*Vidya Sagar Ponnam Int. Journal of Engineering Research and Applications* www.ijera.com
*ISSN: 2248-9622, Vol. 5, Issue 7, (Part - 3) July 2015, pp.23-29*

RESEARCH ARTICLE OPEN ACCESS

# An Improved Parallel Activity scheduling algorithm for large datasets

Vidya Sagar Ponnam [#1], Dr.N.Geethanjali[#2]
[1] Research Scholar, Dept. of Computer Science & Technology,Sri Krishnadevaraya University, Ananthapuram
[2] Associate Professor & Head Dept. of Computer Science & Technology,Sri Krishnadevaraya University,Ananthapuram

**ABSTRACT**
Parallel processing is capable of executing a large number of tasks on a multiprocessor at the same time period, and it is also one of the emerging concepts. Complex and computational problems can be resolved in an efficient way with the help of parallel processing. The parallel processing system can be divided into two categories depending on the nature of tasks such are homogenous parallel system and the heterogeneous parallel processing system. In the homogeneous environment, the number of processors required for executing different tasks is similar in capacity. In case of heterogeneous environments, tasks are allocated to various processors with different capacity and speed. The main objective of parallel processing is to optimize the execution speed and to shorten the duration of task execution with independent of environment. In this proposed work, an optimized parallel project selection method was implemented to find the optimal resource utilization and project scheduling. The execution speeds of the task increases and the overall average execution time of the task decreases by allocating different tasks to various processors with the task scheduling algorithm.

## I. INTRODUCTION

The Task Scheduling problem implemented in the parallel multiprocessor system can be classified into different classes depend on the criteria of the multiprocessor system, tasks to be scheduled and the availability of resources, this paper mainly focussed on solving the parallel deterministic scheduling problem . A deterministic scheduling issue [1,2] is one in which all data about the tasks and its relation such as precedence relation and execution time are known in advance.The task should be non–preemptive , which means task execution must be fully done before another task takes control of the processor with homogeneous speeds or processing capabilities.

The computational issue in the traditional approach was mainly due to multiprocessor task scheduling. Powerful computing was achieved by executing a real time application, specifically a single processor is not enough to execute all activities. In order to determine when and on which processor a given task should execute was decided by an optimal algorithm in computing environment. A task can be divided into a group of subtasks and shown as DAG to schedule multiple tasks in parallel multiprocessor system. The mapping meta-tasks on a machine are shown to be NP-complete. By using heuristic approach the NP-complete problem can be solved. The processing time and requirements of all applications are assumed to be stochastic. Any given problem is to be scheduled in a given multiprocessor system using multiprocessor scheduling problem to minimize program's execution time and the last task must be completed as soon as possible. For constrained optimization, genetic algorithm is one of the approaches which is widely used to schedule tasks. Execution of genetic algorithm can be optimized with the knowledge implementation of the scheduling problem. In this traditional approach, the challenge of execution, completion time and the precedence order in the parallel processing system were resolved by using the concept of Top –level or bottom-level.

The multiprocessor computing contains a set of m homogenous processor P={Pi:=1,2,3….m}

They are completely linked with each other via identical links .Figure1 shows a fully linked three parallel system with identical link.

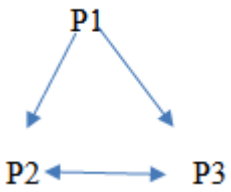


FIG 1: DAG parallel processor

In the paper [1], a directed acyclic graph with each node specifies a task .The main goal is to link each task to a set of m processors . Each task is associated with a weight . Each directed edge

shows the communication between two tasks. The successor task could not be executed until all its procedors have been executed and their results are available on the processor on which the task is scheduled to execute. If the task at the processor is not ready for execution, then the processor remains constant until the task state is ready. The weight becomes associated with the edge becomes null [7,12] if both the tasks are scheduled to same processor[1].

It is key to apply an enhanced scheduling algorithm for task sequencing or proper resource allocation on multiprocessors in order to reduce the computing power and to increase scheduling performance in a parallel computing system.For the given parallel computing system modeled by an edge weighted and acyclic graph, this system assumes a multi-scheduling with a least turn around time without considering task constraints [1],[3],[4]. Only for a few specified cases [4],[5],[1],[9],[3], if any problem exists it was solved in polynomial time complexity. Optimal solutions are needed to use to minimize search space and time taken to schedule the multiple tasks . The majority of the traditional approaches is still required to execute under resource constraint systems and not capable in normal task scheduling environments. Even moderate scheduling tasks cannot be resolved within the specified execution time by using these techniques[2-5]. These heuristics is to optimize the time and space complexity.The considerable weakness of this heuristic is that they usually apply a greedy mechanism on the scheduling factors such as the structure of the input task graphs and the number of available target processors. Enhancing performance of a heuristically increases its complexity . In addition, these approaches are not scalable in their solution quality or running time for large tasks sets[4].

**Problem Statement**

In any static scheduling process, a parallel system can be executed using the task directed acyclic graph(DAG) . A node in DAG shows a project which is a set of interrelated tasks that must execute sequentially without preempting its dependent tasks. The edges in the DAG represent communication constraints and dependency order among the nodes.The communication & computation ratio of a parallel system measures average communication cost to the average computation cost on a given parallel system . Directed edges represent the task dependencies as well as the duration of task completion, were commonly applicable in static scheduling of a parallel program task on multiprocessors. The static scheduling used to minimize the total project completion time with a limited number of resources and fixed task duration. Static scheduling doesn't consider dynamic task allocation as well as dynamic resource scheduling. A task precedence graph or DAG works accurately for most static and limited parallel task scheduling applications since it depends on the resource and duration dependencies between tasks.

## III RELATED WORK

In multi resource constrained project scheduling, each task may require a set of activities or a set of successive operations. For a given activity,several resources may execute in a parallel manner, which means the task can consider any one of the available resources for processing.These issues are often known as machine scheduling problems[3]. In SM-RCPSP, multiple task specifies a minimum and maximum time delay between the project activities. The minimum time delay represents that a task can only start or finish when the predecessor task has already finished in a given time.The maximum time delay represents that a task can only start or finish at the certain time beyond the end of the next successor project. Let $st_i$ denotes the task start time i and $lt_{ij}$ denotes the last delay time between the i and j tasks, and this can be represented in the constraint form as ,

$$st_i + lt_{ij} < st_j$$

Let $\phi$ represents the set of tasks with precedence relationship, the pair (i,j) is in $\phi$ if either activity I must complete before activity j can start. The minimum delay with task preemption can be formulated as follows:

$$\text{Min} \quad Fin_t$$

S.t

$$st_i > 0$$

$$lt_{ij} > 0$$

$$\sum_{s_t} r_{tk} < R$$

As the number of tasks in the project increases and thus the complexity of the sequential ordering, the need for scheduling and organized planning. This requires further increases the large number of activities in terms of standardizing the nature of the work. So, finding a relevant, feasible solution is a challenging task within the multitask project scheduling.

The conventional scheduling methods such as program evaluation and review technique and critical path method are not enough for multitasking, because they generate infinite schedules which cannot take resource constraints into consideration. Traditional methods generate a large set of feasible and infeasible solutions.

**Limitations in Classical techniques:**
The classical optimization techniques are useful in finding the optimum solution or constrained minima or maxima with differentiation function. These analytical problems, make use of differential calculus in locating the optimum solution. Classical optimization approaches can be handled by the single variable functions, multivariate functions with no constraints.

## IV. PROPOSED APPROACH

Multi-Project optimization is the process of simulating the input tasks in a mathematical experiment or process to find the minimum constraint and the maximum constraint as shown in Fig 2. Best solution implies that there is more than one feasible solution and all the solutions are not same. The objective function or cost function is the mathematical model of the given problem to be resolved. Each constraint should be formulated with varying parameters.The main goal of the optimization process is to find the optimal value which minimizes or maximizes the objective function.

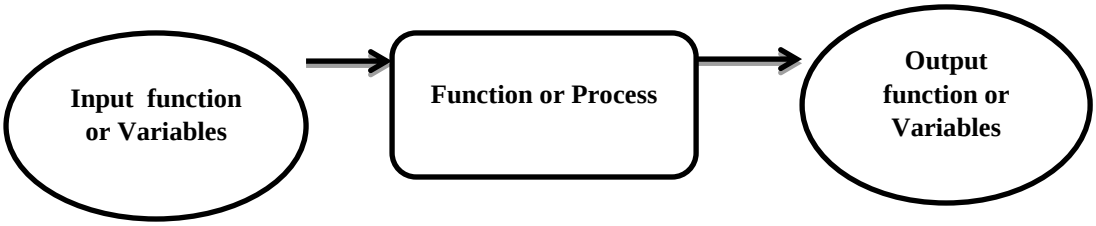


Fig 2: Optimization Process

Project scheduling and selection problems have been received significant role for business organizations, including private, public sectors and R&D projects. Proposed approach considers the follow framework to handle multiple project scheduling as shown in Fig3. Each project is partitioned to find the relevant, relational project for efficiently process the multiple projects.

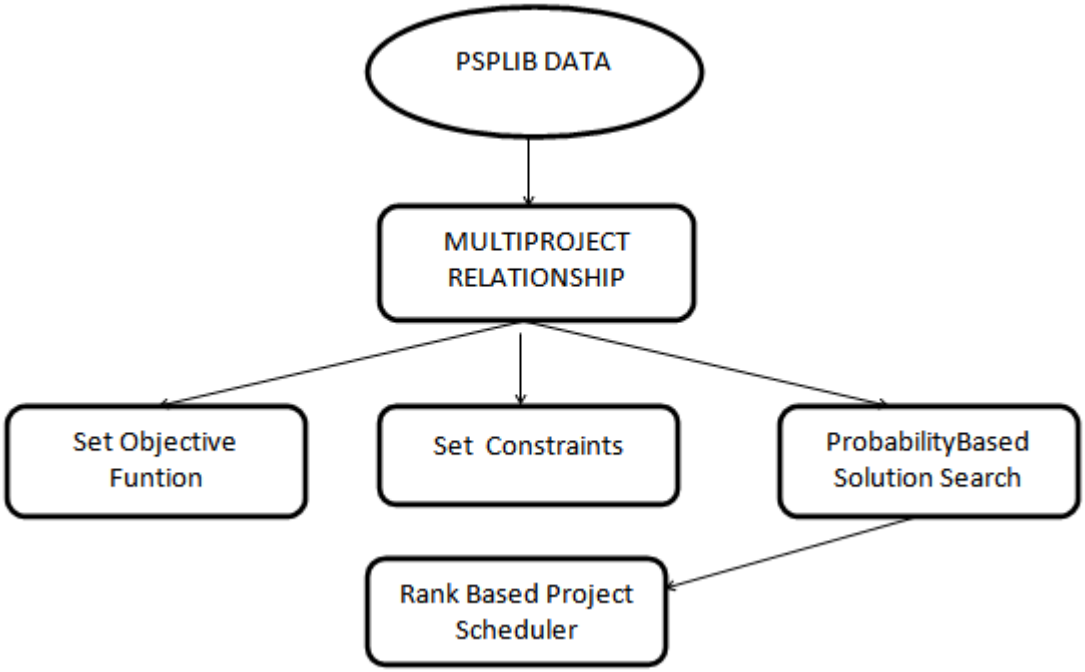


Fig 3: Overall Proposed framework

**Algorithm Steps:**

**Step 1:** Assign overall project optimization function as objective function.

Max $z = F(P_i, st_i, dt_i)$

**Step 2**: Assign constraints with the domain list.

Let $c_1, c_2, c_3, c_4 ... c_n$ be the n constraints,

$c_i > 0$ , i=1, 2… n.

**Step 3**: for each project $p_i$ do

For each activity $A_j$ do

Find the prior probability of each activity in the project.

$$p(\mathrm{a}_j / p_i) = (1 + e^{pro(a_j \cap p_i)/ pro(a_j)}) / \mathrm{pro}(\mathrm{a}_j)$$

Add( $p_i$ , $A_j$ )= $p(\mathrm{a}_j / p_i)$ ;

End for

End for

**Step 4:** // find the relational projects to schedule first.

Find the resource list and its duration.

$$\phi(R_i, T(st_i, dt_i)) := getresources(\mathrm{P}_i)$$

Find the activities list within the lower bound and upper bound of the time duration.

$$ub = dt_i + \mu_{list(p_i, A_j)} \times list(p_i, A_j)$$

$$lb = dt_i - \mu_{list(p_i, A_j)} \times list(p_i, A_j)$$

**Step 5:** Check the constraints for resource bounds and time bounds.

$$st_i + dt_i \geq FinishTime$$

$$lb \leq p_i \leq ub; i = 1....n$$

**Step 6:** Finish the project schedule using PSO optimizer using the polynomial initial feasible solution.

$$\partial p^2{}_i(x) / \partial x + \partial p_i(x) / \partial x + R \leq Fin_t$$

$$\partial p^2{}_i(x) / \partial x + \partial p_i(x) / \partial x + R \leq Fin_t$$

## Experimental Results

The proposed system implemented on PSPLIB data set with a large number of project activities.From the experimental results it has been found that performance of the proposed work is better with the initial feasible solution from pre-selected solutions.Proposed approach efficiently minimizes the search space due to the correlation between the project activities.This system helps to find the local minima of the time duration for resource allocation.The computational time required to find the best solution is significantly minimized by using the optimization function.Efficient performance of the proposed algorithm is checked accordingly by using the objective function,makespan and project constraints.

**Sample Data:**

PRECEDENCE RELATIONS:

| jobnr. | #modes | #successors | successors |
|---|---|---|---|
| 1 | 1 | 3 | 2 3 4 |
| 2 | 3 | 2 | 8 15 |
| 3 | 3 | 3 | 6 8 10 |
| 4 | 3 | 3 | 5 12 14 |
| 5 | 3 | 3 | 7 10 11 |

REQUESTS/DURATIONS:

jobnr. mode duration R 1 R 2 N 1 N 2

---

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 8 | 0 | 8 | 0 |
| | 2 | 1 | 7 | 0 | 0 | 6 |
| | 3 | 8 | 0 | 7 | 8 | 0 |
| 3 | 1 | 3 | 10 | 0 | 0 | 5 |
| | 2 | 7 | 0 | 9 | 0 | 4 |
| | 3 | 10 | 6 | 0 | 0 | 4 |
| 4 | 1 | 3 | 9 | 0 | 7 | 0 |
| | 2 | 6 | 0 | 8 | 4 | 0 |
| | 3 | 9 | 5 | 0 | 2 | 0 |
| 5 | 1 | 1 | 3 | 0 | 10 | 0 |
| | 2 | 2 | 2 | 0 | 7 | 0 |
| | 3 | 10 | 0 | 2 | 3 | 0 |

RESOURCEAVAILABILITIES:

| R 1 | R 2 | N 1 | N 2 |
|---|---|---|---|
| 11 | 11 | 54 | 62 |

**Results:**

Project #1
Probability of Relevance :0.45566122600653036
Correlation with activities :0.7119374883170021
Project #2
Probability of Relevance :0.34133110991917504
Correlation with activities :0.8030370471410827
Project #3
Probability of Relevance :0.4302457851958701
Correlation with activities :0.0656144686559309
Project #4
Probability of Relevance :0.3074715237080108
Correlation with activities :0.49277278818441794
Project #5
Probability of Relevance :0.7674237298256747
Correlation with activities :0.36097951560743835
Project #6
Probability of Relevance :0.6028646726535574
Correlation with activities :0.5791952887637609
Project #7
Probability of Relevance :0.37444237674881575
Correlation with activities :0.3983739029005475
Project #8
Probability of Relevance :0.5388288551175188
Correlation with activities :0.5076945473129358
Project #9
Probability of Relevance :0.68685757749859
Correlation with activities :0.4966037410033388

| JOBS | Resources | PSO-ACO | Event-ACO | ProposedAvgTime(secs) |
|---|---|---|---|---|
| 50 | 12 | 217 | 311 | 142 |
| 100 | 27 | 388 | 356 | 215 |
| 150 | 31 | 521 | 519 | 424 |
| 200 | 43 | 785 | 712 | 644 |
| 250 | 53 | 1233 | 1089 | 789 |

Table 1: Job performance with traditional algorithms

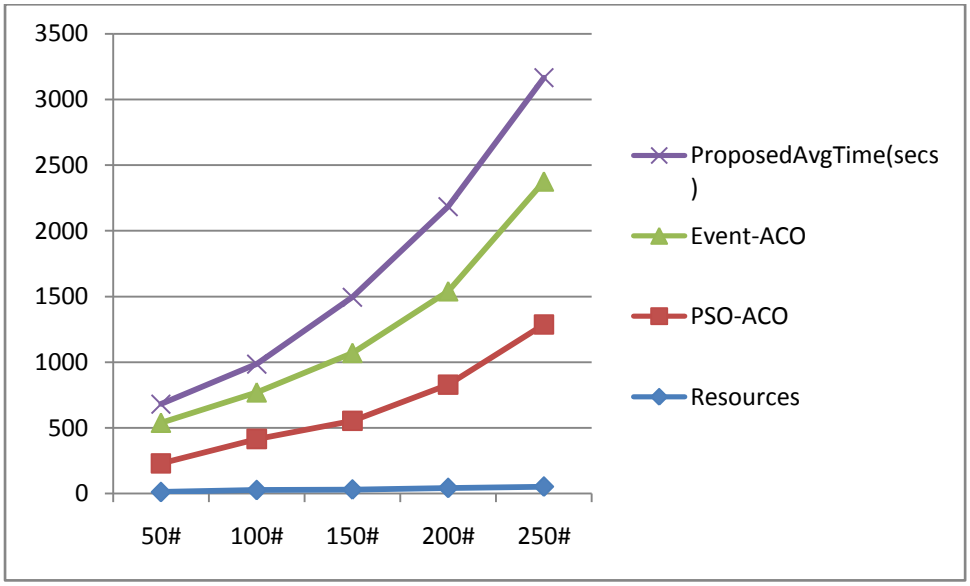


Fig 4: Performance analysis of traditional scheduling algorithm with proposed algorithm

| JOBS | Resources | AllocationTime |
|---|---|---|
| 50# | 12 | 24 |
| 100# | 27 | 32 |
| 150# | 31 | 45 |
| 200# | 43 | 76 |
| 250# | 53 | 98 |

Table 2: Proposed resources vs Allocation Time

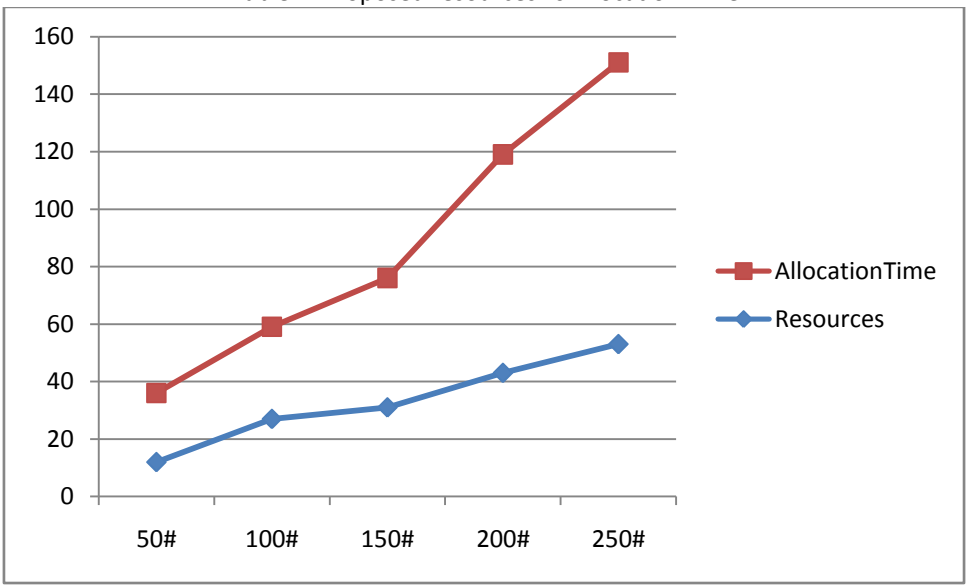


Fig 52: Proposed resources vs Allocation Time

## V.CONCLUSION

Complex and computational problems can be resolved in an efficient way with the help of parallel processing. The parallel processing system can be divided into two categories depending on the nature of tasks such are homogenous parallel system and the heterogeneous parallel processing system. In the homogeneous environment, the number of processors required for executing different tasks is similar in capacity. In case of heterogeneous environments, tasks are allocated to various processors with different capacity and speed. The main objective of parallel processing is to optimize the execution speed and to shorten the duration of task execution with independent of environment. In this proposed work, an optimized

parallel project selection method was implemented to find the optimal resource utilization and project scheduling. The execution speeds of the task increases and the overall average execution time of the task decreases by allocating different tasks to various processors with the task scheduling algorithm.

## REFERENCES

[1] K. C. Ying, S. W. Lin, and Z. J. Lee, "Hybrid-directional planning: Improving improvement heuristics for scheduling resource-constrained projects," International Journal of Advanced Manufacturing Technology, vol. 41, pp. 358– 366, 2009 .

[2] L. Deng, Y. Lin, and M. Chen, "Hybrid ant colony optimization for the resource-constrained project scheduling problem," Journal of Systems Engineering and Electronics, vol. 21, pp. 67–71, 2010.

[3] S-P. Chen and M-J. Tsai,"Time–cost trade-off analysis of project networks in fuzzy environments". Euro J. Oper Res, vol. 212, pp. 386– 397, Jul. 2009.

[4] K. Moumene and J. A. Ferland, "Activity list representation for a generalization of the resource-constrained project scheduling problem," European Journal of Operational Research, vol. 199, pp. 46–54, 2009.

[5] P. Wuliang and W. Chengen,"A multi-mode resource-constrained discrete time–cost tradeoff problem and its genetic algorithm based solution". Inter J. of Proj Manage.Vol.27, pp. 600–609, Aug. 2009.

[6] S. Sakellaropoulos and A.P. Chassiakos,"Project time–cost analysis under generalised precedence relations". AdvEngSoftw. Vol. 35, pp. 715–724, Oct.-Nov. 2004.

[7] F. Kazemi, R. Tavakkoli-Moghaddam, "Solving a multi-objective multi-mode Resource-constrained project scheduling problem with particle swarm optimization" International Journal of Academic Research, Vol. 3, pp. 103-110, 2011.

[8] S. Hartmann, R. Kolisch "Experimental evaluation of state–of–the–art heuristics for the resource–constrained project scheduling problem", European Journal of Operational Research, Vol. 127, pp. 394–407, 2000.

[9] Project Management Institute (PMI), A guide to the project management book of knowledge, Project Management Institute, Newton square, PA, 2008.

[10] M. Pinedo, Scheduling: theory, algorithms and systems, Springer, New York, Third Edition, 2008.