International Journal of Latest Trends in Engineering and Technology
Vol.(20)Issue(2), pp.027-031
DOI: http://dx.doi.org/10.21172/1.202.05
e-ISSN:2278-621X

EVALUATION OF LIST SCHEDULING AND CLUSTERING-BASED HYBRID SIMULATED ANNEALING ALGORITHMS FOR TASK SCHEDULING IN MULTIPROCESSOR SYSTEMS

Sunita Dhingra[1]

Abstract- Task scheduling with precedence constraints on a multiprocessor system to optimize various performance metrics is a significant challenge in parallel computing. Determining an optimal schedule for task scheduling problem in multiprocessor system is known to be NP-hard and a variety of heuristics and metaheuristics approaches have already been developed. Heuristic based scheduling algorithms are efficient for specific problems and do not perform well with increasing problem complexity. Metaheuristics based algorithms find optimal schedule at the expense of more computing efforts. Thus, hybrid algorithms that combine the strengths of heuristics and metaheuristics are essential in today's problem-solving landscape. Present work compares two hybrid simulated annealing algorithms based on HLFET list scheduling and Linear clustering heuristics for bi-criteria multiprocessor task scheduling and reveals that under minimal communication delays, both algorithms exhibit comparable performance. However, as communication delay increases, the linear clustering-based hybrid simulated annealing algorithm outperforms the HLFET list scheduling-based hybrid algorithm.

Keywords- Scheduling, Makespan, Total Completion Time, Simulated Annealing.

I. INTRODUCTION

Scheduling involves allocating a finite set of resources in a way that improves the system's overall performance. Effectively scheduling computationally intensive programs is crucial and complicated for achieving enhanced performance in parallel computing. Typically, a large program is broken down into subtasks with dependencies, which represent the precedence constraints. These constraints ensure that a task begins only when all prior tasks are completed. The objective of a task scheduling algorithm is to assign all tasks to a specified number of processors while adhering to the given constraints. This aims to minimise various performance metrics, thereby maximizing system throughput and utilization. Finding an optimal solution for the task scheduling problem in multiprocessor system is NP-hard [4], and numerous heuristics, metaheuristics, and hybrid approaches have been developed by researchers to address these complex problems. In practice, a heuristic can provide a good initial solution within a reasonable computational time. But, heuristic-based approaches tend to have a greedy nature, which can lead to unreliable results, especially as the complexity of scheduling problems increases. Therefore, researchers often turn to metaheuristic approaches such as Tabu Search, Genetic Algorithm(GA), Simulated Annealing(SA) and others to achieve higher-quality schedules. Most metaheuristic methods have shown superior performance over conventional heuristic-based scheduling algorithms, but with increased time and computational requirements [10]. So, hybridizing metaheuristics with heuristics is often considered the next step to further enhance solution quality.

Heuristics have been demonstrated to provide effective initial solutions for metaheuristics, thereby improving the efficiency of multiprocessor systems. Therefore, in the present work, two hybrid algorithms based on simulated annealing have been implemented: one combining simulated annealing with HLFET list scheduling, and the other combining simulated annealing with linear clustering. These hybrids are compared to analyze their effectiveness in addressing the multiprocessor task scheduling problem with communication delay.

[1]*Department of Computer Science & Engineering, University Institute of Engineering & Technology,M. D University Rohtak-124001 Haryana INDIA*

II. Related Work

Simulated annealing (SA) is a meta-heuristic technique developed by Kirkpatrick et al. [1] for addressing optimization problems with combinatorial aspects. Kwok and Ahmad [4] conducted a thorough review of deterministic static scheduling algorithms. They further evaluated various scheduling strategies using a nine-task problem as a benchmark. Sivanandam et al. [5] introduced a hybrid approach PSO/SA, aims to optimize the static task allocation in diverse distributed computing systems with a focus on optimizing costs. The researchers implemented and tested their hybrid approach using various PSO algorithms combined with simulated annealing. Through experiments on benchmark problems, they demonstrated the effectiveness and efficiency of their proposed method in achieving near-optimal solutions. Lin, Wei-Ming, and Gu, Qiuyan[6] introduced a systematic approach for finding an optimal schedule length when task duplication is permitted. Their method, which has a reasonable time complexity, significantly improves schedule length compared to faster techniques, while delivering performance on par with more complex methods. Bonyadi and Moghaddam[7] introduced the Bipartite Genetic Algorithm (BGA) aimed at minimizing makespan in multiprocessor task scheduling. Their study demonstrated that BGA outperforms existing heuristics and metaheuristics(GA) in terms iteration count, best achieved makespan and average makespan. Mishra et al. [8] introduced a clustering algorithm utilizing the Computation-Communication-Load (CCLoad) approach. Their research demonstrated that this algorithm outperforms a generic version of Sarkar's algorithm in terms of efficiency or effectiveness. Dhingra et al.[9] conducted a comparative evaluation of five frequently utilized list scheduling heuristics for homogeneous multiprocessors. They evaluated these heuristics based on the criteria of total completion time and makespan. Their study concluded that among the evaluated heuristics, the Earliest Time First Algorithm and Insertion Scheduling Heuristic consistently provided the optimal results in terms of balancing the trade-off between total completion time and makespan of the schedule. H. Wang and O. Sinnen[11] conducted a comparative analysis of List based scheduling and Cluster based scheduling algorithms. They found that Cluster schedulers, which focus on reducing communication, generally perform better with high CCR , while list scheduling excels at lower CCRs. However, In high CCR scenarios, cluster scheduling demonstrates a notable benefit over list scheduling, especially in fork-and-join structures that pose challenges in tree-based scenarios. For scheduling across an unlimited number of processors, clustering algorithms proved to be more advantageous than List scheduling. Orr J. and Baruah S. [12] developed Scheduling algorithms for elastic tasks within both partitioned and global scheduling frameworks. Through extensive simulations, they found that partitioned scheduling outperformed global scheduling and recommended its use. B. Acharya and S. Panda [13] introduced an effective method for tackling Multiprocessor Machine Scheduling Problems (MMSPs) using a modified salp swarm algorithm (MSSA). They evaluated MSSA under real-time scheduling constraints across different processor counts. Their approach demonstrated superior performance in both small and large-scale problem scenarios.

III. Problem Statement

The present work addresses task scheduling problem with precedence constraints in multiprocessor system utilizing homogeneous processors, aiming to minimize a bi-criteria objective function (F). This function represents the weighted total of the makespan and the total completion time and is formulated as[10]:

$$F = Min[\alpha C_{max} + (1-\alpha)\sum_{i=1}^{n} C_i]$$

Here, C_i denotes the completion time of the i_{th} task in the schedule, C_{max} represents the makespan and α is the weight coefficient reflecting the preference of the performance metrics to be minimized.

This study uses a deterministic model where the precedence constraints, count of processors, task execution times and task relationships are all predefined. Additionally, the tasks are non-preemptive, meaning that once a task starts, it must complete before any new tasks can begin.

IV. Proposed Methodology

In this study, two hybrid simulated annealing algorithms have been implemented and analyzed. One algorithm is based on the HLFET list scheduling heuristic, while the other employs linear clustering. Scheduling heuristics are mainly of three types : Clustering based, List Scheduling, and duplication based heuristics[4]. List scheduling heuristics are commonly employed for task scheduling problems to assign priorities to tasks. As soon as a processor is available, it is allocated the top priority task from the list, which is then removed. If multiple tasks have the same priority, the selection of the task can be either random or based on specific rules. Typically, priorities are assigned based on characteristics such as t-level (top level), ALAP (As-Late-As-Possible) start time, b-level (bottom level) and static level (SL). Various commonly used list scheduling heuristics are : HLFET,ISH, ETF and MCP [4]. Clustering based heuristics are multiphase where in first phase, tasks that need to be scheduled together on the same processor due to high inter-processor communication costs are grouped and designated to a common cluster[4,11]. Thereby, Clusters are merged to match the number of available processors, assigned to them, and finally, task ordering is determined to create a schedule[11]. Linear clustering and dominant sequence clustering are widely used multiprocessor task scheduling heuristics.

Simulated Annealing (SA) is a widely used local search based metaheuristic based on the physical process of cooling metals[1]. It has proven effective in optimizing scheduling problems. In classical SA, the process starts with a randomly generated initial sequence and seeks optimal solutions through a neighborhood search mechanism. Since the choice of the initial sequence significantly impacts the solution quality, heuristics are often employed to generate a better initial sequence. By integrating heuristics with metaheuristics, hybrid algorithms are created, which can yield improved results within a reasonable time frame[10].

In the present work, Highest Level First with Estimated Time (HLFET) list scheduling heuristic has been used to generate the seed sequence for Simulated Annealing, resulting in the HLFET-SA hybrid algorithm. HLFET algorithm is the most conventional list scheduling heuristic that assigns the priority to each node on the basis of static b-level, which represents the longest path length from a node to an exit node[4]. Additionally, a Linear Clustering-based heuristic was employed to generate the seed sequence for Simulated Annealing, leading to the LC-SA hybrid algorithm. Linear Clustering is a straightforward clustering algorithm that operates by determining the set of nodes constituting the critical path and assigning them to a cluster. It then removes these nodes and their incident edges from further consideration [4,11]. This process is repeated iteratively, with each iteration finding the longest path among the remaining unexamined nodes in the task graph and continues until all nodes have been assigned to a cluster.

V. Results and Discussion

This study compares two hybrid algorithms, HLFET-SA and LC-SA, to solve the task scheduling problem in multiprocessor systems with homogeneous processors. These algorithms were implemented in a MATLAB environment and evaluated their performance using a weighted total of makespan(MS) and total completion time(TCT). A comparative analysis has been conducted using several established problems and their associated weight coefficients (α)[10] as detailed in Table-1. Parameters fixed for Simulated annealing in both hybrid algorithms has been represented in Table-2. Since simulated annealing is an approximate method, each algorithm was run five times per instance to calculate the final average.

TABLE-1

CONVENTIONAL MULTIPROCESSOR TASK SCHEDULING PROBLEMS

Problems	Task Count	Processor Count	Communication cost(fixed)	Reference	Weight Cofficient
T14_1	14	4	20	Tsuchiya et al.[3]	0.89
T14_2	14	4	80	Tsuchiya et al.[3]	0.89
T16_1	16	4	40	Wu and Gajski[2]	0.90
T16_2	16	4	160	Wu and Gajski[2]	0.90

TABLE -2

PARAMETERS FIXED FOR SA IN HLFET-SA AND LC-SA

Parameter	Value
Maximum No. of iterations	150
Initial Temperature	240
Re-anneal Interval	150
Temperature Function	Fast
Move Function	Block

The comparative analysis was conducted by calculating the performance index (PI) [10] as:-

$$Performance\ Index(PI)(\%) = \left[1 - \frac{Algorithm_{soultion} - Best_{solution}}{Best_{solution}}\right] \times 100$$

$Algorithm_{solution}$ refers to the average solution achieved for a specific task problem across distinct algorithms, while $Best_{Solution}$ denotes the best available solution or the best result among all algorithms for the problem across all runs. A Performance Index (PI) closer to 100% indicates better results.

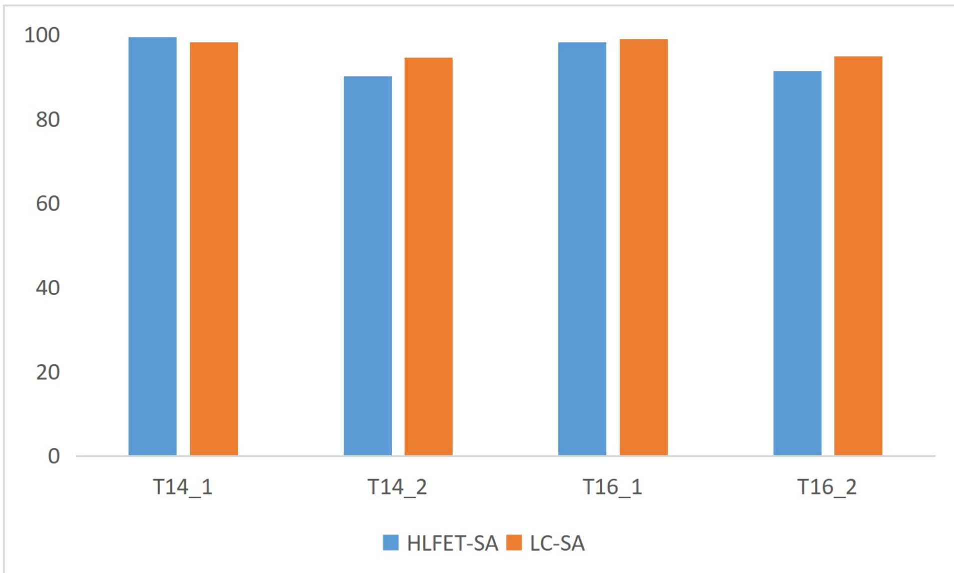

Fig. 1 Performance Index(%) of HLFET-SA and LC-SA in terms of Makespan

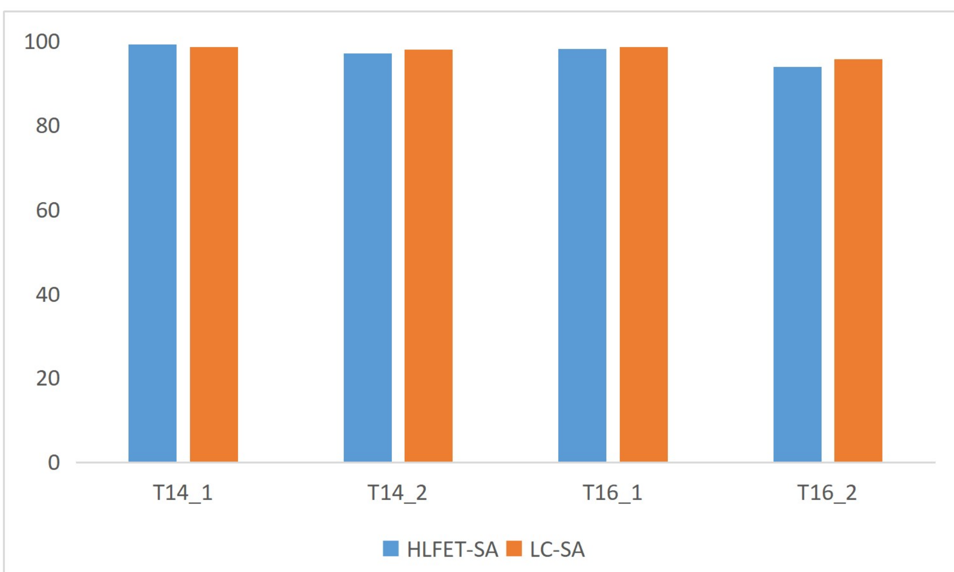

Fig. 2 Performance Index (%) of HLFET-SA and LC-SA in terms of Total Completion Time

Fig. 1 and Fig. 2 display the performance indices of both hybrid algorithms for makespan and total completion time, respectively. These figures clearly indicate that when communication delays are minimal(T14_1 & T16_1), both algorithms perform almost equally. However, as communication delays increase(T14_2 & T16_2), the clustering-based hybrid simulated annealing algorithm (LC-SA) outperforms the list scheduling-based hybrid simulated annealing algorithm (HLFET-SA).

VI. CONCLUSION

The list scheduling heuristic (HLFET) based hybrid simulated annealing algorithm (HLFET-SA) has been compared with hybrid Simulated Annealing Algorithm based on linear clustering (LC-SA) for the multiprocessor task scheduling problem with communication delay, focusing on the weighted total for makespan and total completion time. Standard benchmark problems out of the literature were applied for this comparative analysis. The outcomes show that both hybrid algorithms function efficiently and perform comparably well when communication costs are low. However, as communication costs rise, both algorithms experience a slight degradation in performance. Despite this, a comparison reveals that the linear clustering-based hybrid simulated annealing algorithm (LC-SA) performs better than hybrid simulated annealing algorithm based on list scheduling (HLFET-SA) for multiprocessor scheduling problems with higher communication cost. Future work could involve hybridizing additional list scheduling and clustering-based heuristics with metaheuristics, and evaluating their performance on problems with larger and variable communication delays

REFERENCES

[1] Kirkpatrick S, Gelatt CD, Vecchi M.P., "Optimization by simulated annealing",Science, Vol. 220, pp. 671–680, 1983.

[2] Wu, M.Y., and Gajski, D.D., "Hypertool A programming aid for message-passing systems", IEEE Transactions on Parallel and Distributing System, Vol. 1, No. 3, pp. 330–343, 1990.

[3] Tsuchiya, T., Osada, T., and Kikuno, T., "Genetics-based multiprocessor scheduling using task duplication", Journal of Microprocessors and Microsystems, Vol. 22, Vol. 3–4, pp. 197–207, 1998.

[4] Kwok, Y.K., and Ahmad, I., " Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors", ACM computing surveys, Vol. 31, No. 4, pp. 406-471, 1999.

[5] Sivanandam, S.N.., Visalakshi, ,P., and Bhuvaneswari, A., "Multiprocessor Scheduling Using Hybrid Particle Swarm Optimization with Dynamically Varying Inertia ",International Journal of Computer Science & Applications, Vol. 4, No.3, pp 95-106, 2007.

[6] Lin, Wei-Ming & Gu, Qiuyan.," An Efficient Clustering-Based Task Scheduling Algorithm for Parallel Programs with Task Duplication ", Journal of Information Science and Engineering, Vol. 23, No.2, pp 589-604, 2007.

[7] 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, pp. 462- 487, 2009.

[8] Mishra, Pramod & Mishra, Kamal & Mishra, Abhishek.," A Clustering Heuristic for Multiprocessor Environments Using Computation and Communication Loads of Modules", International Journal of Computer Science & Information Technology, Vol. 2, No. 5, pp. 170-182, 2010.

[9] Dhingra, S., Gupta, S.B. and Biswas, R., "Comparative analysis of heuristics for multiprocessor task scheduling problem with homogeneous processors", Advances in Applied Science Research, Vol. 5, No. 3, pp. 280 – 285, 2014.

[10] Dhingra, S., Dhingra A.K., Gupta, S.B. and Biswas, R,"Hybrid Genetic Algorithm for Bi-Criteria Multiprocessor Task Scheduling with Communication Delay", Advanced Computational Intelligence: An International Journal (ACII), Vol. 3, No. 4, pp 19-29, 2016.

[11] H.Wang and O. Sinnen , "List-Scheduling versus Cluster-Scheduling" IEEE Transactions on Parallel & Distributed Systems, Vol. 29, No. 08, pp. 1736-1749, 2018.

[12] Orr J. and Baruah S.," Multiprocessor scheduling of elastic tasks," In Proceedings of the 27th International Conference on Real-Time Networks and Systems (RTNS '19). Association for Computing Machinery, New York, NY, USA, 133–142, 2019.

[13] B. Acharya and S. Panda, "Modified SSA for Solving Multiprocessor Scheduling Problems," 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, pp. 1075-1080, 2021.