ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.11, 9959–9972
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.11.104

AN APPROACH FOR JOB SHOP SCHEDULING USING PETRI NETS

K. KAVITHA[1] AND T. BABITHA

ABSTRACT. In this work, Petri Net-based united approach, for instantaneously modelling and scheduling industrial systems, is offered. A model that explain the implementation of the sequencing in single, multiple workstation, and "implements priority dispatching rules" to solve the ultimate conflicts, arediscussed. The proposed model was tested by taking a firm environment example that has flexible job shop-type system. proposed RK_{max} Algorithm has been experimented for solving sequence problems and compared the result of proposed method with the result of existing methods.

1. INTRODUCTION

The selection of jobs, and selecting machines for processingand assigning of resources are operative decisions related to job scheduling which affect the performance in most manufacturing settings. Such decisions have significant impact on system operational costs, efficiency of the system and are often taken automatically by the users, based on their practice.In order to help this conclusion process, a computational application is required such that it provide a modelling aid and scheduling techniques, proper production plan [11]. It is difficult to forecast the performance of industrial systems without modelling, analyzing, and control systems [9]. Therefore, several methods have been developed to define the performance of industrial systems. Petri nets (PNs) is one

[1]*corresponding author*

2010 *Mathematics Subject Classification.* 90B06, 90B10.

Key words and phrases. Scheduling, Petri Nets, Dispatching Rules.

of them. In 1962 Karl Petri introduces the technique of Petri Net for communication system analysis. PNis a leading graphical tool for modelling, analysing systems [10]. Using of Petri nets, can visualizing acompound systems, hierarchically and analyse qualitative and quantitative aspects.The qualitative analysis of physical properties like the absence of deadlocks,overflows and resource sharing, and the quantitative analysis for presentation properties such as material, consumption rates, typical queue lengths, average completion timescan be visualized by using PN's. "A Beam Search algorithm was implemented on Petri Nets in order to find an optimal production schedule [2]. The Branch and Bound method was used in robot task programming and for selecting the firing transition [6, 7]". This paper has a completeanalysisabout the Petri Net applications inmodelling, scheduling of jobs in industrial systems and the number of dispatching rules were implemented for solving eventual scheduling conflicts (selection of jobs in queue, machine priorities) and describing the Gantt chart, and system performance measures [5, 12]. And also we proposed R_{max} Algorithm for job sequencing and compare it with the Johnson's algorithm at last [8].

2. PETRI NETS

Petri Net (PN) is a mathematical modelling tool. It is a direct bipartite graphwhose nodesare called places and transitions and edges are called directed arrows. Transition(t) of the petri net represents events that should take place and are regularlyshowed as rectangles. Places (P) represent the cause or result the events, places are explicitlyshowed as circles. Arcs (a)path from a place to a transition or from a transition to a place, but not everamong two nodes of the same kind [13]. Each arc is linked with a number called weight or valuation function (if omitted, it is assumed to equal 1).

Definition 2.1. *A Petri-Net is a five-tuples,* $PN = (P, T, F, W, M_0)$*, where,*
$P = \{ p_1, p_2, \ldots, p_m\}$ *"is a finite set of places"*
$T = \{ t_1, t_2, \ldots, t_n\}$ *"is a finite set of transitions"*
$F \subset (P \times T)\, U\, (T \times P)$ *"is a set of arcs (flow relation)"*
$W : f \rightarrow \{1, 2, 3, \ldots\}$ *"is a weight function"*
$M_0 : P \rightarrow \{0, 1, 2, 3, \ldots\}$ *"is the initial marking"*
$P \cap T = \phi$ *and* $PUT \neq \phi$*.*

A Petri net with the initial marking is symbolized by $(PN,\ M_0)$.
A Petri net without any specific initial marking is symbolized by $PN\ =\ (P,\ T,\ F,\ W)$.

Definition 2.2. *[3] The marking of a Petri net (PN) is a function M from the set of places to the set of natural number, i,e.,* $M : P \to N$. *(A marked Petri net is a Petri net with an associated marking).*
A marking of a Petri net with n-places is a "$n\ \times 1$*" vector, which connected with each place a certain number of tokens represented by black dots, and represents a state of the Petri net.*
The initial marking is denoted by M_0.
"A change in net marking represents the change in the system which is a result of the occurrence of a sequence of transition firings".
Firing Rule:*"Firing an enabled transition is removes one token from each input place of the transition and adds one token to each output place of the transition".*

Definition 2.3. *[4] A transition* (t) *is said to be enabled if each input place P of* (t) *is marked with at least one token. A firing of an enabled transition* (t) *removes one tokens from each input place P of* (t)*, and one token to each output place P of* (t). *"An enabled transition may or may not fire (depending on whether or not the event actually takes place)".*

Definition 2.4. *[4] A transition* (t) *has no input place is called a source transition, and it unreservedly enabled. And a transition has no output place is called a sink transition.*
(The firing of a sink transition gets tokens, but does not produce anything).

Definition 2.5. *[4] A marked Petri net (PN) is said to be in deadlock if and only if no transition is enabled in that marking.*
A marked Petri net (PN) is said to be live if every transition can eventually be fired.

Definition 2.6. *[4] In PN, the input function* $I : (P \times T) \to N$ *is defined by*

$$I\left(P_i, t_j\right)\ =\ \begin{cases} 1, & if\ a\ directed\ arc\ from\ place\ P_i\ to\ transition\ t_j \\ 0, & if\ no\ arc\ from\ place\ P_i\ to\ transition\ t_j \end{cases}$$

The output function $O : (P \times T) \to N$ *is defined by*

$$O\left(P_i, t_j\right) = \begin{cases} 1, & if\ a\ directed\ arc\ from\ transition\ t_j\ to\ place\ P_i \\ 0, & if\ no\ arc\ from\ transition\ t_j\ to\ place\ P_i \end{cases}$$

Definition 2.7. *[4] The input and output places of transition t_i is defined by* $IP(t_i) = \{P_j \in P \;/\; I(P_j, t_i) \neq 0\}, OP(t_i) = \{P_j \in P \;/\; O(P_j, t_i) \neq 0\}$
The input and output transition of places P_j is defined by $IT(P_j) = \{t_i \in T \;/\; O(P_j, t_i) \neq 0\}, OP(P_j) = \{t_i \in T \;/\; I(P_j, t_i) \neq 0\}$

Definition 2.8. *[4] Timed Petri net is a seven-tuple,* $TPN = (P,\ T, I,\ O,\ M_0, M_1, \tau)$*, where,*

(1) *P is a set of places graphically represented by circles.*
(2) *T is a set of transitions graphically represented by bars, with* $P \cup T \neq 0$ *and* $P \cap T = 0$*.*
(3) $I : P \times T \to [0, 1]$ *is a function that specifies arcs going from transition to place.*
(4) $O : P \times T \to [0, 1]$ *is a function that specifies arcs going from places to transitions.*
(5) $M_0 : P \to \{0,\ 1,\ 2,\ 3,\ \ldots\}$ *is the initial marking,*
(6) M_t*-Initial timevector, ("The time left for tokens to enable the consistent output transition").*
(7) τ*-Time delays related with places.*

3. Petri Net representation of a Industrial system

Consider amanufacturing system having two machines M_1 and M_2 and three jobs, each job process through one phase of operation, which can be processed on either M_1 or M_2. On end of processing, the job is discharged from the machine and a fresh job is loaded into that machine. The following figure (Figure 1) express the Petri nets model of the said manufacturing system and following table (Table 1) provides the clarification of the places and transitions in this model.

3.1. **Simulating the Petri net(Petri Net Algorithm).** The steps agrees to the firing of transitions till the final state is encounter.

Step 1. Let the initial state marking M_0.
Step 2. Identify the enabled transitions:

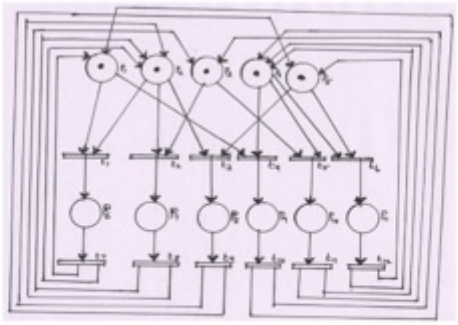

Figure1: Petri Nets with initial marking M_0

Places:	Transitions:
P_1: Job 1	t_1: M_1 starts processing the Job 1
P_2: Machine M_1 available	t_2: M_1 starts processing the Job 2
P_3: Job 2	t_3: M_1 starts processing the Job 3
P_4: Machine M_2 available	t_4: M_2 starts processing the Job 1
P_5: Job 3	t_5: M_2 starts processing the Job 2
P_6: M_1 processing the Job 1	t_6: M_2 starts processing the Job 3
P_7: M_1 processing the Job 2	t_7: M_1finishes processing the Job 1
P_8: M_1 processing the Job 3	t_8: M_1finishes processing the Job 2
P_9:M_2 processing the Job 1	t_9: M_1finishes processing the Job 3
P_{10}: M_2 processing the Job 2	t_{10}: M_2finishes processing the Job 1
P_{11}:M_2 processing the Job 3	t_{11}: M_2finishes processing the Job 2
	t_{12}: M_2finishes processing the Job 3

Table 1. Places and Transitions

$$
\begin{aligned}
F \;=\; & \{(P_1,t_1),(P_1,t_4),(P_2,t_1),(P_2,t_2),(P_2,t_3),(P_3,t_2),(P_3,t_5),(P_4,t_4),(P_4,t_5),(P_4,t_6),\\
& (P_5,t_3),(P_5,t_6),(P_6,t_7),(P_7,t_8),(P_8,t_9),(P_1,t_1),(P_9,t_10),(P_10,t_11),(P_11,t_12),(t_1,p_6),\\
& (t_2,P_7),(t_3,P_8),(t_4,P_9),(t_5,P_10),(t_6,(P_11),(t_7,P_1),(t_7,P_2),(t_8,P_2),(t_8,P_3),(t_9,P_2),\\
& (t_9,P_5),(t_10,P_1),(t_10,P_4),(t_11,P_3),(t_11,P_4),(t_12,P_4),(t_12,P_5)\}
\end{aligned}
$$

Step 3. Select a io transition
Step 4. Fireing Instruction.
4.1. Remove tokens from input places of transition t_j.
4.2. Calculate the time elapse for firing transition t_j.
4.3. Put tokens into the output places of transition t_j.
Step 5. Apprise the marking M_i and left over process time M_r vectors.
Step 6. -Carry on the system clock.
Step 7. If $M_i =$ final state then stop, else goto step 2.

4. SCHEDULING APPROACH

Scheduling is the activity of deciding which transition should fires, see [1]. The list of enabled transition rules aregiven below:

1. "SPT :(shortest processing time)".
2. "LPT :(longest processing time)".
3. "WSPT: (weighted shortest processing time)".
4. "EDD :(earliest due date)".
5. "CR :(critical ratio)".
6. "MS :(minimum slack)".
7. "ATC :(apparent tardiness cost)".

4.1. **Execution structure (For scheduling).** Create a PN model of the particular production strategy and generates a Gantt graph for showing the job sequences in the particular machines and decide the transition rule then:

(1) Develop PN objects.
(2) Develop object Scheduler $\&$ choose the firing transition rule.
(3) Run the PN- algorithm till it reaches the final state.
(4) Find the system pointers, Gantt graph and system routine. Measures,
   (i) Flow time for job in k^{th} positionis:
   $F_k = \sum_{i=1}^{k} P_i$, P_i-Processing time of the i^{th} Job.
   Mean Flow time for n jobs $\bar{F} = \frac{\sum_{i=1}^{n}(n-i+1)P_i}{n}$,
   Flow time (F_i) $=$ Finish time (F_i)- Job arrived time(F_i).
   (ii) Lateness of job i is defined by $L_i = C_i - d_i$,
   where C_i- finishing time of job i.
   d_i- due date of the job i.
   (iv) Tardiness of job i is defined by
   $T_i = \max(C_i - d_i,\ 0) = \max(L_i, 0)$

$$Total\ \ tardiness = \sum T_i\,.$$

   (iv) Makespan:"Supreme difference between the start and the finish time of every job"
   $Makespan\ =\ Max\ (C_1,\ C_2, \ldots C_n)\,,$

$$Makespan\ = \sum_i Flow\ time\ \ (F_i).$$

(v) System total time: Time span of the start of the 1stjoband the end of the last.
(vi) Resource utilization: Percentage of consumption time of each machine during the total system time.
(vii) Lateness and Flow time are indicators measure the service quality and the Makespan, system total time and Resources Utilization are describing the system's efficiency.

4.2. **Scheduling Jobs For One Station Flow Shop.** Any significance sequencing rule can be used to schedule any number of workstations. The specific reule used to scheduling of several jobs at a single workstation can be divide into two categories:
(1) "single-dimension rules",
(2) "multiple-dimension rules".

4.3. **Single -Dimensional rule.** The dispatching rules, for instance FCFS, EDD and SPT are called single-dimension rulesbecause they regulate priority based on a single characteristic of the job, for example arrival time, due date, processing time.

4.4. **Multiple-Dimension rules.** Priority rules, for instance CR and SRO are called multiple-dimension rulesbecause they relate to more than one characteristic of the job for examplejoindata about the residual workstations at which the job needto be processed, in addition to the processing time at the current workstation or the due date measured by single-dimension rules. Sequence the jobs in the lowest SRO index order.

Example 1. *Consider a job shop problem with five jobs and only one machine. The jobs processing times, due dates and current situation are tabulated below. And assume the machine allowed running hours is from 8:00 a.m. to 5:00 p.m. each weekday, plus holiday hours as needed. The Petri Net representations of the problem and the scheduling of the machine by using various dispatching rules are analysed. Per scheduling rule, the average flow time, average times early, and average times past due are calculated and compared the final results.*

Solution: Petri Net of the Problem:

Jobs	Jobs arrived Time	Processing time	Scheduled time (Due Date)	Time Remaining till Due date (days)	No. of operation Remaining	Shop time Remaining (days)
J_1	12	8	10	15	10	5.1
J_2	10	6	12	10	2	7.5
J_3	1	15	20	20	12	13.5
J_4	3	3	18	8	5	11.2
J_5	0	12	22	5	14	5.6

Table2.Job Scheduling for Manufacturing

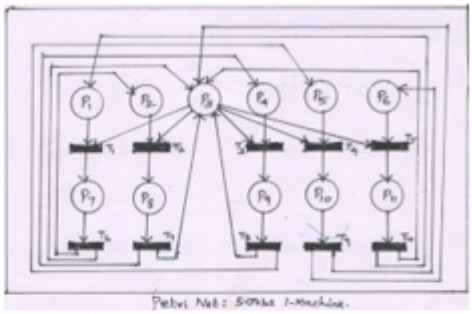

Figure 2:PetriNet

Places:	Transitions:
P_1: Job 1	T_1: M starts processing the Job 1
P_2: Job 2	T_2: M starts processing the Job 2
P_3: Machine M available	T_3: M starts processing the Job 3
P_4: Job 3	T_4: M starts processing the Job 4
P_5: Job 4	T_5: M starts processing the Job 5
P_6: Job 5	T_6: Mfinishes processing the Job 1
P_7: M processing the Job 1	T_7: M finishes processing the Job 2
P_8:M processing the Job 2	T_8: M finishes processing the Job 3
P_9:M processing the Job 3	T_9: M finishes processing the Job 4
P_{10}:M processing the Job 4	T_{10}: Mfinishes processing the Job 5
P_{11}:M processing the Job 5	

Consider the above Petri Net for all rules, with the places are taken as follows

Rules	P_1	P_2	P_3	P_4	P_5	P_6
FCFS	J_1	J_2	Machine	J_3	J_4	J_5
EDD	J_1	J_2	Machine	J_4	J_3	J_5
SPT	J_4	J_2	Machine	J_1	J_5	J_3
CR	J_4	J_5	Machine	J_2	J_3	J_1
S/RO	J_4	J_5	Machine	J_3	J_1	J_2

Table 3.Rules for Petrinet

I. **Scheduling by Single -Dimensional rules**

a) FCFS Rule: The jobs are sequence under they come in the order

b) EDD Rule : The jobs are sequence under the one with the closest due date, From the compared results, the EDD rule gives better result than the SPT

$J_1 - J_2 - J_3 - J_4 - J_5$.

Jobs	Jobs arrived time	Begin processing	Processing time	Finish time	Flow time	Scheduled time	Actual job processing ends	Hours early	Hours past due (Tardiness)
J_1	12	0	8	8	20	10	10	2	-
J_2	10	8	6	14	24	12	14	-	2
J_3	1	14	15	29	30	20	29	-	9
J_4	3	29	3	32	35	18	32	-	14
J_5	0	32	12	44	44	22	44	-	22

Table 4.FCFS Rule

$J_1 - J_2 - J_4 - J_3 - J_5$.

Jobs	Jobs arrived time	Begin processing	Processing time	Finish time	Flow time	Scheduled time	Actual job processing ends	Hours early	Hours past due
J_1	12	0	8	8	20	10	10	2	-
J_2	10	8	6	14	24	12	14	-	2
J_4	3	14	3	17	20	18	18	1	-
J_3	1	17	15	32	33	20	32	-	12
J_5	0	32	12	44	44	22	44	-	22

Table 5.EDD Rule

c). SPT Rule : The jobs are sequence under its shortest processing time.

$J_4 - J_2 - J_1 - J_5 - J_3$.

Jobs	Jobs arrived time	Begin processing	Processing time	Finish time	Flow time	Scheduled time	Actual job processing ends	Hours early	Hours past due
J_4	3	0	3	3	6	18	18	15	-
J_2	10	3	6	9	19	12	12	3	-
J_1	12	9	8	17	29	10	17	-	7
J_5	0	17	12	29	29	22	29	-	7
J_3	1	29	15	44	45	20	44	-	24

Table 6.SPT Rule

II. **Scheduling by Multi -Dimensional rules:**

Jobs	Jobs arrived Time	Processing time	Scheduled time (Due Date)	Time remaining until due-date(days)	Number of operation Remaining	Shop time Remaining (days)	CR	S/RO
J_1	12	8	10	15	10	5.1	2.94	0.99
J_2	10	6	12	10	2	7.5	1.33	1.25
J_3	1	15	20	20	12	13.5	1.48	0.54
J_4	3	3	18	8	5	11.2	0.71	-0.64
J_5	0	12	22	5	14	5.6	0.89	-0.04

Table 7.Scheduling by MultiDimensional Rules

and regarding average past dueFCFS rule best,but gives worse result regarding average flow time.

When S/RO rule is compare with EDD,CR rule is better, but it is much worse than the SPT, FCFS rule. Though, EDD, CR, and SRO all have the advantage of letting schedule changes when due dates change. We can test that, is these results can be generalized to other situations by processing n jobs.

d) *CR*Rule, The jobs are sequencing under its smallest *CR*-index:

$$J_4 - J_5 - J_2 - J_3 - J_1.$$

Jobs	Jobs arrived time	Begin processing	Processing time	Finish time	Flow time	Scheduled time	Actual job processing ends	Hours early	Hours past due
J_4	3	0	3	3	6	18	18	15	-
J_5	0	3	12	15	15	22	22	7	-
J_2	10	15	6	21	31	12	21	-	9
J_3	1	21	15	36	37	20	36	-	16
J_1	12	36	8	44	56	10	44	-	34

Table 8.Scheduling by CRRule

e) *S/RO*Rule:The jobs are sequencing under its smallest *S/RO* index

$$: J_4 - J_5 - J_3 - J_1 - J_2.$$

Jobs	Jobs arrived time	Begin processing	Processing time	Finish time	Flow time	Scheduled time	Actual job processing ends	Hours early	Hours past due
J_4	3	0	3	3	6	18	18	15	-
J_5	0	3	12	15	15	22	22	7	-
J_3	1	15	15	30	31	20	30	-	10
J_1	12	30	8	38	50	10	38	-	28
J_2	10	38	6	44	54	12	44	-	32

Table 9.Job Sequencing with S/RORule

COMPARED RESULTS:

PERFORMANCE MEASURES FOR 5 JOBS					
	FCFS	***EDD***	***SPT***	***CR***	***S/RO***
Average Flow time	30.6	28.2	25.6	29	31.2
Average hours early	0.3	0.6	3.6	4.4	4.4
Average hour past due	9.4	7.2	7.6	11.8	14
Makespan	44	44	44	44	44

Table 10.Performance Measures for 5Jobs

This example have exposed that SRO rule is better than EDD with regard to the percentage of jobs past due but worse than SPT and EDD with regard to average flow times,and also point out that CR results in extended flow times than SPT, but CR also results in fewer variance in the delivery of earlier due hours. Thus, even though the use of the multiple-dimension rules needs more information, no choice is obviously best.

4.5. **Scheduling Jobs For Two-Station Flow Shop.** Assume that a flow shop has more than a few jobs ready for processing at two workstations and that the course-plotting of all jobs are identical.

In the scheduling of two or more workstations in a flow shop, the makespandiffers according to the sequence preferred. Determining a manufacture order for a collection of jobs to minimize the makespan has two advantages:

(1) The collection of jobs are finished in least time.
(2) The usage of the two-station flow shop is maximized.
(3) Using the 1stworkstationconstantlytill it processes the last job reduces the idle time of the *second* workstation.

While modelling this example synchronization of the alternative routes of the two jobs and sharing resource are processed. And when sequencing the jobs it is essential to state job due dates and precedence, which gives the data about the machine's processing excellence and functional consistency.The Petri Net representation of this problem is given in the figure 3, in which, at first state tokens islocated at both start and resource places, sense that the system is prepared to initiate processing. "When the tokens is in each of the termination places P7 and P15 after a sequence of fired transitions, the simulation are finished".

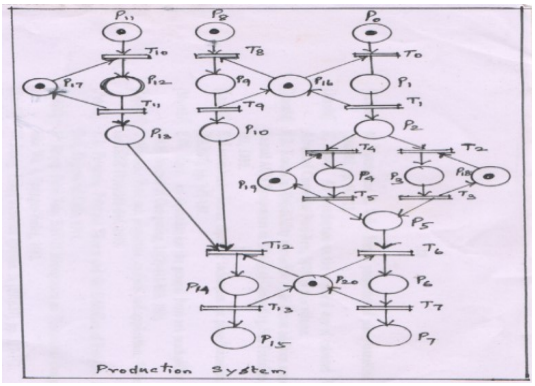

FIGURE 1. PetriNet representation

Here the dispatching rules of EDD and SPT are used; the Gantt graph and the system measures are tabulated in Table 11. "From the compared result table, the SPT rule needs a longer total time than EDD rule but the SPT rule shows just 1 late job while the EDD rule shows 2 late jobs. On the other hand, EDD only uses Cutter II and SPT only uses Cutter I. These results show that the use of different dispatching rules has a significant impact on the system performance".

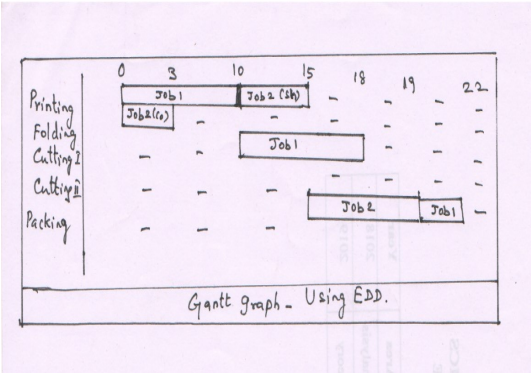

FIGURE 2. Cutting Model

Compared Results:

Performance measures	SPT	EDD
Makespan	21	22
Total weight tardiness	11	13
System total time	26	22
Late Jobs	1	2

Table 11.Performance Measures for SPT,EDD

5. CONCLUSIONS

In this work, the Petri-Nets representation to the manufacturing system is studied. The modelling of PN's in mixture with transition rules in scheduling isdiscussed, and also evaluates and comparesaltered schedules for the alike production strategy with example. Visualizedworkstation/machine utilization. Consumptions strength be avital decision's principles for selecting the greatest schedule.Tested altered options, varying parameters such as processing periods, production plan dimensions, alternative paths with suitable examples.Alsowe have proposed an algorithm called R_{max} intended for job sequencing, the result of the propsedmethod is either optimal or approximate to the existing methods. In multi jobs and machine problems, Johnson's rule is not validif the problem is converted into 2 jobs, whereas the proposed method valid for all types of problems.

REFERENCES

[1] K. R. BAKER: *Elements of Sequencing and Scheduling*, Hanover, NH: Baker Press, 2002.

[2] A. GUIA, F. DICESARE: *GRAFCET and Petri Nets in manufacturing systems, Intelligent Manufacturing: Programming Environments for CIM*, Springer Verlag, London, 1993, 153 - 176.

[3] D. DUBOIS, K. STECKE: *Using Petri Nets to represent production processes*, Proc of the 22nd IEEE Conf. on Decision and Control, San Antonio,TX, 1983, 1062 - 1067.

[4] J. EZPELETA, J. M. COLOM, J. MARTINEZ: *A Petri Net based deadlock prevention policy for flexible manufacturing systems*, IEEE Transactions onRobotics and Automation, **11**(2) (1995), 173–184.

[5] G. MEJIA, N. ODREY: *An approach using Petri Nets and improved heuristic search for manufacturing system scheduling*, Journal of ManufacturingSystems, **24**(2) (2003), 103 - 117.

[6] L. SHEN, Q. CHEN, J. Y. LUTH, Z. ZANGH: *Truncation of Petri Net models of scheduling problems for optimum solutions*, Proc. of Japan/USASymposium on Flexible Automation, 1992, 1681 - 1688.

[7] M. C. ZHOU, H. CHIU, H. H. XIONG: *Petri Net scheduling of FMS using branch and bound method*, Proc. of 1995 IEEE Int. Conf on Industrial Electronics, Control and Instrumentation. Orlando, FL, 1995, 211 - 216.

[8] M. C. ZHOU, F. DICESARE: *Petri Net synthesis for discrete event control of manufacturing systems*, Kluwer Academic Publisher, (USA), 1993, 187 - 188.

[9] M. C. ZHOU, M. VENKATESH: *Modeling, simulation and control of flexible manufacturing systems*, Intelligent Control and Intelligent Automation, **6** (1999), 43 - 44.

[10] J. SHIH, T. SEKIGUCHI: *A timed Petri Net andbeam search based on-line FMS scheduling systemwith routing flexibility*, Proc. of the 1991 IEEE Int.Conf. on Robotics and Automation. Sacramento, CA, 1991, 2548 - 2553.

[11] S. HAHMIAS: *Production and Operations analysis*, 3rd. Edition, Irwin/Mc. Graw-Hill, USA, , 1997, 404.

[12] M. PINEDO: *Scheduling: Theory, Algorithms,and Systems*, Prentice Hall, USA, 2002, 34 - 55.

[13] M. D. JENG: *Petri Nets for modelingautomatedmanufacturing systems with error recovery*, IEEETransactions on Robotics and Automation, **13**(5) (1997), 752–760.

Department of Mathematics
PSGR Krishnammal College for Women
Coimbatore -641004, TamilNadu, India
Email address: kavithakselvaraj@gmail.com

Department of Mathematics
Government Arts and Science College
Mettupalayam -641301, TamilNadu, India
Email address: babithadhana@yahoo.com