Dez-Baca [FERN 1989] for obtaining the optimal assignment through local search.

2.2.3 Decomposition Approach

The reliability evaluation of complex networks using decomposition method is also one of the important techniques. The reliability of an oriented type-I network is
computed by Nakazawa [NAKA 1976] by decomposing the network into short-and-open-removed sub networks according to its keystone line called a removable line. In another paper [NAKA 1977] the type-I of networks is extended to types-II, II & III and equivalence of one non oriented line and one pair of oriented lines with equal 1-reliability is proved for types I, II & III. It is also proved that the type III networks have to be transformed into a Boolean expression and then the Boolean decomposition method [NAKA 1977] can be applied. A network [NAKA 1978] is represented by matrices which is suitable for computer use. The criteria for removable lines of complex networks are proved. It is also proved that there are always removable lines of complex networks except oriented type III one, using the matrix expression [NAKA 1981]. For determining the conditional success events Aggarwal et al [AGGA 1982] evolved a technique using both the node removal and connection multiplication methods for path enumeration. Li et al [LI 1992] suggested an algorithm for optimization of large-system reliability with the help of decomposition method.

2.2.4 Genetic Algorithmic Approach

This optimization approach is a heuristic optimization technique, which includes simulated annealing. Tabu searched an evolutionary strategy [REEV 1993]. Coit and Smith [COIT 1996] developed a specific genetic algorithm to analyze series-parallel system and to determine the optimal design configuration where redundancy allocation problem for a series-parallel system had a specified number of sub systems and for each sub system, there are multiple component choices which can be selected and used in parallel. For those system designed using off-the-shelf component types, with known cost, reliability and weight, the system design and component selection become a
combinational optimization problem. They [COIT 1998a] also presented an algorithm to solve the redundancy allocation problem when the objective is to maximize a lower percentile of the system time-to-failure distribution. Levitin et al [LEVI 1998] discussed a redundancy optimization problem to multi-static systems. Here the systems and their components have the range of performance levels. Dengiz et al [DENG 1997] used the similar approach to solve the all-terminal network design problem when considering cost and reliability. Genetic algorithm was also used in optimization of system reliability [KUMA 1995b, PAIN 1995], while Kumar et al [KUMA 1995a] worked on the topological design for such systems. Exploring Genetic Programming and Boosting Techniques to Model Software Reliability has been studied by Costa and E.O. de Souza, G.A., Pozo, A.T.R., Vergilio, S.R. [COST 2007].

2.2.5 Heuristic Approach

Heuristic Algorithm has also played a vital role for providing solutions to different types of problems. Several research papers [DENG 1997, LU 1986, WARD 1984, PRIC 1984, WILL 1983, ALFO 1988, HWAN 1993, EFE 1982, AROR 1980] have been reported in the literature regarding heuristic approaches for task allocation problem of these systems. The use of CS has been rapidly increasing in order to share expensive hardware and software resource and provides access to main systems from the distant locations. Dengiz et al [DENG 1997] suggested a heuristic algorithm inspired by evolutionary approach to solve the all terminal network design problem when considering cost and reliability. Kartik and Murthy [KART 1995] provided a heuristic algorithm to find an optimal and sub optimal task allocation in redundant distributed computing.
systems, to maximizing system reliability. Further it repeatedly consider all the possible reassignments of tasks at a time, performs the most advantages reassignments and terminals when no more profile reassignments is discovered. On the other hand, the problem of minimizing the cost by making the clusters of those tasks that is heavily communicating and assigning the tasks cluster to appropriate processors. On the basis of execution drew considerable attention [WILL 1983, PRIC 1984, WARD 1984]. Some heuristic approaches for task scheduling based on characterization of inter module communications [LAN 1985] and reliability maximization [HWAN 1993] in the distributed computing system also deserves mention. The heuristic load balancing approaches [BOLC 1983, YANG 1987] suggested reducing the number of tasks by forming task clusters equal to the number of processors. This reduction is accomplished by fusing those tasks between them data exchanges are maximum. The task-clusters so formed are assigned to the processors. Finally tasks are shifted from the heavily loaded processors to the minimally loaded processors to balance the load on all the processors. Liu X. et al [LIU 1999] has presented a study of building systems from components to answer the questions related the configuration of components and performance relation with monolithic system. They has also discussed the generality of the technique with regard to systems other than CS. Wang, N, Lu, J.C. and Kvam, P [WANG 2007] has also given the reliability modeling in spatially distributed logistics systems. Another simple heuristic algorithm for generating all minimal paths has been suggested by Yen Wei Chang [YEN 2007].
2.2.6 Integer Programming Approach

A task allocation model becomes more significant and realistic when it incorporates real-time constraints such as inter-processor communication, memory limitations of each processor, etc. For the task allocation problems, Integer Programming is a useful and exhaustive approach as it is capable to reflect real-life situations of distributed processing and simplicity in application. A number of researchers have worked on task allocation problems employing this technique to determine the optimum solution [MA 1982, DESS 1980, CHU 1980a]. A model based on this approach is developed by Chu for optimum file allocation in a multiple computer system [CHU 1969]. Another similar approach for data file allocation has been reported by R. Marcogliese and Novarese [MARC 1981].

2.2.7 Network Flow and Network Partitioning Algorithm

Network Flow algorithm keeps an important role for the task allocation problems. Stone used this technique for the task allocation in the multiprocessor systems in 1977 [STON 1977] by making use of Ford Fulkerson algorithm [FORD 1962] for finding maximum flows in commodity networks as modified by Edmonds and Karp [EDMO 1972, KARZ 1974, DINI 1970]. The complexity of this algorithm is known to be $O((m+2)^3)$, where $m$ is the number of tasks. This method provides no mechanism for representing memory size or other constraints. Wu et al [WU 1980a] employed network flow algorithm for task allocation and also described the partition algorithm for parallel and distributed processing [WU 1980b]. In another algorithm load balancing in a circuit-switched multi computer has been used by Bokhari [BOKH 1993]. Ke and Wang [KE
1997] proposed an efficient reliability evaluation algorithm accounting for imperfect nodes in distributed computing networks. Based on the concept of network partition, the algorithm exploits some simple efficient techniques to handle the unreliable nodes, for directly computing the network reliability expression considering imperfect nodes instead of using any compensating method.

2.2.8 Problem Reduction Method

A smaller problem can be handled more effectively in comparison to a bigger problem. It is therefore, always advisable to reduce the size of a given problem, if possible. A task allocation problem can be simplified to a certain extent by reducing the size of the problem. Arora and Rana, who proposed module assignment in two processor distributed system [AROR 1979], adopted this idea for proposing the concept of clustering the tasks which exhibit certain peculiar behavior to reduce the problem size [AROR 1981]. A task is either assigns to a processor or fused with another task(s) depending upon some criterion. Sagar and Sarje [SAGA 1991] used the above technique for task allocation model for evolving new distributed systems. A clustering algorithm for assignment problem of arbitrary process systems to heterogeneous distributed computer system is implemented by Bowen et al [BOWE 1992]. He showed that clustering algorithm exhibit better time complexity. The clustering technique used by Kim and Browne [KIM 1988] iteratively applies a critical path algorithm to transform the graph into a virtual architecture graph which consists of a set of linear clusters and the interconnection between them. A new heuristic, based on the concept of clustering, to allocate tasks for maximizing reliability is proposed by [SRIN 1999]. Based on clustering approach, a new multiprocessor scheduling technique named de-clustering has been
given by Sih and Lee [SIH 1993b]. They claimed that their de-clustering approach not only retains the clustering advantages but at the same time overcomes its drawbacks.

2.2.9 Petri Net Modeling & Reliability Evaluation

The probabilistic graphs, being static in nature can not efficient model and analyze the dynamic behavior of the system [JOLL 1980]. A Petri net approach introduced by Petri [PETR 1962] and Holt [HOLT 1978] is recognized as one of the most powerful modeling and analysis tool for a communicating processing environment. Petri net theory to model a distributed processing environment is also studied by Murata [MURA 1979]. Kumar and Aggarwal [KUMA 1993] suggested Petri net model for the evaluation of reliability for the execution of a computer program in a CS. The execution of a program in a distributed computing system may require access to several files residing at different sites and communications paths between several node pairs. Then, by using the ratability, firing and marking concepts of Petri net, an algorithm is developed to study the distributed computer program reliability and the CS reliability. Earlier Kumar et al [KUMA 1988, KUMA 1990] used this approach to obtain the reliability of a broadcasting network. Lopez [LOPE 1994] presented a model based on stochastic Petri net to estimate the reliability and availability of programs in a DPE. Some models [SHAT 1992, SHAT 1989] for reliability oriented task allocation in redundant distributed computer system and estimating the reliability and availability of programs in a distributed processing environment. Recently Schnee Weiss, W.G. [SCHN 2007] presented a review of Petri net picture book and Petri nets for reliability modeling. The reliability of tree structured grid services has been studies by Yuan – Shun, Dai [YUAN
2006, YUAN 2007]. The study related to confidence bounds for system reliability was presented by Ramirez-Marquez, J.E. and Wei, Jian [RAMI 2006].

2.2.10 Shortest Path Algorithmic Method

The solution of the task allocation problem for a CS can be obtained by using the shortest path algorithms. Work based on such algorithms is reported in the literature [BOKH 1981a, PRIC 1982, TOWS 1986]. In order to apply this approach, a program graph is transformed into an assignment graph. For each node in the program graph, there exists a set of nodes corresponding to the number of processors in this assignment graph. The nodes and edges between them are labeled suitably depending on the method applied to obtain a shortest path from source-nodes to terminal-node. A shortest tree algorithm described in [BOKH 1981a] minimizes the sum of the execution and communication costs for arbitrarily connected distributed systems with arbitrarily number of processors, provided the interconnection pattern of modules form a tree. The objective of the study was to allocate the tasks, whenever possible, to the processors on which they execute fastest while taking into account the over head due to IPC. Dynamic programming approach was applied for determining the shortest path and then optimal assignment of the tasks of a program over the processors of an inhomogeneous DCS was obtained. The worst-case complexity of the method is found to be \(O(n^3\phi^3)\), where, ‘n’ is number of nodes in assignment graph and ‘\(\phi\)’ is the number of phases. Price and Pooch claimed that their shortest path method is applicable to all cases with the limitation that an optimal solution is possible in limited cases only [PRIC 1982]. The shortest path algorithm evaluates the set of nodes in the assignment graph corresponding to a program graph to
select the best possible allocation of a task to a processor. Towsley’s work [TOWS 1986] is a special cases where the inter task communication pattern is series parallel in nature. A search is made in the assignment graph for the parts of the program graph where inter-task communication patterns are in series parallel or tree in nature. For such inter task communication patterns, shortest paths are derived and these shortest paths are combined to get the overall shortest path of the assignment graph. In most of the other related studies [BOKH 1987, IQBA 1986a, IQBA 1986b, KOHL 1975, BOKH 1988], and optimal task assignment graph is constructed which contains as many layers as number of processors. The calculation of weights of edges depends on the nature of the program graph and the processor mechanisms. Allocation of task interaction graphs to processor in heterogeneous networks is reported by Hui and Chanson [HUI 1997]. Keinghan [KEIN 1971] used a form of dynamic programming approach by partitioning a chain structured program graph. Bhardwaj et al. [BHAR 1994] have applied a similar method for optimal sequencing and arrangement in distributed single level tree networks with communication delay. Some bottlenecks of the shortest path methods are that the assignment graph becomes very large and complex as the number of processors in the distributed system increases and subsequently number of computations increase for calculating the edge weights. Further to add, the entire assignment graph has to be retained in the memory until the final assignment is made. Ramirez–Marquez, J.E. and Gebre, B.A., [RAMI 2007] suggested a classification tree based approach for the development of minimal cut and path vectors of a capacitated network.
2.3 CONSIDERATION OF A PARTICULAR ISSUE

2.3.1 Precedence Constraint
Earlier task allocation models did not consider modules precedence constraints as one of the significant factor. However, practically, it is not possible to start execution of a task before the completion of another task until the task is independent. Therefore, besides considering load balancing and minimization of IPC cost the precedence relation among task must also be taken into account. Some studies of task allocation models [CHOU 1986, MARK 1986] considered the precedence relation as a criterion for assignment. The model developed by Markencoff and Liaw [MARK 1986] is applicable for the problems exhibiting parallelism. The job comprising a set of tasks is modeled as a directed a cyclic graph. The graph has been partitioned into a set of sub graphs. Each directed a cyclic graph corresponds to an independent process without any data flow between sub graphs. The distribution of sub directed cyclic graphs, to processors, as evenly as possible, using branch and bound method, results in optimal assignment without IPC cost. The application for the response time of the threads is important rather than the entire application. Chu and Leung [CHU 1984b] reported an algorithm to search for task allocation that minimizes response time. The suggested algorithm considers the task replication and assignment together. Under any given task allocation process, to reduce response time, tasks reallocated from the longest weight processor to shortest weight processors until no further improvement is possible by such task reallocation. An extension for this algorithm with the objective of minimizing response time is proposed by Chu and Lan [CHU 1987b]. The objective of task allocation is to counter balance the related but apposite factor of IPC cost and load balancing to achieve maximum throughput and to satisfy real time constraints.
2.3.2 Particular Architecture Consideration

The problem of assigning a set of \( m \) tasks to a particular multiprocessor architecture having \( n \) processors, where \( m > n \), is known to be a mapping problem. This problem has been attempted by many researchers [BERM 1984, BERG 1985, BERG 1987, BOKH 1987, FUKU 1987, LEE 1987, MENED 1987, MURT 1988, MURT 1989, SIVA 1988, SIVA 1989, SHMI 1991, CHUA 1992, LIAN 1994, OLSO 1994] to obtain efficient mapping schemes. Chuang et al [CHUA 1992] suggested a fast recognition complete processor allocation strategy for hypercube computers. This model referred to a dynamic processor allocation scheme where the search space generated is dependent upon the dimension of the requested sub cube dynamically, rather than being predetermined and fixed. Olson et al [OLSO 1994] have evolved a model for the fault-tolerant routing in mesh architectures for the distributed computing system. Network flow models of message flow in the mesh and hypercube have been developed by Bokhari [BOKH 1993]. Sih and Lee [SIH 1993a] assumed the target architecture for their proposed scheduling technique to be shared-bus multiprocessor. Another polynomial-time algorithm is developed for computing the distributed program reliability for star topologies of CS in which data files are restricted to a certain type of distribution [LIN 1990].

2.3.3. Reliability Evaluation Consideration

Distributed computer systems are increasingly being employed for critical applications, such as aircraft control, industrial process control, and banking systems. Maximizing performance has been the conventional objective in the allocation of tasks
for such systems. There are many measures to evaluate the performances of CS. Reliability of a CS is another very important issue that need special attention [AGGA 1982, GARC 1982, SEGE 1994]. A reliability measure called, distributed program reliability, to model accurately the reliability of CS has also been reported [KUMA 1988]. Some models of reliability oriented task allocation in redundant distributed computer systems [SHAT 1989, SHAT 1992] and estimating the reliability and availability of programs have also appeared in the literature. A 1-step algorithm GEAR (Generalized Evaluation Algorithm for Reliability), capable of computing several reliability parameters of a CS such a terminal-pair reliability, computer network reliability, distributed program reliability, and distributed system reliability, is also studied [KUMA 1993]. Software reliability allocation, based on several factors such as structure, utility, price, and cost, is proposed by Zahedi [ZAHE 1991]. Strategies are devised that achieve a maximal combination of safety and reliability [RAI 1982]. Efficient methods to optimize the system reliability have been reported in [PAIN 1995, RAGH 1985]. Reliability analysis for a distributed program is mentioned in [KUMA 1996, KE 1997]. A genetic algorithm based reliability optimization model for computer networks is proposed by Kumar et al [KUMA 1995a]. Further, an investigation of the problem of distributed-program reliability in various classes of distributed computing systems has been presented [LIN 1990]. Srinivasan and Jha suggested a new clustering-based safety and reliability driven heuristic for a heterogeneous distributed system [SRIN 1999].