Activity Scheduling on Identical Parallel Processors

Satyasundara Mahapatra, Rati Ranjan Dash, Niroj Kumar Pani

Abstract: ***The efficiency of parallel processors is achieved for the purpose of quick computing is mainly depending on the scheduling of activity. The most important factor for scheduling of activities are depend on the waiting time which is directly influence the computation time of overall activity. Minimizing the Variance of Waiting Time otherwise known as Waiting Time Variance (WTV) is one of the metrics of Quality of Services (QoS) which enhance the efficiency of activity scheduling. Allocate the activity from a set of activity pool and schedule them for each identical parallel processor for execution in a large scale by minimizing WTV is the main focusing area of this paper. In case of large scale computing activities are complex in nature. A prior knowledge of each activity must be known before the preparation of activity scheduling for efficient and rapid computing. A snake walks style of activity distribution among the parallel processor is presented in this paper for $P_m|prec|WTV$ minimization problem. The minimization of WTV is measured with the help of three heuristic intend methods named as RSS, VS and BS. The results of the experiment are compared with current conspires and demonstrate the new snake style conspire is presenting the best practices for proven conspires and challenges in a wide range of activity. The algorithm's predictable findings appear as illustrated with graph.***

Index Terms: ***Activity, Combinatorial, Deterministic, Identical Parallel Processors, Scheduling, Waiting Time Variance****.*

I. INTRODUCTION

In real world, numbers of problems are happening at the same time. These problems are so large and complex that it is not possible to solve them on a single processor with limited memory. From the present perspective, computation on high-speed processors is a process of the dynamic change of information and the execution activities. As semiconductor industries are shifted to parallel processors, there is a paradigm shift in the computational world which is now emerged as the prevalent computing system. Such computing systems are work simultaneously with emerging technologies in parallel processors. These systems are nothing but numerous mathematical calculations can be done several times concomitantly. Hence the best practice is dividing the large and complex problems in to a set of small problems and then computes with the help of parallel processors of same type. Nowadays, parallel processors are becoming more popular due to their high performance and low cost features. This is only possible due to the ongoing developments of the computational technologies named as image processing, machine learning, deep learning, forecasting, large scale computational simulation, etc in the field of engineering. To achieve the higher performances in these technologies depends on not only using faster and more reliable hardware but also the major improvements to the software used in computer building and management techniques. The parallel activity scheduling is one of the concepts that have taken to compute the *'n'* number of activities in batch and execute them on parallel processors of size *'m'* [1].

Required action for scheduling include groupings, sequencing, or sequence sets of groups that meet the usual conditions or limitations of activity scheduling. These bodies are made up of countable discrete structures known as combinatorial structure. These structures are derived through a vector of decisive variables that can take values from a finite or a countable infinite set. For these settings, a scheduling for a $P_m|prec|WTV$ minimization problem solving is the placement of a variable that meets the specified criteria. Such problems create a schedule for the use of constraint satisfaction or optimization problems. From the optimality point of view of the satisfaction experience, it is a difficult process for activity scheduling. The types of problems encountered to perform the activity scheduling on the timeline are similar to the problems encountered in combinatorial optimization modeling. Due to this behavior scheduling is treated as combinatorial optimization problem. The basic problem of activity scheduling is to determine the exact order of schedule chosen with available constraints to reduce the overall completion time to complete the schedule. Rather than looking for a better schedule to complete work that has the earliest possible finishing time, it is best to create a quick schedule with a better and significant time setting up impact. The accuracy of such model is analyzed with the help of actual activity scheduling problem with the importance of data. The deterministic activity scheduling problems with multiple objectives are always belonging to a broader class of scheduling problems. These problems are otherwise known as combinatorial optimization scheduling problem. Hence it belongs to the class NP-hard and creates an optimal solution in linear time is not possible. Hence, it is an interesting area for many researchers to prepare an improved optimized schedule with minimum time complexity. For such problems heuristic is the only solution suggested by the researchers [2]-[4]. But for this, it is very important to know the peculiarities of such problems. These peculiarities provide a clue for finding a quick solution based on heuristic. Such solutions take minimum time for execution and provide a minimum time complexity algorithm for implementation.

Manuscript published on 30 August 2019.
***Correspondence Author(s)**

Satyasundara Mahapatra, Department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur, India.
Rati Ranjan Dash, Department of Mechanical Engineering, College of Engineering and Technology, Bhubaneswar, India.
Niroj Kumar Pani, Department of Computer Science Engineering and Application, Indira Gandhi Institute of Technology, Sarang, India.

Retrieval Number: J99260881019/19©BEIESP
DOI: 10.35940/ijitee.J9926.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

In Fig. (a) the process cycle of scheduling problem belongs to deterministic is explained diagrammatically below.

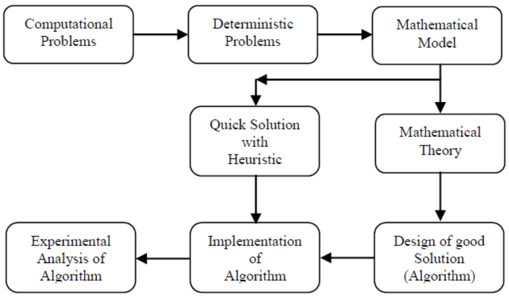
Computational Problems
Deterministic Problems
Mathematical Model
Quick Solution with Heuristic
Mathematical Theory
Experimental Analysis of Algorithm
Implementation of Algorithm
Design of good Solution (Algorithm)

Fig. (a) : Problem solving cycle for deterministic scheduling problem

Parallel processor is consisting of several single processing units, which are connected through one or more interconnected memory hierarchy, auxiliary processors and other devices. Over the time, multiple reviews on different conspires on processor scheduling are done and discussed by the researchers [5]. These conspires are based on exact algorithm, genetic algorithm, simulated annealing, tabu search and heuristic algorithm for single processor rather than parallel processors [6]-[10]. Two types of measurement methods named as variance of waiting time and completion time are presented for stabilizing the scheduling service on single processor in [11].The variance of flow and waiting time is referred to as performance metrics for a single processor activity scheduling problem. It was observed that the order that reduces the difference in flow time is in line with the order that it can reduce the difference in waiting time w.r.t variance. An admission control system based on Batch method is discussed in [12] for give consistency in errand sitting tight time for high need activities. This strategy change the dynamic undertaking arriving issue into a static issue which permits the presentation of assignment planning systems that require a static arrangement of errands to work on for balancing out their holding up time. The produced ideal assignment succession by utilizing WTV must be V-formed is demonstrated and proved in [13] as a significant ideal property with heuristic calculations. A precise guess named as Scharge's Conjecture is portrayed in [14] delivered an ideal arrangement by setting the first longest activity is placed at the last index of the schedule, the second longest activity is placed at the first index of the schedule, the third and fourth longest activities are placed at last but one and last but two index position and so on for the CTV issue. These scheduling problems in general are considered as NP-hard, because the placing of activities is based on the decision or choice issues. Such problem can be optimized in future. The computational hardness of such problems can be examined methodically in detail. It was also observed that the decision taking problems are computationally easy than the optimization based problem. It means that if optimization based problem is solvable in polynomial time than the decision based problem will be solvable. Hence it implies that decision based problem is NP-complete is explained in [15]. The scheduling issues for parallel processors, where the activities with different earliness and tardiness penalty values and a prohibitive basic due date is NP-hard. The parallel processor booking issue, where assignments have distinctive earliness-lateness punishments and a prohibitive due time or date is NP-hard in the valid sense is examined in [16]. The planning issue of parallel processors of identical type is exhibited in [17] with the makespan paradigm is NP-hard, precise calculations are wasteful to limit makespan for medium and enormous estimated issues. In this manner, some heuristic techniques concentrating on the revise procedures are created dependent on dispatching rules. The activity scheduling issue for parallel processor belongs to identical category with makespan basis is known as NP-hard category problems delineated in [18]. Two types of spiral activity scheduling techniques on single processor named as Balanced and Verified is portrayed in [2] with certain demonstrated properties. A heuristic scheduling conspire known as RSS on single processor for WTV issue is exhibited in [3]. Various algorithms based on heuristic for single processor scheduling issues issue is displayed in [4]. From [4], it was observed that there is no such algorithm based on polynomial calculation that will able to produce the optimal solution in each instance. Hence the scheduling issue is viewed as NP-hard class and the computational hardness of such problems might be analyzed. As everybody knows that a computer having single processor can only execute only one activity in a single time. But when the concept of concurrency comes in mind of the researchers then there is a need of a computer with multiple processors executed simultaneously (i.e parallel) to execute the same activity. As per the behavior these processors are classified in to three categories and discussed in [19]. They are otherwise known as Identical (P), Uniform (Q) and Unrelated (R) parallel processors.

Dispersion of the activities from the activity pool to the identical processors and then schedules them efficiently is an interesting problem area for researchers. The most important thing is developing a quick and efficient heuristic algorithm so that the variance limit of waiting time should be minimized. The main focus of this paper is developing an efficient and quick solution for such problem $(P_m|prec|WTV)$ when the size of the activity grows up exponentially. When the activity grows exponentially then the problems are known as hardest form of combinatorial optimization problem and belong to NP-hard class. Finding an exact solution (i.e. schedule) is really impossible for these categories of problem in polynomial time. Hence heuristic conspire is the only way for finding solutions.

The remainder of the paper is organized as pursues. The section 2 contains a quick discussion on the activity scheduling problem for minimizing WTV on identical parallel processor (P_m) with its classification. A notes on activity scheduling problem formulation for obtaining mean WTV on P_m is presented in section 3. Four activity allotment conspires for P_m is discussed in section 4. In Section 5, the procedure of heuristics are accumulate with allotment conspires is discussed for calculating mean WTV. Formulation of test cases as examples and evaluate the effectiveness of proposed algorithm after implementation on P_m is discussed in section 6. Finally the section 7 contains the conclusion and recommendation for future execution of the work.

II. ACTIVITY SCHEDULING ON P_m

A triplet $(\alpha\,|\,\beta\,|\,\gamma)$ representation format for activity scheduling problem is depicted in [19].The $'\alpha'$ symbol is used for the environment types of processer. These are single or multiple processors. The single processor are again divided into three categories and identified by name as Single, Parallel and Dedicated processor. The $'\beta'$ symbol is used for different types of activities and the constraints for available resources. The input on this symbol may be null or one input or multiple inputs. The types of possible inputs are Preemption $(pmtn)$, Resource (res), Precedence $(prec)$, Ready Time (r_j), Delivery Time (q_j), Processing time (pt_j), Deadline $(\tilde{d})$ and No-wait $(no-wait)$. Finally the $'\gamma'$ is used as an objective function for minimizing different criteria like Makespan $(C_{\max})$, Mean Flow time $(\overline{F})$, Mean weighted Flow time $(\overline{F}_w)$, Maximum Lateness (L_{max}), Mean Tardiness $(\overline{D})$, Mean weighted Tardiness $(\overline{D}_w)$, Mean Earliness $(\overline{E})$, Mean weighted Earliness $(\overline{E}_w)$, Number of Tardy task $(\overline{U})$ and Weighted number of Tardy Task $(\overline{U}_w)$.

The activity scheduling problem on P_m is consists of two steps. First step is responsible for the allocation of activity to the processors. Second step is responsible for the schedule the activity of individual processor. Preemptions play a vital role with parallel processors. In case of single processor preemptions is an important aspect when activities are released at different time interval. But on identical parallel processors, preemptions are more important as all the activities are released at the same time. The general identical parallel processor (P_m) scheduling problem classification without preemption and with preemption is depicted in Fig. (b) and Fig. (c) respectively with the help of triplet.

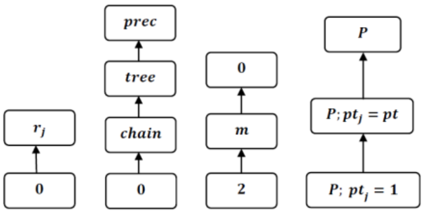

Fig. (b) : Classification of identical parallel processor problems without preemption

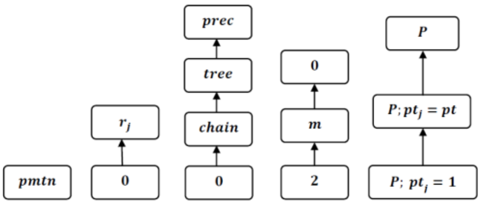

Fig. (c) : Classification of identical parallel processors problems with preemption

In the both case of identical parallel processors the speeds are logical in numbers. It was observed that, the times required to process each unit are equivalent to constant, so the case $pt_j = 1$ was not considered.

III. PROBLEM FORMULATION FOR P_m

This paper focus on a problem where a number of identical processors are executed parallelly for accomplishing an objective. From the real parallel working environment it was observed that n numbers of activities are executed by m numbers of identical processors in k number of location. Where,

- $i \in \mathcal{P} = \{1,2,\dots,m$ represent for identical processors and the value of m must be greater than 1.
- $j \in \mathcal{T} = \{1,2,\dots,n\}$ represent for independent activity and the value of n must be greater than m
- $k \in \mathcal{L} = \{1,2,\dots,D\}$ represent for the location of identical processor and the value of D is $\left\lceil \frac{n}{m} \right\rceil$.

The activity $j \in \mathcal{T}$ has assigned by a execution time pt_j and can be executed by any one processor $i \in \mathcal{P}$. This problem is defined under four numbers of presumptions. They are given below.

- The initial execution time of all processors are 0(i.e. zero).
- When an activity is executing in any processor then no other activity will be allowed to enter on the same processor. Hence preemption is not allowed.
- Once an activity assigned to one processor, then the same activity cannot be switchover to other processor in any condition.
- All the dispensed processors will hold up as indicated by the request of allotment

It was observed from the review that researchers have developed and proved a number of authoritative properties and depicted in [2][8[[13][21].In case of activity scheduling problem WTV minimization is one of the important measure through which the efficiency of a single processor is measured and discussed in [2]-[4]. The identical processors is the multiple version of single processor and work simultaneously (i.e. parallelly) to achieve the speedup in execution of activity. Hence to find an optimized schedule with minimized WTV is the goal of the activity scheduling in parallel condition. To calculate the WTV in parallel environment with the help of identical processors for activities is written in equation (1) as follows

$$WTV = \frac{1}{m}\sum\nolimits_{i\in\mathcal{P}}\left(\frac{1}{n_i-1}\sum\nolimits_{j\in\mathcal{L}_{ki}}\left(wt_{kij} - \frac{1}{n_i}\sum\nolimits_{j\in\mathcal{L}_k} wt_{kij}\right)^2\right) \qquad (1)$$

The goal of this paper is to locate an optimal or closely optimal schedule by minimizing WTV for activity scheduling problem in parallel environment with identical processors. This can be calculated by the help of following equation (2)

$$Minimize\ (P_m|prec|WTV) \qquad (2)$$

Subject to:

$$\sum\nolimits_{j\in\mathcal{P}_i} s_{kij} = 1 \qquad (3)$$
$$\sum\nolimits_{i\in\mathcal{T}_j} s_{kij} = 1 \qquad (4)$$
$$\sum\nolimits_{j\in\mathcal{P}_i} wt_{kij} = 0 \qquad (5)$$
$$wt_{kij} = wt_{k-1ij} + \sum\nolimits_{j\in\mathcal{P}_i} s_{kij} * pt_j \qquad (6)$$
$$\sum\nolimits_{j\in\mathcal{P}_i} q_{kij} = 1 \qquad (7)$$
$$q_{kij} * N + wt_{k+1ij} \geq wt_{kij} + \sum\nolimits_{j\in\mathcal{P}_i} s_{kij} * pt_j \qquad (8)$$
$$s_{kij} \in \{0,1\} \qquad (9)$$
$$q_{kij} \in \{0,1\} \qquad (10)$$
$$C_{kij} \geq 0 \qquad (11)$$
$$wt_{kij} \geq 0 \qquad (12)$$

where N is a very large number

Equation 3 represent for the allocation of each activity such that once it was allocated it should not be allocated to other processor. Equation 4 is responsible for the sequence of activities that is utilized precisely once. Equation 5 is responsible for the first undertaking activities for each processor. Equation 6 is responsible for calculating the other activities of a schedule except the first one for each processor.

Equation 7 and 8 express that on the off chance that two activities are on a similar processor; at that point one must be scheduled after the other. Equation 9 and 10 represent for the decision taken by the variable and binary in nature. Equation 10 and 11 is store completion time of each activity and waiting time of each activity on their respective processor.

IV. ACTIVITY ALLOTMENT CONSPIRES FOR P_m

Distribution of the activities to each processor (P_m) and then schedule them in such a way, so that the WTV of activity scheduling problem ($P_m|prec|WTV$) should be minimized is the main focus of this paper. From the review that the define problem is belongs to the class NP-complete. It was also observed that out of number of heuristic method three heuristic methods are highly efficient for minimizing the WTV calculation and discussed in [2]-[4]. Out of these three methods two methods are based on spiral concept and known as Verified and Balanced method. The third one is RSS method based on the fish cutting and distributes them by the fishmonger. But the most important thing is how to allocate the activity to each processor (P_m) uniformly, so that heuristic algorithm is applicable for Minimize WTV calculation. For that a new method named as (RS) is introduced on the concept of rattle snake walk and explained below with the help of two other allocation methods named as Random Order (RO) and Shortest Processing Time (SPT).

A. Random Order (RO)

Random order activity distribution method is based on the concept that the algorithm pick up the activities randomly from the activity pool. Then allocate that activity with its execution time to the processor (P_m) for whom that activity is selected. Let the total number of independent activity is *'n'* and the total number of processors (P_m) is *'m'*. So each processor (P_m) gets maximum $\left\lceil \frac{n}{m} \right\rceil$ number (i.e. *'D'*) of activity to each processor sequentially. The distribution of *'D'* number of activities is start from processor *1* and end at processor '*m-1'*. Rest all available activities are distributed to processor *'m'*.

B. Shortest Processing Time (SPT)

In case of shortest processing time activity distribution method, first all the jobs are sorted in ascending order with respect to their processing time. Then allocate the first $\left\lceil \frac{n}{m} \right\rceil$ number (i.e. *'D'*) of activities sequentially to the 1st processor and then select next $\left\lceil \frac{n}{m} \right\rceil$ number (i.e. *'D'*) of activities sequentially to the 2nd processor. Do the same allocation process up to '*m-1'* processor. Then rest all available activities are distributed sequentially to the processor *'m'.*

C. Longest Processing Time (SPT)

In case of shortest processing time activity distribution method, first all the jobs are sorted in descending order with respect to their processing time. Then allocate the first $\left\lceil \frac{n}{m} \right\rceil$ number (i.e. *'D'*) of activities sequentially to the 1st processor and then select next $\left\lceil \frac{n}{m} \right\rceil$ number (i.e. *'D'*) of activities sequentially to the 2nd processor. Do the same allocation process up to '*m-1'* processor. Then rest all available activities are distributed sequentially to the processor *'m'*.

D. Proposed allotment conspire (RS)

After observing the number of snakes crawling style, it was observed that the rattle snake crawl style is very peculiar. Hence the new proposed conspire if fully based on Rattle snake crawl style and the activities are allocated accordingly for execution in the individual $P_m$processor by maintaining the average processing time or nearer to same. The maximum tasks allocated to each processor are $\left\lceil \frac{n}{m} \right\rceil$ numbers. For example a chart for the distribution style for twenty numbers of activities among five numbers of processors is shown on Table I. All the tasks are allotted in descending order as per their processing time prior to commencement of allocation.

Table I : Distribution style for twenty numbers of activities among five numbers of processors

P_1	P_2	P_3	P_4	P_5
A_{20} A_{11} A_{10} A_1	A_{19} A_{12} A_9 A_2	A_{18} AJ_{13} A_8 A_3	A_{17} A_{14} A_7 A_4	A_{16} A_{15} A_6 A_5

Then the efficiency of the three heuristic methods discussed in [2]-[4] are implemented for the sequences generated by these above four allocation method. Thus the computational efficiency on identical parallel processor (P_m) is justified. The mechanisms of assimilate allocation methods are discussed in next session

V. ALLOTMENT CONSPIRES FOR P_m WITH HEURISTICS

The above discussed four activity allotments conspires are now assimilate with two spiral type heuristic methods (i.e. VS, BS) and one fish cutting style heuristic method (i.e RSS) discussed in [2]-[4] for enhancing the efficiency of activity allotments conspires for P_m processors. Table II describes a chart for easy understanding.

Table II : Activity allotments conspires with heuristics

		Heuristics		
		VS	BS	RSS
Activity Allotments Conspire	RO	RO ∪ VS	RO ∪ BS	RO ∪ RSS
	SPT	SPT ∪ VS	SPT ∪ BS	SPT ∪ RSS
	LPT	LPT ∪ VS	LPT ∪ BS	LPT ∪ RSS
	RS	RS ∪ VS	RS ∪ BS	RS ∪ RSS

At first Random Order (RO) activity allotments conspire is described. It is followed by the same scheduling method along with SPT, LPT and proposed method RS respectively.

A. RO with heuristics

The algorithm for calculating mean WTV of Random Order (RO) activity allotments conspire with the help of heuristics is presented below with pseudo code.

```
function Random Order(RO)
    Let n ← number of activities
    Let m ← number of $P_m$ processor
    If (n>m)
        Calculate the number of activities distributed to all
        the $P_m$ processors    except the last one (i.e. D=$\left\lceil \frac{n}{m} \right\rceil$)
```

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

```
For i ← 1 to m-1 processors
        Randomly pick up D number of activities
        from the activity pool and distribute to $i^{th}$ processor
    End for
    Then transfer rest all activities available in the
    activity pool to $m^{th}$ processor sequentially.
    For i ← 1 to m processors
        Calculate WTV by using VS, BS and RSS
        heuristics for $i^{th}$ processor simultaneously.
    End for
    Find the mean WTV through each heuristic.
  End if
 End function
```

B. SPT with heuristics

The algorithm for calculating mean WTV of Shortest Processing Time (SPT) activity allotments conspire with the help of heuristics is presented below with pseudo code.

```
function Shortest Processing Time(SPT)
    Let n ← number of activities
    Let m ← number of $P_m$ processor
    If (n>m)
        Calculate the number of activities distributed to all
        the $P_m$ processors except the last one (i.e. D=$\lceil\frac{n}{m}\rceil$).
        Sort the activity pool in ascending order by using an
        efficient sorting algorithm
        For i ← 1 to m-1 processors
            Sequentially pick up D number of activities
            from the activity pool and distribute to $i^{th}$
            processor
        End for
        Then transfer rest all activities available in the
        activity pool to $m^{th}$ processor sequentially.
        For i ← 1 to m processors
            Calculate WTV by using VS, BS and RSS
            heuristics for $i^{th}$ processor simultaneously.
        End for
        Find the mean WTV through each heuristic.
    End if
End function
```

C. LPT with heuristics

The algorithm for calculating mean WTV of Longest Processing Time (LPT) activity allotments conspire with the help of heuristics is presented below with pseudo code.

```
function Longest Processing Time(LPT)
    Let n ← number of activities
    Let m ← number of $P_m$ processor
    If (n>m)
        Calculate the number of activities distributed to all
        the $P_m$ processors except the last one (i.e. D=$\lceil\frac{n}{m}\rceil$)
        Sort the activity pool in descending order by using
        an efficient sorting algorithm
        For i ← 1 to m-1 processors
            Sequentially pick up D number of activities
            from the activity pool and distribute to $i^{th}$
            processor
        End for
        Then transfer rest all activities available in the
        activity pool to $m^{th}$ processor sequentially.
        For i ← 1 to m processors
            Calculate WTV by using VS, BS and RSS
            heuristics for $i^{th}$ processor simultaneously.
        End for
        Find the mean WTV through each heuristic.
    End if
End function
```

D. RS with heuristics

The algorithm for calculating mean WTV of Rattle Snake (RS) activity allotments conspire with the help of heuristics is presented below with pseudo code.

```
function Rattle Snake(RS)
    Let n ← number of activities
    Let m ← number of $P_m$ processor
    If (n>m)
        Calculate the number of activities distributed to all
        the $P_m$ processors except the last one (i.e. D=$\lceil\frac{n}{m}\rceil$)
        Sort the activity pool in descending order by using
        an efficient sorting algorithm
        For i ← 1 to D times
            Distribute the activity as per the rattle snake
            crawl style (discussed in section 4.3)
        End for
        For i ← 1 to m processors
            Calculate WTV by using VS, BS and RSS
            heuristics for $i^{th}$ processor simultaneously.
        End for
        Find the mean WTV through each heuristic.
    End if
End function
```

The productivity of the above examined heuristic functions with activity allocation conspire for P_m processors is tried with an enormous number of experimental cases is talked about in the following segment

VI. PROBLEM TESTING & EFFECTIVENESS OF P_m

To evaluate the effectiveness of above discussed activity allocation conspires with heuristics with respect to mean WVT, a number of experimental test cases are created randomly. For this three probability distribution functions named as Normal, Poisson and Exponential are used. From the outset with the assistance of these three distributions function 900 numbers of experimental test cases are created randomly. Each test case contains 100 to 1000 numbers of activities. These test cases are then tested with 5 to 10 numbers of P_m processors respectively for evaluating the performance of heuristic algorithm.

The outcomes of the randomly generated problems on 7 and 10 numbers of P_m processors are delineated through nine figures (i.e. Fig. (d) through Fig. (l)). Each figure comprises of four subfigures (i) through (ii) consecutively. The mean WTV acquired by one of the three heuristic algorithms with four activity distribution method is appeared in every subfigure. Four different colors are used for the representation of four activity distribution methods in each subfigure (i.e. (i) and (ii) for all the nine figures. They are distinguish with the help of blue, green, black and red colors for RO, SPT, LPT and RS activity allocation conspires respectively. Out of all activities, 221 to 226 serial numbers activities' augmented view on mean WTV is exhibited in every subfigure for elucidation. The beauty of heuristic algorithms is that, every algorithm has produced only one sequence for each experimental test case and gives us optimum or near optimum WTV as discussed in [5]. Another interesting thing is that, the structure of the generated sequences is always V-formed.

The essential point is to calculate the mean WTV for each activity allotment conspire on P_m processors and optimize it.Out of three distribution technique discussed above, at first by using Normal Distribution the experimental test cases are generated. Then all the experimental test cases are allocated among 5 to 10 numbers of P_m processors by using four activity allocation conspires namely RO, SPT, LPT and RS respectively. Then with the help of VS heuristic the execution schedule of allotted activities for each process are formed and processed. After the activity schedule processed the mean WTV is calculated for each activity schedule for the performance evaluation. The experimental outcome of VS heuristic with four activity allotment conspires on 7 and 10 numbers of P_m processors is shown below in Fig. (d).

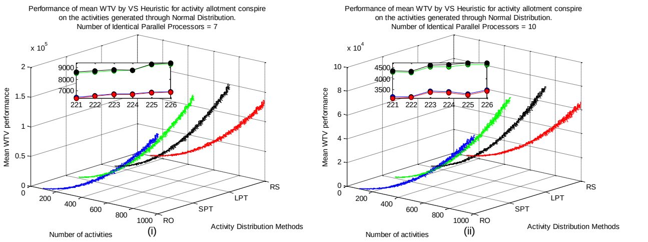

Fig. (d) : Mean WTV performance calculated by VS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Normal distribution.

The obtained experimental result in fig. 4 shows that the proposed activity allotment conspires named as RS with VS heuristic is always gives least mean WTV and the heuristic generated schedule for each experimental test cases is satisfied the property of V-formed . Then again with the help of BS heuristic the execution schedule of allotted activities for each process are formed and processed. After the activity schedule processed by the processor the mean WTV is calculated for each activity schedule and evaluate its performance. The experimental outcome of BS heuristic with four activity allotment conspires on 7 and 10 numbers of P_m processors is shown below in Fig. (e).

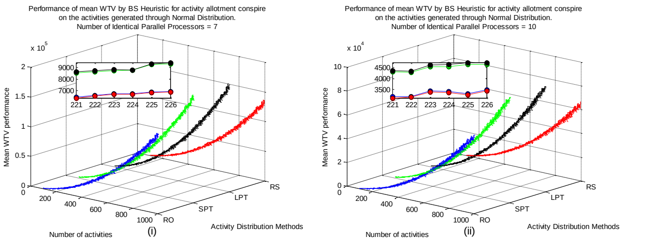

Fig. (e) : Mean WTV performance calculated by BS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Normal distribution.

The obtained experimental result in fig. 5 shows that, among all the activity allotment conspires the proposed activity allotment conspire named as RS with BS heuristic is always gives least mean WTV and the generated sequences for each experimental test cases are satisfied the property of V-formed. Then again with the help of RSS heuristic the execution schedule of allotted activities for each process are formed and processed. After the activity schedule processed the mean WTV calculated for each activity schedule for the performance evaluation. The experimental outcome of RSS heuristic with four activity allotment conspires on 7 and 10 numbers of P_m processors is shown below in Fig. (f).

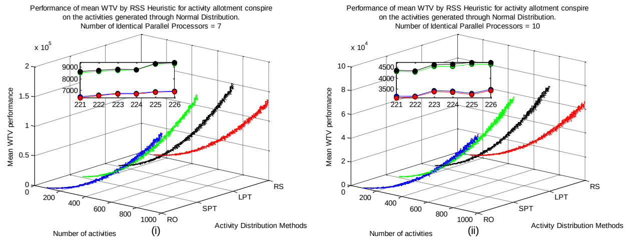

Fig. (f) : Mean WTV performance calculated by RSS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Normal distribution.

The obtained experimental result in fig. 6 shows that, among all the activity allotment conspires the proposed activity allotment conspire named as RS with RSS heuristic is always gives least mean WTV and the generated sequences for each experimental test cases are satisfied the property of V-formed . From the three heuristic execution it was observed that activities generated through Normal distribution, allotted to 5 to 10 numbers P_m processors by RS allotment conspire obtained mean WTV is always optimum (i.e. minimum) as compared with mean WTV obtained by other activity allotment conspires.

Poisson distribution is one of the well-utilized discrete distribution methods for activity generation purpose. Hence for the second time the four activity allotment conspires and three heuristics are again tested with the activity generated by Poisson distribution on 5 to 10 numbers P_m processors. 100 to 1000 experimental test cases are again generated with the help of Poisson distribution. Then all the experimental test cases are distributed among 5 to10 numbers of P_m processors by using four activities allotment conspires namely RO, SPT, LPT and RS respectively. Then with the help of VS heuristic the execution schedule of allotted activities for each process are formed and processed. After the activity schedule processed by the processor the mean WTV is calculated for each activity schedule and evaluate its performance. The experimental outcome of VS heuristic with four activity allotment conspires on 7 and 10 numbers of P_m processors is shown below in Fig. (g).

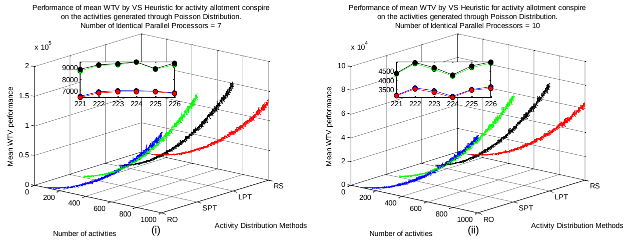

Fig. (g) : Mean WTV performance calculated by VS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Poisson distribution.

The obtained experimental result in fig. 7 shows that the proposed activity allotment conspire named as RS with VS heuristic is always gives least mean WTV once again. The heuristic generated schedule for each experimental test cases is also satisfied the property of V-formed. Then again with the help of BS heuristic the execution schedule of allotted activities for each process are formed and processed. After the activity schedule processed by the processor the mean WTV is calculated for each activity schedule and evaluate its performance.

Retrieval Number: J99260881019/19©BEIESP
DOI: 10.35940/ijitee.J9926.0881019
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

The experimental outcome of BS heuristic with four activity allotment conspires on 7 and 10 numbers of P_m processors is shown below in Fig. (h).

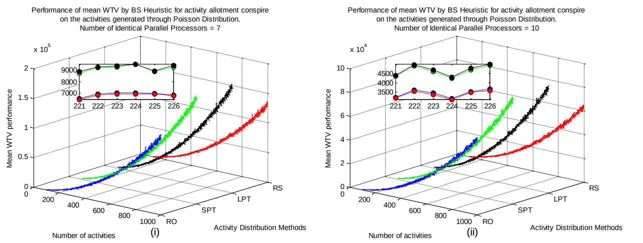

Fig. (h) : Mean WTV performance calculated by BS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Poisson distribution.

The obtained experimental result in fig. 8 shows that, among all the activity allotment conspires the proposed activity allotment conspire named as RS with BS heuristic is always gives least mean WTV and the generated sequences for each experimental test cases are satisfied the property of V-formed. Then again with the help of RSS heuristic the execution schedule of allotted activities for each process are formed and processed. After the activity schedule processed the mean WTV calculated for each activity schedule for the performance evaluation. The experimental outcome of RSS heuristic with four activity allotment conspires on 7 and 10 numbers of P_m processors is shown below in Fig. (i)

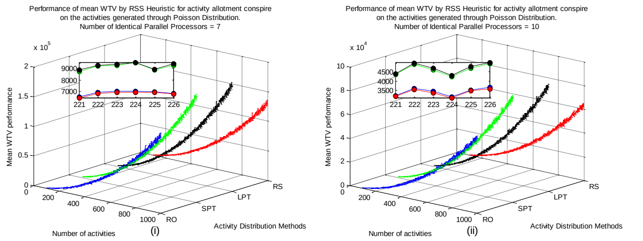

Fig. (i) : Mean WTV performance calculated by RSS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Poisson distribution.

The obtained experimental result in fig. 9 shows that, among all the activity allotment conspires the proposed activity allotment conspire named as RS with RSS heuristic is always gives least mean WTV and the generated sequences for each experimental test cases are satisfied the property of V-formed . From the three heuristic execution it was observed that activities generated through Poisson distribution, allotted to 5 to10 numbers P_m processors by RS allotment conspire again obtained mean WTV is always optimum (i.e. minimum) as compared with mean WTV obtained by other activity allotment conspires.

Exponential distributions are often used to model the time between independent tasks that happen at a constant average rate. To strengthen the concept of activity allotment conspires and heuristics, once again 100 to 1000 experimental test cases generated by exponential distribution for the third time are again tested on 5 to10 numbers P_m processors. Again the allotment of activities are done by four activity allotment conspires to each processor. Then the allotted activities of each processor are scheduled by three heuristic and the outcomes are evaluated. The experimental outcome of VS heuristic with four activity allotment conspires on 7 and 10 numbers of P_m processors is shown below in Fig. (j).

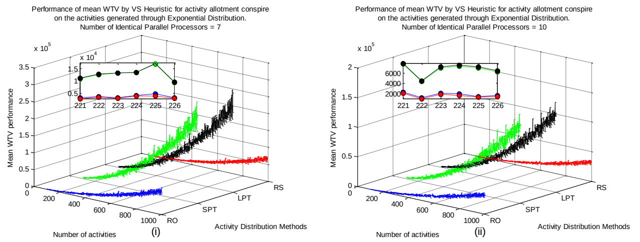

Fig. (j) : Mean WTV performance calculated by VS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Exponential distribution.

After all the activity test cases are tested with VS heuristic it was observed that the mean WTV presented in Fig. 10 depicts that the activity allocation done by RS activity allotment conspire gives best results. Then the allotted activities of each processor are scheduled by BS heuristic and again tested on 5 to10 numbers of P_m processors. The mean WTV performances of all the activity allotment conspires on 7 and 10 numbers of P_m processors are presented in Fig. (k).

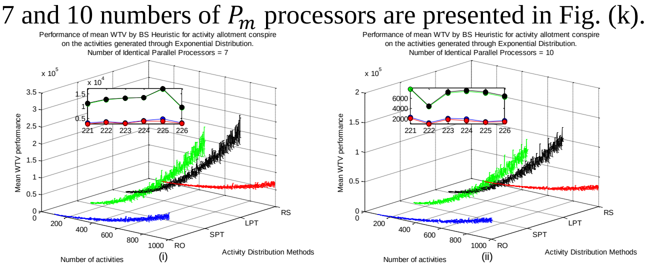

Fig. (k) : Mean WTV performance calculated by BS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Exponential distribution.

The mean WTV performance of fig. 11 shows activity allotment conspire done by RS allotment conspire always gives best results than the other three activity allotment conspires. Finally the allotted activities of each processor are scheduled by RSS heuristic and tested on 5 to 10 numbers of P_m processors. The mean WTV performances obtained by RSS heuristic are conferred for 7 and 10 numbers of P_m processors is displayed in Fig. (l).

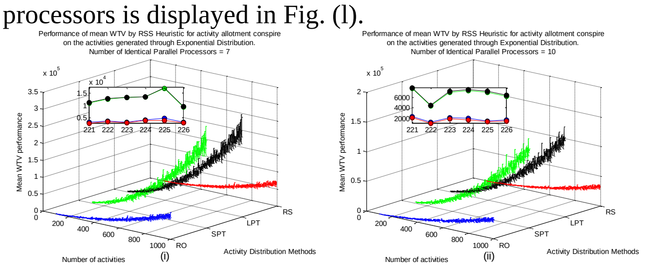

Fig. (l) : Mean WTV performance calculated by RSS heuristic w. r. t. four activity allotment conspire implemented on the activities generated through Exponential distribution.

The above said Fig. 12 as exhibited the performance of mean WTV of four activity allotment conspires. It was again found that the activity allocated through RS allotment conspire is again the best and the obtained mean WTV is optimum. The purpose of implementing heuristic for generating an execution sequence swiftly by minimizing WTV for the allotted activity to an individual processor is truly achieved. Because it enhances the performance of all the four activity allotment conspires by calculating the mean WTV. From the above performance calculation it was observed that due to heuristics the performance of Rattle Snake (RS) activity allotment conspire is found more

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

suitable than the other activity allotment conspires on P_m processors.

VII. CONCLUSION

In this paper, classification of P_m processors are studied and presented. The performance of three heuristic algorithms i.e. VS, BS and RSS to minimize WTV for an activity scheduling issues was studied and implemented on P_m processors. The main objective of this paper is to minimize the mean WTV of P_m processors. It is a two phase process. In the first phase three activity allotments conspires namely RO, SPT, LPT and one newly proposed allotment conspire named as RS based on rattle snake crawl style is presented. While in second phase, these four activity allotment conspires are assimilated with the three heuristics (i.e. VS, BS and RSS) for enhancing the performance of activity allotment conspires. The assimilate algorithms are then implanted and tested on 900 experimental test cases. These test cases are generated through Normal, Poisson and Exponential distribution. The performance calculation of these assimilate algorithms are obtained and presented in this paper. After comparing the obtained mean WTV, it was observed that allocation of activities through RS activity allotment conspire with all the heuristic is always performing best. To fortify and give a decent quality service oriented application by utilizing the resources of computer world and system in parallel like routing of data through packet, provide services through scheduling on web server, the proposed RS with heuristic might be utilized. The RS activity allotment conspire can likewise be connected for parallel assignment planning for clump on an assembling unit of an industry.

REFERENCES

1. Singh N, Kaur G, Kaur P, Singh G. “Analytical Performance Comparison of BNP Scheduling Algorithms”, Global Journal of Computer Science and Technology, Hardware & Computation, Volume 73, Issue 10, 2012, pp. 17-24.
2. Nong Ye, Li X, Farley T, Xu X. “Job scheduling methods for reducing waiting time variance.”, Computer & Operations Research (Elsevier) , Volume 18, 2007, pp. 3069-3083.
3. Mahapatra S, Dash R R, Pradhan S K. “RSS Algorithm for Job Scheduling in Single Machine, A Novel Approach”, International Journal of Advanced Computer Engineering and Communication Technology (IJACECT), ISSN (Print): 2319-2526, Volume-3, Issue 3, 2014, pp.24-30.
4. Mahapatra S, Dash R R, Pradhan S K. “A Heuristic Job Scheduling Algorithm for minimizing the Waiting Time Variance”, International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Amity University, Greater Noida, IEEE Xplore Digital Library, Print ISBN: 978-1-4799-8432-9, 25-27 February, 2015, pp. 446 – 451.
5. Chen B, Potts CN, Woeginger GJ (1998) A review of machine scheduling: Complexity, algorithms and approximability. In: Du D-Z, Pardalos PM (ed) Handbook of Combinatorial Optimization, Kluwer, Dordrecht, The Netherlands, vol. 3, 1998, pp. 21-169.
6. Senthilkumar P. and Narayanan S., “Simulated Annealing Algorithm to Minimize Makespan in Single Machine Scheduling Problem with Uniform Parallel Machines,” Intelligent Information Management, Volume 3, Issue 1, 2011, pp. 22-31.
7. Nowicki E, Smutnicki C. “A Fast Taboo Search Algorithm for the Job Shop Problem”, Management Science, Issue 6, 1996, pp. 797-813.
8. Watson J P, Howe A E, Darrell Whitley L. “Deconstructing Nowicki and Smutnicki”s I-Tsab Tabu Search Algorithm for the Job-Shop Scheduling Problem”, Computers and Operations Research, Volume 2, Issue 9, 2006, pp. 2623-2644.
9. Yamada T, Reeves C. “Permutation Flowshop Scheduling by Genetic Local Search”, 2nd IEE/IEEE Int. Conf. on Genetic Algorithms in Engineering Systems (GALESIA “97). Glasglow, UK, Volume 34, 1997, pp. 232-238.
10. Murata T., Ishibuchi H., Tanaka H., “Genetic Algorithms for Flowshop Scheduling Problems”, Computers & Industrial Engineering, Volume 30, Issue 4, 1996, pp. 1061-1071.
11. Merten A G, Muller M E. “Variance Minimization in Single Machine Sequencing Problems”, Management Science, Volume 38, Issue 9, 1972, pp. 518-528.
12. Farley Ye N, Li X, Harish B. “Batch Scheduled Admission Control for Computer and Network Systems”, Information Knowledge Systems Management, Volume 5, Issue 4, 2007, pp. 211-226.
13. Eilon S, Chowdhury I G, “Minimizing Waiting Time Variance in the Single Machine Problem”, Management Science, Volume 34, Issue 6, 1977, pp. 567-575.
14. Schrage L. “Minimizing the Time-in-System Variance For A Finite Jobset”, Management Science, Volume 207, Issue 5, 1975, pp. 540-543.
15. Chakraborty U K, Laha D. “An improved heuristic for permutation flowshop scheduling”, International Journal of Information & Communication Technology, Volume 6, 2007, pp. 89-97.
16. Rios-Solis Y A, Sourd F. “Exponential neighbourhood search for a parallel machine scheduling problem”, Computers & Operations Research, Volume 1, 2008, pp.1697-1712.
17. Jozefowska J, Mika M, Rozycki R, Waligora G, Weglarz J. “An almost optimal heuristic for preemptive C_{max} scheduling of dependent tasks on identical parallel processors.”, Annals of Operation Research, Volume 129, Issue 129, 2004, pp. 205-216.
18. Akyol D E. “Identical Parallel Machine Scheduling with Dynamical Networks using Time-Varying Penalty Parameters”, Multiprocessor Scheduling: Theory and Applications, Book edited by Eugene Levner, ISBN 978-3-902613-02-8, Itech Education and Publishing, Vienna, Austria, 2007, pp. 293-314.
19. Pinedo M. “Scheduling theory, algorithms and systems”, Englewood Cliffs, NJ: Prentice-Hall; 1995.
20. Hall N G, Kubiak W. “Proof of a conjecture of Schrage about the completion time variance problem”, Operations Research Letters, Volume 27, 1991, pp.467-472.
21. Xu X , Ye N. “Minimization of job waiting time variance on identical parallel machines”, IEEE transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews, Volume 37, Issue 5, 2007, pp. 917-927.

AUTHORS PROFILE

Satyasundara Mahapatra is an Associate Professor in the department of Computer Science and Engineering, Pranveer Singh Institute of Technology, Kanpur, Uttar Pradesh. He received his Master degree and Ph.D in Computer Science from Utkal University, Bhubaneswar, Odisha in 2006 and 2016 respectively. He also holds a MCA degree from Utkal University, Bhubaneswar, Odisha in 1997. His current research interests include Machine Learning, Image Processing,, Scheduling and Internet of Things. He has authored or co-authored in international scientific journals, two book Chapters and three patents approval in his field of expertise.

Rati Ranjan Dash is Professor of Mechanical Engineering Department, College of engineering and Technology, Bhubaneswar, Odisha, India. He received the Bachelor degree in Mechanical Engineering from the University College of Engineering, Burla, Odisha in 1987, Master degree in Applied Mechanics from Indian Institute of Technology, Delhi, India in 1993 and the Ph.D. degree in Robotics from Utkal University, Bhubaneswar, India in 2004. His main research interests are in control of mechanical systems, robotics, multibody dynamics, robot manipulation, scheduling and soft computing.

Niroj Kumar Pani received his M.Tech in Computer Science with specialization in Information security from National Institute of Technology, Rourkela, India in 2009 and Ph.D. in computer science from Utkal University, Odisha, India in 2018. He had worked as Assistant Professor in Indian Institute of Science and Information Technology and as a senior analyst in PMAP India. At present, he is working as Assistant Professor in the Department of Computer Science Engineering and Applications in Indira Gandhi Institute of Technology, Sarang, India. His research interests include applied cryptography, network security, wireless ad hoc and sensor networks, clouds, and Internet of Things.