World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:9, No:9, 2015

Analysis of Heuristic Based Hybrid Simulated Annealing Algorithm for Multiprocessor Task Scheduling

Supriya Arya, Sunita Dhingra

***Abstract*—**Multiprocessor task scheduling problem for dependent and independent tasks is computationally complex problem. Many methods are proposed to achieve optimal running time. As the multiprocessor task scheduling is NP hard in nature, therefore, many heuristics are proposed which have improved the makespan of the problem. But due to problem specific nature, the heuristic method which provide best results for one problem, might not provide good results for another problem. So, Simulated Annealing which is meta heuristic approach is considered. It can be applied on all types of problems. However, due to many runs, meta heuristic approach takes large computation time. Hence, the hybrid approach is proposed by combining the Duplication Scheduling Heuristic and Simulated Annealing (SA) and the makespan results of Simple Simulated Annealing and Hybrid approach are analyzed.

***Keywords*—**Multiprocessor task scheduling Problem, Makespan, Duplication Scheduling Heuristic, Simulated Annealing, Hybrid Approach.

I. Introduction

MULTIPROCESSOR systems have widely been used in parallel computing. The scheduling is a decision making process and concerned with the allocation of limited resources to tasks overtime with goal of optimizing one or more objectives [5]. The aim of multiprocessor task scheduling is to allocate the tasks to available processors such that the precedence requirements between tasks are satisfied and the overall length of time required to execute the entire program, also referred as the schedule length or makespan, is minimized.

To achieve high performance, task partitioning of an application and tasks scheduling to the processors should be done efficiently. There are dependent tasks and independent tasks for scheduling. Various tasks dependencies of an application are represented by the Directed Acyclic Graph (DAG). DAG scheduling problem is shown to be an NP-complete problem in its general form [1]. Therefore, many heuristics with polynomial-time complexity have been suggested by [2]. The Heuristic scheduling approaches are classified into three categories: priority-based scheduling, cluster based scheduling and duplication based scheduling [6]. It is found that performance of the duplication based algorithms is better than the non-duplication based ones in terms of generating minimum schedule lengths. Therefore, Duplication Scheduling Heuristic (DSH) method proposed by [3] is chosen to solve the scheduling problem. DSH combines the ideas of list scheduling with duplication. In duplication-based scheduling, the task is duplicated and allocated to plural processors. By the simultaneous execution of the same task on more than one processors, the other dependent tasks of this duplicated task can finish their execution simultaneously. With this, their start time can be reduced. However, in the study of heuristic algorithms, [7] and [8] have shown that one heuristic which provides a best result for a problem might not give that much successful result for another problem.

Simulated Annealing (SA), introduced by [4] is an iterative technique which can deal with nonlinear problems. SA is a random-search technique which exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; it forms the basis of an optimization technique for combinatorial and other problems [9]. Its major advantage over other methods is the ability to avoid becoming trapped in local minima. Another advantages of SA are that its robustness, reliability and flexibility. However, disadvantages include the large computation time for many runs and choosing tuneable parameters carefully. The right selection of initial position for SA is still a big question [10].

Majority of research has done for effective scheduling using hybrid approach. Younes et al. [11] investigated the usefulness of SA by using two stages to generate the makespans. In first stage, an initial-heuristic seed sequence is obtained which is improved-upon in the second stage by the SA technique. Empirical results show that the SA technique is able to generate improved Flow Shop with Multiple Processors (FSMP) makespan schedules over the first-stage makespans obtained using extant pure flow-shop heuristics [12]. Laha and Chakraborty Kumar [13] proposed a new hybrid heuristic named PSA which uses simulated annealing in conjunction with the constructive heuristic of [14]. The results of this proposed hybrid approach are much improved than the simple Simulated annealing. It is found that the hybrid approach gives better results than the simple Meta heuristic approaches in less time. Therefore, to get the effective solution, the one possible Hybrid approach can be the combination of DSH as heuristic approach and SA as metaheuristic approach such that the

Supriya Arya is a M.tech degree student at the University Institute of Engineering and Technology,Mahrashi Dayanand University, Rohtak (corresponding author to provide e-mail: supriya.arya20@ gmail.com).

Sunita Dhingra is an assistant professor at the University Institute of Engineering and Technology, Mahrashi Dayanand University, Rohtak.

strengths of one approach overcome the weakness of another approach.

II. Multiprocessor Task Scheduling Problem

The Multiprocessor task scheduling problem can be differentiated on the basis of

a. the communication cost between the parent and child tasks
b. the processing time of the tasks
c. the number of tasks and processors
d. the dependencies between the tasks
e. Topology of the task graph

Multiprocessor Task Scheduling problem for dependent tasks can be represented by the Directed Acyclic Graph (DAG) G = (V, E), where V is set of n nodes and E is a set of *e* directed edges. In DAG, the nodes represent the tasks and the edges represent the dependency relationship as well as the amount of communication between the tasks. The weight of node n_i is called the computation cost and is denoted by $w(n_i)$. The weight of an edge (n_i, n_j) is called the communication cost of an edge and is denoted by $c(n_i, n_j)$. The source node is called the parent node while the sink node is called the child node. A node with no parent is called the entry node while the node with no child is called the exit node.

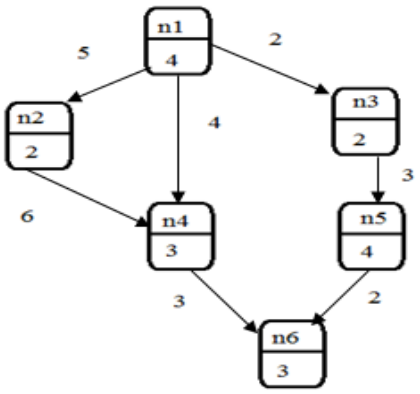

Fig. 1 Directed Acyclic Graph

III. Problem Statement

The goal of multiprocessor task scheduling problem on homogeneous processors is to allocate the tasks to the processors such that the makespan (total length of schedule) is minimum. To minimize the make span, the elementary criteria is the latest execution time among all the jobs, i.e. C_{max}.

A. Assumptions

To solve the multiprocessor task scheduling problem, following assumptions are followed:

a. Static scheduling is considered in which the parameters like no. of tasks, no. of processors, computation cost, communication cost etc. are known in advance.
b. Each processor is available to process the tasks.
c. The Machines never break down.
d. All the tasks and processors are available at time zero.
e. Non- Preemptive scheduling is considered. [5]

IV. Outline of Hybrid Approach

For simulated annealing, the choice of initial seed sequence is a factor which affects the quality of solution. The random generation of initial solution provides the weak results due to a large search space of multiprocessor task scheduling. So, in hybrid approach the initial solution is obtained by simple DSH and this solution becomes the input schedule of SA to find the optimal solution in reasonable amount of time. Outline of hybrid approach is described below:

Step1. Initialization of DSH- The task sequence is sorted on the basis of static b-level values. (static b-level of a node n in DAG is the sum of computation costs of all nodes from n to exit node without considering the communication cost between the nodes).

Step2. Calculating Duplication time slot- Let RT is ready time of particular processor which is the finish time of last node on processor P and ST is start time of particular node on that processor. Therefore, Duplication time slot length on P is (RT- ST).

Step3. Check Parents for Duplication- Let a node has m parents; then, consider the parent which is not scheduled on the P and whose messages for the node has the latest arrival time.

Step4. Duplication- If parent's computation cost is less than Duplication slot, then, Duplicate parents in that slot of P, otherwise, note the start times of node and try another processor.

Step5. Scheduling- Schedule the node on the processor which give the earliest start time among all the processors. Similarly, all the nodes on the basis of task sequence is scheduled on the processor and the finish time of the last task of sequence gives the makespan value.

Step6. Initialization of SA- The schedule of DSH is assigned along with initializations of initial temperature and reanneal interval.

Step7. Neighbourhood generation and acceptance- New schedules are generated at this step. The steps for neighbourhood generation are following:

a. Separate the task sequence (T) and Processor Sequence (P) from the schedule. To get the new task sequence (T') and processor sequence (P'), the neighbourhood generation method is applied individually on both sequences (T and P).
b. The new generated task and processor sequence may not be valid in terms of dependency, so validation algorithm stated by [15] is used to validate the newly generated task sequence (T').
c. The validated task sequence (T'') is combined with the processor sequence.

If the new schedule is better than the current schedule in terms of fitness function or objective function, then, it becomes the next schedule; otherwise, it can still become the next schedule on the basis of probability of acceptance which is computed from acceptance function.

Step8. Cooling Strategy- It is related to systematically

lowering the temperature which is proportional to the cooling rate [5].

Step9. Reannealing- When the algorithm accepts a certain number of new points, the temperature is raised and the search is started again at high temperature.

Step10. Stopping Criteria- The algorithm stops when the maximum number of iterations limit is reached.

V. Results and Discussions

Hybrid approach, the combination of DSH and SA, is implemented in MATLAB for multiprocessor task scheduling problem. The results for simple SA and Hybrid approach is taken for the randomly generated five problems. The Processing time and communication cost of the tasks are randomly generated in the specified range. The five multiprocessor task scheduling problems are given in Table I.

TABLE I
Multiprocessor Task Scheduling Problem

Problems	NT	NP	PT (Range)	CC (Range)
PR1	5	2	1-5	1-10
PR2	10	4	1-5	1-10
PR3	15	4	1-5	1-10
PR4	20	4	1-5	1-10
PR5	25	4	1-5	1-10

NT = No. of tasks, NP = No. of Processors, PT = Processing Time of tasks, CC = Communication cost.

The randomly generated problems with variable Processing times of tasks and communication cost is given as input to the SA and Hybrid algorithm. In SA various parameters are to be fixed which are given in Table II.

TABLE II
Parameters Fixed for SA and Hybrid Algorithm

PROBLEMS	VALUE
AnnealingFcn	Swap function
StallIterLimit	100
PlotFcns	[@saplotf, @saplotbestf]
Plot Interval	50
Initial Temperature	100
Reanneal Interval	150

The randomly generated schedule is given as an input to the simple simulated annealing algorithm. As the initial seed sequence affects the simulated annealing solution, hence, the optimal initial solution for SA is generated by the Duplication Scheduling Heuristic in hybrid approach so that the best solution can be achieved. The Simulated Annealing is a meta heuristic approach which is an approximate method, so the simple SA and Hybrid approach have run five times for each problem in MATLAB to find the average and best makespan value. The results of average and best makespan for all the five problems are shown in Fig. 2.

It is found from the makespan results that the proposed Hybrid approach gives best results as compared to simple SA in terms of average and best makespan for all the five problems. During execution, the time taken by the hybrid approach is less than the simple simulated annealing algorithm.

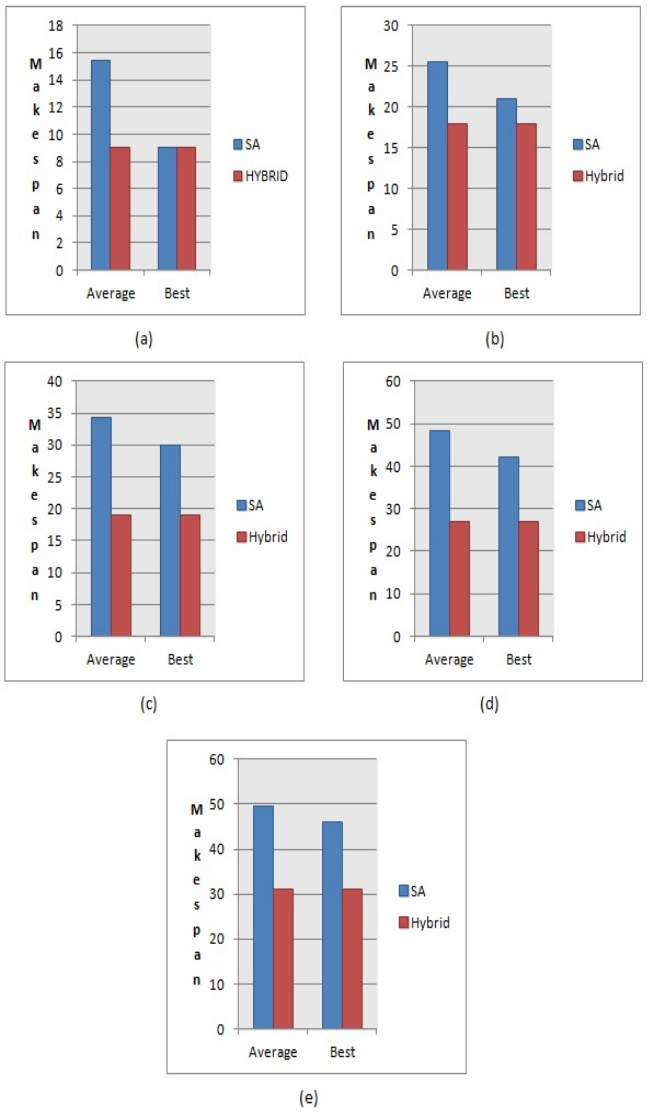

Fig. 2 Average and Best makespan for (a) PR1 (b) PR2 (c) PR3 (d) PR4 (e) PR5 Multiprocessor Task Scheduling Problems

VI. Conclusion

The Multiprocessor task scheduling problem which is NP hard in nature is solved with the aim of minimization of makespan. The initial seed sequence is an important factor for simulated annealing, so, if the optimal solution evaluated from a heuristic approach becomes the initial input for SA, then, SA can give effective solution after improving this initial solution if possible. So, Duplication Scheduling Heuristic is considered to find the optimal solution for SA. To compare the makespan results, the simple SA and hybrid approach (DSH-SA) is tested on the multiprocessor task scheduling problem up to 25 tasks and 4 processors with communication cost and precedence constraints. From the analysis, it is concluded that the hybrid approach provides the effective results in less time as compared to simple Simulated Annealing.

REFERENCES

[1] M. Garey and D. Johnson, "Computers and Intractability, A Guide to the Theory of NP Completeness", W.H. Freeman and Co., 1979.
[2] Yu-Kwong and Ishfaq Ahmad, "Static Scheduling Algorithms for allocating directed task graphs to multiprocessors", ACM Computing Surveys, Vol.31, No. 4, December 1999, pp. 407-427.
[3] Kruatrachue, B. And Lewis, T.G., "Grain size determination for parallel processing", IEEE software 5,1988, pp.23-32.
[4] Kirkpatrick S, Gelatt CD, Vecchi M.P., "Optimization by simulated annealing, Science", Vol.220, 1983, pp. 671–680.
[5] Sunita Dhingra, Satinder Bal Gupta and Ranjit Biswas, "Hybrid GASA for bi-criteria multiprocessor task scheduling with precedence constraints", Computer Applications: An International Journal, Vol.1, No.1, August 2014, pp. 11-21.
[6] Wie-Ming Lin and Qiuyan Gu, "An Efficient Clustering-Based Task Scheduling Algorithm for Parallel Programs with Task Duplication", Journal of Information Science and Engineering 23, 2007, pp. 589-604.
[7] Davis, EW., and Patterson, J. H., "A comparison of heuristic and optimum solutions in resource-constrained project scheduling", Management Science, 21,1975,pp. 718- 722.
[8] Davis, EW., "Project Network Summary Measures and Constrained Resource Scheduling", lIE Transactions, 7, 1975, pp. 132-142.
[9] Yogendra Kumar Soni and Rajesh Bhatt, "Simulated Annealing optimized PID Controller design using ISE, IAE, IATE and MSE error criteria", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Vol. 2, Issue 7, 2013, pp. 2337-2340.
[10] Suchismita Pattanaik, Subhendu Prakash Bhoi, Rakesh Mohanty, "Simulated Annealing Based Placement Algorithms and research challenges: a survey", Journal of Global Search in computer Science, Vol. 3, No. 6, 2012, pp. 33-37.
[11] Younes, N., Santo, D.L., Maria, A., "A Simulated Annealing approach to scheduling in a flow shop with multiple processors", Industrial Engineering Research Conference Proceedings, Banff, Canada,1998.
[12] Hooda N, Dhingra A. K, "Flow Shop Scheduling using Simulated Annealing: A review", International Journal of Applied Engineering Research, Dindigul, Vol. 2, No.1, 2011, pp.234-249.
[13] Laha, D., Chakraborty, U.K., "An efficient hybrid heuristic for makespan minimization in permutation flow shop scheduling", International Journal of Advance Manufacturing Technology, 2008, pp.559-569.
[14] Nawaz, M., Enscore, E., Ham, I., "A heuristic for the m-machine n-job flow shop sequencing problem", Omega, 11, 1983, pp. 91–95.
[15] Bonyadi, M. R. and Moghaddam, M. E., "A bipartite genetic algorithm for multi-processor task scheduling", International Journal of Parallel Programming, Vol. 37, No. 5, 2009, pp. 462- 487.