A Survey of Static Scheduling Algorithm for Distributed Computing System

Khushboo Singh	Mahfooz Alam	Sushil Kumar Sharma
M.Tech (CSE)	M.Tech (CSE)	M.Tech (CSE)
Institute of Technology & Management	Institute of Technology & Management	Institute of Technology & Management
Aligarh, India	Aligarh, India	Aligarh, India

ABSTRACT
The static scheduling algorithms are widely used to evaluate the performance of distributed computing system. In such systems, purpose of scheduling algorithm is to allocate tasks to available processor so as to efficiently utilize this processor and to reduce the makespan, total computational cost and various other such factors, with the motive of achieving optimal solution. Scheduling algorithms are classified into two broad categories i.e., static scheduling algorithms and dynamic scheduling algorithms. In this paper we are discussing various static scheduling algorithm and numerous problems in various levels of the homogeneous and heterogeneous distributed systems. Also we are comparing these algorithms on the basis of various factors such as speedup, time complexity, scheduling length ratio, normalized scheduling length and so on.

General Terms
Multiprocessor task scheduling, Static scheduling, Distributed System

Keywords
Performance evaluation, Speedup, Scheduling length ratio

1. INTRODUCTION
In multiprocessor computing system numerous of tasks are running at the same time on parallel processors [6]. So, in order to achieve high performances in such system, scheduling plays an important role. We can define scheduling as the process of arranging tasks in a particular way, especially with the reference of sequence of their arrival or according to their computational time. A good scheduling strategy helps in increasing the efficiency of a system and in utilization of available resources in the best possible way. Scheduling can be broadly classified as; static Scheduling and dynamic scheduling[13].

Static scheduling is also known as deterministic or offline scheduling. In such type of scheduling algorithm, scheduling is done at compile time and no run time scheduling is done. All the parameters regarding the task is known in advance such as, data dependencies between the tasks, execution time, and so on. They can be further classified as heuristic based algorithm and Guided random search algorithm. Static scheduling found useful in many areas like for simulation studies, for post-mortem analyses and also for designing a system.

Dynamic scheduling is also known as non- deterministic scheduling or online scheduling. Scheduling decisions are done during run time. Scheduling is based on parameters done during run time. Scheduling is based on dynamic parameters, which may change during run time.Further this paper is organized as follows: Section2 describes the classification of task scheduling algorithm. Section 3 covers the review of work done in the field of static task scheduling. In section 4, comparative study of various static scheduling algorithms is discussed on the basis of various factors in task scheduling algorithms in tabular form. Conclusion and future scope is given in section 5.

2. TASK SCHEDULING
The efficiency of executing parallel applications on the distributed computing systems depends on the method used for scheduling the tasks of the parallel application onto the available processors. Scheduling the task includes dividing the tasks into smaller subtasks. Dependencies among these subtasks are represented using Directed acyclic graph (DAG). DAGs helps in finding an optimal solution for the multiprocessor task scheduling problem. Driving an optimal solution for task scheduling problem is an NP hard problem [14]. Directed Acyclic Graph (DAG), In multiprocessor system, tasks are divided into several subtasks for parallel execution. To repesent data depedencies and communication time between tasks, directed acyclic graphs are used. Proper representation of tasks is necessary for efficient execution of of scheduling algorithm. DAG is a dircted graph which contains set of edges and vertices. Each vertex represents the task and every edge represents the relation between two nodes or vertices connected through that edge. Starting node of the directed graph is known as entry node, it is a node without any parent node. Node with no child is know as exit node. Figure 1 represents the DAG for seven tasks named T1, T2, T3, T4, T5, T6 and T7.
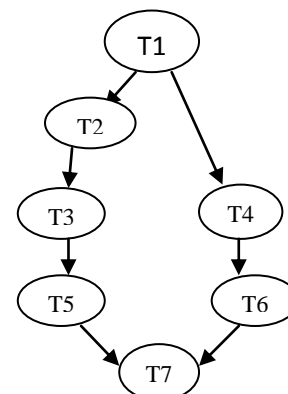

Figure 1: Directed Acyclic Graph (DAG).

Task scheduling strategies are required in distributed computing system where, numerous of tasks are executed in

same time. Task scheduling can be done in both Homogeneous distributed computing system and in Heterogeneous distributed computing system.Classification of task scheduling is shown with the help of figure 2, given below.
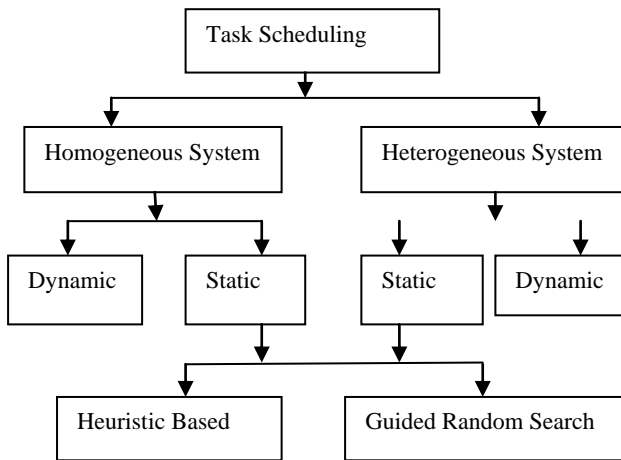

Figure 2: A Taxonomy of Task scheduling Algorithm.

2.1 Homogeneous Distributed Computing System (HMDCS)

A bunch of processors is connected to a high speed network for accomplishing the required task, is defined as distributed computed system (DCS). HMDCS is a part of Distributed computing system. Itcomprises of group of identical processors connected to a high speed network. Identical processors means all the processors connected to the system are of same computationalspeed and processor complexity.To achieve maximal utilization of all the available resources tasks needs to be divided equally among all the processors present in HMDCS [15]. Some examples of HMDCS algorithms are ISH, HLFET, TSB, MTSB etc.

2.2 Heterogeneous Distributed Computing System (HTDCS)

HTDCS can be defined as the interconnection of several processors, having different computational speed, and areconnected with different speed links used for solving group of problems which have different computational requirements. All the processors in this type of computing system, usually incorporatesspecialized processing capabilities and thus tasks are assigns to them accordingly. HTDCSs [15] are generally applicable in the commercial and scientific research fields where, real-time security, robustness and high performance, is the main priority.Some examples if HTDCS algorithms [1] are SNLDD, ACS, Tabu Search, NSGA-II, etc.

In both heterogeneous and homogeneous distributed computing system, proper task scheduling strategy is required in order to achieve high performance and to efficiently utilize available resources and processors.

3. REVIEW OF STATIC SHEDULING ALGORITHM

Task scheduling problem is hot topic for research these days. Many researchers have studied in the field of static task scheduling problem for multiprocessor system. Some of their findings on static scheduling are discussed below:

3.1 Heterogeneous earliest finish time (HEFT) Algorithm

HEFT Algorithm was introduced in 2005. Execution of this algorithm takes place in two main phases: task prioritizing phase and process selection phase. In first phase, priority of each task is calculated. In second phaseinsertion of task in an earliest idle time slot which is created between two already scheduled task on a processor. When immediate predecessors in task graph get executed then, task becomes ready for execution.HEFT scheduling algorithm [16] is applicable for bounded number of processors.

3.2 Monte Carlo Algorithm (MCA)

Monte Carlo algorithm reduces the makespan by applying probabilistic distribution approach. Threshold values are used to optimize scheduling. This algorithm avoids the complex computation with random variables. This algorithm was proposed by Wei Zheng and Rizos Sakellariou [5].

3.3 Critical Path on a Processor Algorithm (CPOPA)

In this algorithm works in two phases i.e. task prioritizing and process selection,just like HEFT algorithm but using different strategies. In Task prioritization phase, a priority queue is maintained with key values, upward rank (rank_u) and downward rank (rank_d). to implement this queue, a binary heap is used. Initially, entry task is marked as critical task. In second phase i.e. process selection phase if the selected path is on critical path, then it is scheduled on critical path processor (cpp); otherwise task is assigned to the processor which minimizes the Earliest Finish Time, EST, of the task. CPOP Algorithm [2] works for heterogeneous computing system.

3.4 Synthesized Heuristic Task Scheduling Algorithm (HCPPEFT)

The proposed algorithm is based on both duplication-based techniques and list-based approach for heterogeneous computing systems. This algorithm [17] comprises two phases, the task prioritizing phase and the resource selection phase. In the first phase, a new approach is used according to which threelevels of priority are proposed to choose task. In the second phase, the duplication of tasks is optimized.

3.5 Longest Dynamic Critical Path (LDCP)

In this algorithm a new attribute is used to identify the priority of each task in HeDCS. LDCP algorithm [17] works in three phases; task selection, process selection and status update phase. In first phase directed acyclic graph is constructed for each processor in the system. In second phase i.e. processor selection phase, task is assigned to the processor which helps in minimizing its finish time executing. In the last phase of this algorithm, status of the system is updated when task is scheduled on the processor. This algorithm is priority based.

3.6 Sorted Nodes in Leveled DAG Division (SNLDD)

It is static task scheduling algorithm for heterogeneous distributed computing system. Proposed algorithm [9] is known as high performance task scheduling algorithm. Unlike LDCP, it performs best with limited number of processors. In this algorithm DAG is divided into levels with considering the dependency priority conditions among tasks and at each level task is sorted in descending order, according to their computation size.

3.7 Highest Level First with Estimated Time (HLFET)

It is list scheduling algorithm for homogeneous multiprocessor systems. HLEFT is a priority based algorithm and uses static b-level priority. Here, minimum start time of tasks is used as the priority and thus, the task with minimum priority is assigned to the processor first. HLEFT algorithm [18] is simplest list scheduling algorithm and does not utilize the idle time slot between the two processors. Due to this drawback the performance of HLEFT algorithm degrades.

3.8 Insertion Scheduling Heuristic (ISH)

ISH algorithm overcomes the major drawback of HLEFT algorithm and thus it utilizes the idle time slot [11] between processors. It is based on HLEFT and hence, initially, it works like HLEFT for first node but once the idle slot is created ISH checks whether any task in ready queue can be inserted into the idle slot but cannot be scheduled earlier on the other processors. ISH algorithm [20] schedules such tasks as many as possible into the idle time slot.

3.9 Earliest Time First (ETF)

ETF algorithm is also a priority based scheduling algorithm for homogeneous system. In this algorithm, for each task in ready queue, earliest start time is computed and the one smallest start time is chosen and processor is allocated to that task. But if in case two nodes have same priority than the node with the higher static priority is scheduled first in comparison with the node with lower static priority. ETF algorithm was proposed by Hwang [19].

3.10 Constrained Earliest Finish Time (CEFT) Algorithm

This approach provides better scheduling for heterogeneous systems using the concept of constrained critical paths (CCPs).CCPs is found in the respective DAG and then tasks are scheduled using the earliest finish time approach. This approach helps in producing schedules with shorter makespan. Constrained Earliest Finish Time was proposed by Minhaj Ahmad Khan [8].

3.11 Multipriority Queueing Genetic Algorithm (MPQGA) Algorithm

MPQGA is a genetic search algorithm [12] and thus is used for solving combinatorial optimization problems. It solves the problem in two phases: first phase, MPQGA uses genetic algorithm to generate multiple priorities queuing for HeDCS and parallel heterogeneous computing system. Second phase, in this phase HEFT algorithm is used for searching a solution for mapping tasks to processors.

3.12 Tabu Search Algorithm (TSA)

Tabu search algorithm is neighborhood search technique that attempts to find the global optimum solution and avoids local optimum solution. It is basically a technique that keeps track of already searched solution region in order to avoid repeating the search near these areas It examines all of the moves done in its immediate neighborhood and then, among all such moves, the move with which results in best makespan is considered. In this way Tabu Search algorithm [12] finds global optimum solution.

3.13 Ant Colony System Algorithm (ACSA)

ACS algorithm is inspired by the method used by the group of ants for searching their nest and food. While searching for food they deposit pheromone (a chemical released by ants) on the path. By following this pheromone, trail of ants get to the food source. As this process continues most of the ants chooses shortest path by following the path where the large amount of pheromone have been deposited.ACS algorithm [2] obtains optimum solution.

3.14 Parallel Genetic Scheduling Algorithm (PGSA)

Proposed algorithmis a fresh approach in the field of genetic based algorithm. Thisalgorithmfocuses on providingabsolute solutions by reducing the computation time. PGS algorithm produces scheduling list better than other algorithms, developed so far, and eventually results in optimum solutions for a given problem. This algorithm was proposed by, Kwok, Yu-Kwong, and Ishfaq Ahmad [24]. Their main objective is to meet the requirements like, fast running,high performance and scalability. This algorithm gives absolute solutions.

3.15 Parallel Bubble Scheduling and Allocation Algorithm (PBSAA)

This algorithm is a static based approach which works well for both heterogeneous and homogeneous processor systems and aims in obtaining nice quality scheduling solutions. PBSA algorithm was proposed by Ishfaq Ahmad and Yu-Kwong Kwok [21]. This algorithm is a fresh approach for scheduling according to which a task graph is comprehensively divided that leadsto the coeval generation of sub schedules. The algorithm works well with both periodic and non-periodic graph structures with arbitrary communication and computation costs, handles arbitrarily connected network of target processors.

3.16 Minimum Independent Assignment Cost Underestimate Algorithm (MIACUA)

MIACUA is another static scheduling approach, which was proposed with the objective of obtaining better assignments of distributed tasks. It was proposed by J. B Sinclair [22]. In this algorithm, branch and bound technique is used with the aim of minimizingthe average time and space complexities and also for finding optimum assignments to greater extent. When MIACUA is compared with other algorithms (proposed before MIACUA) of the same class, MIACUA gives better results.

3.17 Heuristic Based Genetic Algorithm (HBGA)

This anotherheuristic based approach of scheduling is usedfor scheduling tasks on homogeneous parallel system. This algorithm works only for scheduling static tasks. Kamaljit Kaur, Amit Chhabra, and Gurvinder Singh proposed HBGA algorithm [10]. This algorithm reduces the completion time and amplifies the throughput of the system.The major drawback of most of the existing heuristics is that they are evaluated with small problems sizes only, thus their scalability is not known. Further, work can be done to overcome such drawbacks.

3.18 Multiprocessor Scheduling of Dependent Tasks to Minimize Makespan and Reliability Cost Using NSGA-II (MSNSGA-II)

It is Non-dominated Genetic Algorithm (NSGA-II) which is used to select the best optimal schedule and also to get sub-optimal solution in minimum time. NSGA-II is non-

preemptive task scheduling algorithm. According to this algorithm each machine will execute a single task at a particular time. There is no dependency among thetasks. In this algorithm Elitism is used to eliminate the problem of loss of good solutions. To get diversity in solution, NSGA-II algorithm [7] uses crowding distance technique.

3.19 A Novel Approach for Task Scheduling in Multiprocessor System (TSB)

TSB algorithm is bounded number of processor (BNP) class of scheduling algorithm. It is queue based scheduling approach, which uses BFS (breath first search) traversal scheduling algorithm [23]. TSB consists of two queues i.e. Ready Task Queue (RTQ) and Not Ready Task Queue (NRTQ). These queues are computed with the help of precedence constrained. Here, RTQ is used as priority attribute for scheduling the tasks on the processor. TSB

algorithm [3] works for finding optimal solution among BNP class of scheduling algorithms.

3.20 A Modified Task Scheduling Algorithm of Task Graph without Communication Time (MTSB)

MTSB algorithm is designed for homogeneous multiprocessor system for task scheduling. Itis modified version of task scheduling algorithm using breath first search i.e. TSB. The basic difference between the two lies in the communication time between the tasks. In TSB algorithm communication time between the tasks is considered while assigning them onto the processor. On the other hand MTSB does not consider communication time. With this modification MTSB algorithm [4] provides shorter scheduling length among all the BNP (bounded number of processor) class of scheduling algorithms.

4. COMPARISION OF VARIOUS FACTORS IN TASK SCHEDULING ALGORITHMS

Table.1 Existing Static Scheduling Algorithm

S. No.	Algorithms	Objectives	Merits	Future Enhancement
1.	HEFT	Minimizes running time	Simplest algorithm of list scheduling	Reducing Time span
2.	MCA	To minimize the makespan	Avoid the complex computation with random variable and applicable to any random distribution	Dynamic scheduling
3.	CPOPA	To minimize the computation cost and time	Gives better results for graphs with higher communication to computation ratio (CCR)	Homogeneous system
4.	HCPPEFT	Improved average SLR, speedup and efficiency	provides a more efficient way to schedulegeneral task graphs	Online Real-time system
5.	LDCP	To give optimum results	Accurately identifies the tasks priorities	To generate high quality task
6.	SNLDD	Provide good Speedup, efficiency, complexity and quality	Generate high quality task schedule	Dynamic
7.	HLFET	Provide good scheduling	Use static b-level level scheduling	Heterogeneous
8.	ISH	Minimize makespan and completion time	Utilizes idle time slots in scheduling	Heterogeneous and dynamic
9.	ETF	Minimize makespan and completion time	Provides good results for large and complex multiprocessor scheduling problem	Heterogeneous
10.	CEFT	To minimize the execution time	Minimum threshold values for schedule length	Use to multiple constrained critical path in subsequent of machine
11.	MQGA	Higher speedup on subtask execution	Provides better solutions with shorter makespan	To make it work for online scheduling
12.	TSA	Finds global optimum solution with shorter makespan time	Avoid local minima and attempts a global minimum solution	Make it work for dynamic scheduling
13.	ACSA	Based on behavior of Ant during searching food	Provides optimized results.	Work can be done to make it work for Homogeneous system

14.	PGSA	High performance, scalability, and fast running time	Provides optimal solutions	Dynamic system
15.	PBSAA	Systematically divides the task graph that guides the concurrent generation of sub schedules	Produces good quality scheduling solutions	Heterogeneous and homogeneous
16.	MIACUA	Optimal assignment of distributed tasks.	Reduces average time and space complexities	Dynamic system
17.	HBGA	Uses genetic algorithm approach	Minimizes the completion time and increase the throughput of the system	Homogeneous
18.	MSNSGA-II	To minimize the makespan and reliability cost	Can be used to solve hard combinatorial optimization problems	Work can be done on dynamic scheduling of both dependent and independent tasks and on maximizing the processor utilization
19.	TSB	To provide an effective tasks order of execution on the processors	Takes less scheduling length among BNP Class of scheduling	Work can be done to improve its scheduling length
20.	MTSB	Communication time is not considered.	Provided shorter scheduling length	Can be applied in heterogeneous environment

HEFT, CPOP, HCPPEFT, CEFT, Monte Carlo, LDCP and SNLDD algorithms works in heterogeneous computing system. When HEFT algorithm is compared withalgorithmfor different values for DAG sizes, it outperforms CPOP algorithm in terms of makespan time and schedule length *ratio*for different scheduling tasks on heterogeneous computing system. Between LDCP and HEFT algorithms, longest dynamic critical path (LDCP) algorithm gives better results than HEFT algorithm. HEFT algorithm also outperforms Monte Carlo Algorithm. Another algorithm named SNLDD generates high quality task schedule in order to achieve high performance in HTDCSs. This algorithm outperforms the LDCP algorithm in terms of efficiency, schedule length and speedup.Random search based algorithms (Genetic algorithm and Tabu search algorithm) yields better results with lesser makespan time than HEFT algorithm.When MPQGA and HEFT algorithms are compared for different task graphs on heterogeneous system than MPQGA algorithm gives better performance than HEFT algorithm. When guided random search algorithms i.e. MPQGA, Tabu search and Ant Colony System (ACS) algorithms are compared, ACS algorithm gives optimized results as compared to other algorithms. When five commonly used list scheduling algorithms i.e. HLEFT, ISH and EFT, are compared on the basis of makespan and total completion time, ISH and EFT gives better solutions as compared to other algorithms. Also, for minimum number of processors required, ETF provides the best trade-off results for most of the problems. Proposed heuristic based genetic algorithms (MIACU, PBSA, PGS, and HBGA) works on reducing the makespan time and generate better solutions by using list scheduling heuristic as a decoding heuristic. Further, among BNP class of scheduling algorithm (HLEFT, ETF, TSB and MTSB), TSB algorithm takes less scheduling length. MTSB algorithm is modified form of TSB algorithm and gives shorter scheduling length as compared to TSB and BNP class of scheduling algorithm. Makespan and reliability cost is minimized by using NSGA-II algorithm.

5. CONCLUSION AND FUTURE SCOPE

In this paper, various static scheduling algorithms for parallel computingsystem have been discussed. Each of them performs well on the basis of certain performance measures. All of them are compared on the basis of several performance evaluation factors like makespan, speed up ratio, computational cost, execution time, efficiency, scheduling length ratio, etc. Ithas been found that HMDCS algorithm works well with the tasks having same computational requirement. Also genetic algorithms perform well in such systems. HTDCS algorithm works well where computational requirement of given tasks are different. In future work can be done in order to make static algorithms more efficient and work can also be done for making some of these algorithms to work for dynamic task scheduling problems for multiprocessor system.

6. ACKNOWLEDGMENT

7. REFERENCES

[1] Kalpana A.M., Avinash W., "Comparative Study of Static Task Scheduling Algorithms for Heterogeneous Systems"International Journal on Computer Science and Engineering (IJCSE), Vol. 5, No. 03, 2013.

[2] Yanyan D and Xiangli Z, "A Synthesized Heuristic Task Scheduling Algorithm", Hindawi Publishing Corporation The Scientific World Journal, Vol. 2014, 2014.

[3] Ranjit R, "A Novel Approach for Task Scheduling in Multiprocessor System", International Journal of Computer Applications, Vol. 44, 2012.

[4] Ranjit R, C.P .Katti and Nidhi R, "A Modified Task Scheduling Algorithm of Task Graph without Communication Time", International Journal of New Computer Architectures and their Applications (IJNCAA) Vol.3, 2013.

[5] Wei Z and Rizos S, "Stochastic DAG scheduling using a Monte Carlo approach", Journal of Parallel Distributed Computing, Vol. 73 , 2013.

[6] Mahfooz A and Ankur K.V "A Comparative Study of Interconnection Network", International Journal of Computer Applications (0975 – 8887) Vol. 127, No. 4, pp.37-43, 2015.

[7] M. Rathna D and A.Anju, "Multiprocessor Scheduling of Dependent Tasks to Minimize Makespan and Reliability Cost Using NSGA-II", International Journal in Foundations of Computer Science & Technology (IJFCST), Vol.4, 2014.

[8] Chitra, Rajaram, and Venkatesh, "Application and comparison of hybrid evolutionary multiobjective optimization algorithms for solving task scheduling problem on heterogeneous systems", Applied soft computing, Vol. 11, 2011.

[9] Nirmeen A. B., Magdy A.K., Mervat M. and Fatma O., "A new algorithm for static task scheduling forheterogeneous distributed computing systems", African Journal of Mathematics and Computer Science Research Vol. 4, 2011.

[10] Kamaljit K, Amit C, and Gurvinder S, "Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system, "International journal of computer science and security, Vol.4, 2010.

[11] Jin S, Schiavone G, Turgut T, "A performance study of multiprocessor task scheduling algorithms", Journal of Supercomputing, Vol.43, 2008.

[12] Sang C K, Sunggu L and Jaegyoon H, "Push-Pull: Deterministic Search-Based DAG Scheduling for HeterogeneousCluster Systems", IEEE Transactions on Parallel and Distributed Systems, Vol. 18, 2007.

[13] Haluk T, Salim H, Min-You Wu, "Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing", IEEE transactions on parallel and Distributed systems, Vol. 13, 2002.

[14] Hou E S, Ansari N, Ren H, "A Genetic Algorithm for Multiprocessor Scheduling",IEEE Transactions on Parallel and Distributed Systems,Vol.5, 1994.

[15] Tarek H, Jan J, "A Near Lower-Bound Complexity Algorithm for Compile-Time Task Scheduling in HeterogeneousComputing Systems", Proceedings of the ISPDC/HeteroPar'04 IEEE, 2004.

[16] Topcuoglu H, Hariri S, Wu MY," Performance-Effective and LowComplexity Task Scheduling for Heterogeneous Computing," IEEE Transactions on Parallel Distributed Systems, Vol.13, 2005.

[17] Daoud MI, Nawwaf K, "A High Performance Algorithm For StaticTask Scheduling In Heterogeneous Distributed Computing Systems", IEEE Transactions on Parallel Distributed Systems, Vol.28, 2007.

[18] Adam T L, Candy K M, Dickson J,"A Comparison of list Scheduling for parallel processing systems", Communication of the ACM, Vol.17, 1974.

[19] Hwang J J, Chow Y C, Anger F D, Lee C Y," Scheduling precedence graph with interprocessor communication time", SIAM Journal of Computing, Vol.18, 1989.

[20] Kruatrachue B, Lewis T G, "Insertion Scheduling Heuristic Algorithm", Technical Report Oregon State University, 87-60-3, 1987.

[21] Ahmad, Ishfaq, and Yu-Kwong K, "On parallelizing the multiprocessor scheduling problem, "Parallel and Distributed Systems, IEEE Transactions, Vol.10, 1999.

[22] Sinclair, J. B., "Efficient computation of optimal assignments for distributed tasks." Journal of Parallel and Distributed Computing, Vol.4, 1987.

[23] Yeshwant K, "Data Structure Using C", BPB Publications 2003.

[24] Kwok, Yu-Kwong, and Ishfaq A, "A Parallel Genetic-Search-Based Algorithm for Scheduling Arbitrary Task Graphs to Multiprocessors", In Proceedings of the 9th IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS'97), Vol. 13, 1997.