

A HYBRID ALGORITHM FOR MULTIPROCESSOR SCHEDULING

R.Vijay Sai, M.Lavanya, B.Srinivasan

School of Computing, SASTRA Deemed To Be University, India.

ABSTRACT

The distributed no-wait flow-shop scheduling problem (DNFSP) is a NP-hard combinatorial optimization problem and plays a significant role in manufacturing and production scheduling. Distributed scheduling has three methods namely exact methods, agent-based algorithms and heuristic-based algorithms. The exact methods cannot be typically used to obtain optimal solutions for distributed scheduling problem. The algorithms based on second and third methods are applicable to numerous real time distributed scheduling problems.

To minimize the make-span of distributed scheduling problem, a new hybrid algorithm is proposed to find an answer for the scheduling problem. This algorithm combines shortest job first and longest job first scheduling algorithm and based on the conclusion, it is proposed whether it is suitable in uniprocessor as well as in multiprocessor environment. Given other parameters like waiting time and turnaround time, this algorithm can achieve efficient solutions.

Keywords : distributed scheduling, make-span, uniprocessor, waiting time, turnaround time

1. INTRODUCTION

An operating system acts as a mediator between user and computer hardware which is declared as a package. An OS allows for various functionalities such as memory management, process management, file management, and device management. Scheduling is used to check the decree of the job which is to be done by the C.P.U of a computer. Scheduling is an important factor in the design of an OS. Scheduling is a scheme which chooses which process should be executed at a particular time. Some of the scheduling algorithms include First come First Served(FCFS), Shortest Job First(SJF), Longest Job First(LJF) and Round Robin(RR). Almost all cpu scheduling algorithms focus on increasing throughput, cpu utilization and belittling turnaround time, response time, waiting time and makespan. Scheduling measures are defined as follows.

CPU utilization	: This is the amount of how much busy the cpu is.
Throughput	: It is the number of processes finished in a particular time period.
Waiting time	: It is the time spent by the process holding in the ready queue.
Turnaround time	: It is the time spent to execute a process.
Response time	: It is the time from entry of a request until the first reply is acquired.
Makespan	: It is the circumstance that appears in scheduling process which shows the time from starting to ending of a particular process.

2. RELATED WORKS

Related works done by the author in the area of flowshop and multiprocessor scheduling are as given here. The DNFSP was formulated for reducing the makespan among multiple processes. Two models were developed for addressing this problem namely Mixed Integer Programming (MIP) and Iterated Cocktail Greedy (ICG) algorithms. Many algorithms have been developed to find out the problem.

In the year 2007, paper [1] proposed that heuristics have been determined to generate optimal solutions in less time. They altered the Nawaz Ensore Ham (NEH) heuristic which resulted in substantial advance of the solution. It deduced its effectiveness from the population based technique. This will outmatch NEH algorithm and HFC heuristic. Yet NEH is the most dependable heuristic for flowshop problems. In the year 2012, paper [2] intended an efficient heuristic algorithm to solve this category of problems. The calculation effects proves the efficiency of this algorithm compared to early ones. This algorithm is named as Heuristic Algorithm Based On Delay (HABD).

In paper [3], processes are not allotted to delay between machines. Here scheduling problem is dealt to reduce makespan and increase performance criteria. In addition several heuristics have been developed other than existing heuristics for this problem.

In paper [4], a Multi Start Simulated Annealing Heuristic is formulated to incur close to optimum solutions. This algorithm plays a predominant part to the scheduling problems in terms of solution quality and computational results.

In paper [5], a constructive heuristic is developed to deal with Distributed Permutation Flowshop Scheduling Problem (DPFSP). Algorithms depending on complexity based on time is found to be equal to NEH. This algorithm is performed on larger data sets and it has been found that it outperforms existent heuristics in terms of solution quality and running time.

In paper [6], an efficient heuristic is applied to reduce the complete flow time in no-wait flowshop scheduling. This concept is depending on the rule that the special feature of a job is given by sum of working times of jobs performed on the machines. This algorithm does not strike the computational times.

In paper [7], a look ahead parallel machines heuristic (LAIPM) has been formulated. Experimentation proves that this heuristic gives better solution than non look-ahead heuristic. In paper [8], a lower bound for no-wait flowshop scheduling problem (NWFSP) is launched. In such, the MIP model is divided into two separate problems with polynomial solutions. This problem uses Lagrangian relaxation method. It is indicated that this method can be used in optimization methods for flowshop problems.

In the year 2016, paper [9] formulated the cloudy based simulated annealing (CSA) algorithm. This algorithm obtains more eminent solutions than another one. A latest mathematical prototype is intended and twin new meta-heuristics are intended for this problem: AIS as a population based method and cloudy based simulated annealing (CSA) as single point approaches are considered. Experimental results show that CSA yielded improved results but AIS produce higher convergence results.

3. PROPOSED TECHNIQUE

To increase the throughput and minimize the makespan for the DNFSP, a new hybrid heuristic algorithm has been proposed in this paper. This algorithm considers computational efficiency as

a factor to improve the existing algorithms for DNFSP. This algorithm will try to minimize the make span when compared to other scheduling algorithms. Jobs are arranged in increasing order of their burst time. Then jobs with shortest burst time and that with longest burst time are combined and executed. This algorithm will prevent system overhead by preventing the movement of tasks among processors and power of the processor is not consumed unnecessarily. This algorithm also takes into consideration Quality of Services like granularity and maximum processing capacity needed for the execution of a particular job. Given other parameters like turnaround time, waiting time and response time, this algorithm will try to give efficient results when compared to other scheduling algorithms for distributed systems.

4. EXPERIMENTAL ANALYSIS

To exemplify the intended algorithm, we illustrate with the sample dataset given below. We assign the task-set (Table-1) : (t1...t10) to processors (p1...p3). Execution time of tasks (e1...e10) are 10,40,29,65,79,70,15,99,47,12. Given task-set is arranged in increasing order of execution times before assigning them to processors. The task-set is grouped by taking tasks one from the first and one from the last. They are Group1=(t1,t10),Group2=(t2,t9),Group3=(t3,t8) and the leftover tasks are assigned to respective processors in order.

Task	Execution Time
T1	10
T2	40
T3	29
T4	65
T5	79
T6	70
T7	15
T8	99
T9	47
T10	12

Table-1 : Task and Execution Time

The execution time of tasks are marked on the gantt chart (Chart-1) and the makespan is calculated. The sample dataset is illustrated below.

10	99	29	
12	79	40	65
15	70	47	

Chart-1 : Gantt chart showing marking of Execution Time

4.1 ALGORITHM

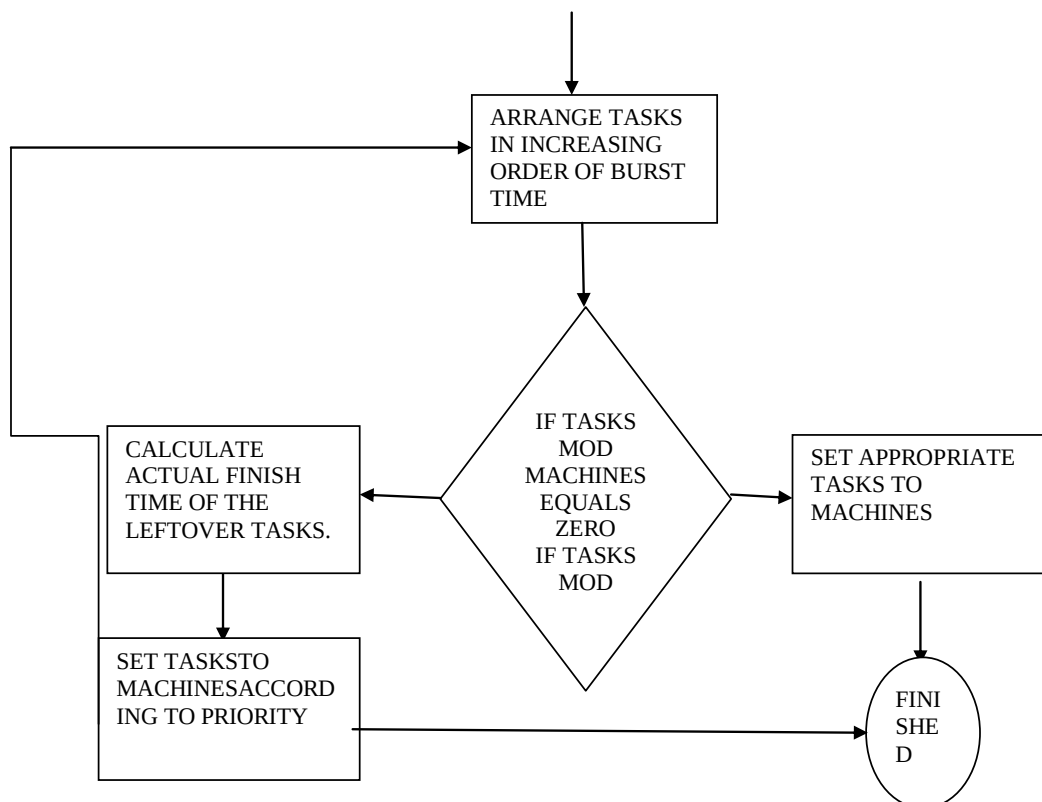
The steps of our proposed hybrid algorithm for multi-processor scheduling are listed as follows, showing set of tasks, set of processors and execution time

```

INPUT:  $t_1 \dots t_n$  = set of tasks
       $p_1 \dots p_n$  = set of processors
       $e_1 \dots e_n$  = execution time
Sort the tasks in higher order
If tasks mod processors equals zero
Assign tasks to processors
Repeat until processors are filled
Else
Calculate execution time of remaining tasks
Assign the tasks according to their priority
End.
  
```

4.2 FLOWCHART

The following flowchart (Flowchart-1) shows the depiction of our proposed algorithm.



Flowchart-1

4.3 COMPARISON

The equivalence and analysis of Shortest Job First (SJF), Longest Job First (LJF) and hybrid method is instanced in the table below (Table-2) and it is absolved that hybrid algorithm minimizes the makespan in a distributed environment. Analysis is performed with 10 tasks and when there are many tasks to be executed, operation of the system increases and processing cost will also be reduced.

Table-2: Comparison analysis

CRITERIA	SJF	LJF	HHA
MAKESPAN	203	203	196

A comparison chart (Chart-2) shows the execution and analysis of SJF, LJF and Hybrid which are illustrated in the figure and it is absolved that Hybrid algorithm minimizes the makespan when compared to SJF and LJF. Analysis is executed with 10 tasks and for vast count of tasks for execution, performance of the system will be increased and processing cost will be reduced considerably.

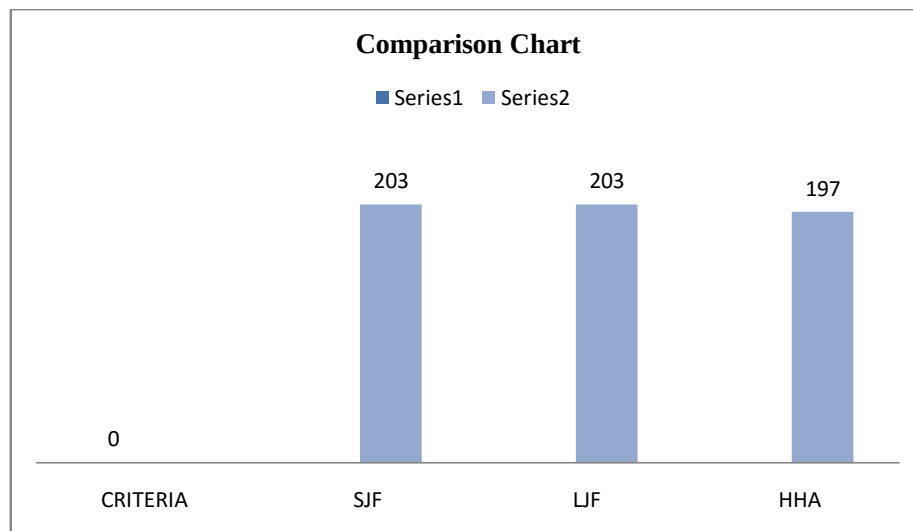


Chart-2 : Comparison Chart

5. CONCLUSION

Distributed systems are widely used in various manufacturing and production industries. These systems require minimization of makespan to run effectively. Dealing with the resources and distributing the jobs to the pertained resource with maximal speed and bandwidth is the necessary standard of scheduling algorithms. In order to minimize the make-span of the DNFSP, the hybrid algorithm is the most suitable heuristic for scheduling the set of tasks which produce the output with maximal speed and usage of resources are also made properly.

REFERENCES

- [1] Chakraborty, Dipak Laha, "An improved heuristic for Permutation flowshop scheduling", *Int.J.Information and Communication Technology*, Vol.1, No.1, 2007.
- [2] GH.M.Komaki, "Solving No-wait flowshop by heuristic algorithm", *International Conference on Industrial Engineering and Operations Management* July-2012.
- [3] Ruben Ruiz, "New heuristics for No-wait Flowshops With Bicriteria of Makespan and Total Lateness", *Department Of Industrial and Management Systems Engineering*.
- [4] Shih-Wei-Lin, Chien-Yi-Huang, Chung-Cheng-Lu, "Minimizing Total Flow Time In Permutation Flowshop Environment", *International Journal Of Innovative Computing, Information And Control*, October 2012.
- [5] Jian Gao, Rong Chen, "A NEH Based heuristic Algorithm For Distributed Flowshop Scheduling Problems", June-2011.
- [6] Sagar.U.Sapkal, Dipak Laha, "A Heuristic For No-wait Flowshop Scheduling", March-2013.
- [7] Yue-Xi, Jaejin Jang, "A Look Ahead Heuristics For Parallel Machines Production In Minimizing Total Weighted Tardiness".
- [8] Morteza Kazemi, "Using Lagrangian Relaxation Approach To generate A Lower Bound For No-Wait Flowshop Problem", *International Journal Of Applied Engineering Research*, 2016.
- [9] Javad Rezaian, "Two New Meta Heuristics For No-Wait Flowshop Scheduling Problem With Capacitated Machines, Mixed Make -To- Order And Make-To-Stock Policy", 2016.
- [10] Shih-wei-Lin, Kuo-Ching-Ying, "Minimizing Makespan For Solving the Distributed No-wait Flowshop Scheduling Problem" *Computers and Industrial Engineering* 99(2016)-202-209.

