Effect of Optimal Sequencing in Flexible Manufacturing System on Physical representation & Analysis time using CPN

Muhammad Salman , Dr. Shaukat Ali Shah
Department of Mechanical Engineering
University of Engineering & Technology,
Peshawar, Pakistan

***Abstract*— For manufacturing system, there are some problems existing in practice when controlling it such as how jobs are sequenced to minimize job completion time, how production rates to be maximized, how jobs are dispatched to individual machines because flexible manufacturing system is a group of machines.**

This study focuses on simplification of the Petri net model of a manufacturing system for reduction in the total number of elementary circuits to decrease the analysis time of systems during investigations. A flexible manufacturing system with a tool sharing environment has been developed and investigated using colored Petri nets. The main objective of this study is to reduce the resultant invariants, as the addition of parts, machines, and tools in black token Petri net models exponentially increases the resultant invariants. Various scenarios have been investigated and compared using the developed model. The structure of the developed colored Petri net (CPN) model allows grouping of the tools that reduces the total number of places in the model. Therefore, it is evident that the developed CPN model reduces the resultant invariant that results in reduced complexity of the system.

With complex systems, monolithic models turn into impractical one and it becomes necessary to model them by dividing into subsystems and components. These components and subsystems are to be structured in a systematic manner to develop a control logic for them otherwise it also becomes complex. Also, most formal analysis methods can't be applied to very large petri net models because of the computational complexity. Hence the colored petri net comes up. Colored petri net can overcome these deficiencies because it is based on the idea that to reuse the petri net models topologies and to let the same petri net models contain more information.

The optimal sequence has very significant effect on the physical representation which results in a very simple graphical model the other benefit observed was that same graphical model can be used for next sequence and no conflict of events occurs. The other effect observed was reduction in analysis time of invariants. Colored Petri Nets have got their name because they allow the use of colored tokens that carry data values and can hence be distinguished from each other.

***Keywords*— Flexible Manufacturing System, Petri Nets, Colored Petri Nets (CPN), Linear Programming, Optimal Sequence.**

I. INTRODUCTION

In the current competitive world of manufacturing the product life has became short and there is a massive pressure from users on manufacturers to alter their products with passage of time to meet the current day needs. Manufacturing Companies are facing problems, to tackle with the frequent changes of product lines to launch new products in the market at economical price associated with operations of line flow. Scheduling is defined as the allocation of available resources over certain period of time to perform a collection of jobs [1]. Scheduling of manufacturing systems indicates the determination of the sequence in which jobs are to be processed over different production stages to be followed by the determination of the start time and finish time of processing jobs and overall task. The significance of Manufacturing System scheduling is to increase the deployment of the resources and to reduce the idle time and to minimize the in-process inventory [2]. In this paper sequencing of manufacturing system is done by means of Color Petri Nets. A Petri net is a bipartite directed diagram that contains places that is represented by circles, transitions which are represented by two parallel bars, and directed arcs connecting transitions and places to one another and also showing movement of parts to respective work station. Dynamic nature of the system modeled in Petri net is represented by the movement of tokens. A token represented as a small black dot is placed in a circle to indicate that the state is active. The existing location and allocation of tokens in a Petri net is called its marking. The marking of the Petri net defines the status of the system. Colored Petri Nets is a graphical oriented language for design, specification, simulation and verification of systems. Particularly it is suitable for systems in which communication, synchronization and resource sharing are important.

Tsukada and Shin [3] proposed the distributed tool sharing mechanism among various manufacturing stations for efficient utilization. To overcome unexpected tool sharing requirements, artificial intelligence was used for such situations. Xu and Randhawa [4] addressed the tool sharing policy on various job scheduling rules. Roh and Kim [5] categorized tool/part loading followed by part sequencing in order to enhance machine utilization and its productivity. To address tool duplication that often affects the plant productivity, Kashyap and Khator [6] implemented a logic and look-ahead policy for efficient tool transportation. To evaluate the FMS by analytical modeling, Tetzlaff [7] suggested an analytical model based on the principle of mean value approximation and classical convolution algorithm to model two interacting closed queuing networks. Amoako-Gyampah and Meredith [8] illustrated three heuristics for allocating tools in the FMS that were evaluated through a simulation study.

Machine tool loading problems were tackled by Atan and Pandit [9]. In their study, they proposed an approach that allowed the operations to be assigned to machines, assuming that machines had access to all the tools required for their operations. Further, a tool scheduling strategy was introduced by Li et al. [10]. Their method comprises integrating various schedule rules for tool exchange. Unevenly, machine load distribution reduces machine capacity and utilization that reduces the efficient operation of a flexible manufacturing system. Rau and Chetty [11] addressed the minimization of the unbalanced workload of machine issues by considering dynamic programming that involved part/pallet/fixture selection. Macchiaroli and Riemma [12] implemented multiple scheduling heuristics based on tool magazine capacity and its handling in an avionics industry. Grieco et al. [13] proposed efficient handling of tool allocation system based on a simulation technique to overcome tool cost by sharing the tools among several machining centers. Similarly, Barkaoui and Ben-Abdallah [14] suggested the modeling and analysis of the system in terms of stochastic Petri nets to quantify the effect of tool sharing policy on machine utilization and tool delivery time. Tool management is not only one of the most difficult aspects of FMS to regulate and control but also one of the most vital design issues. The key objective of this paper is to demonstrate the use of CPN as a new useful tool for modeling and evaluation of large FMSs. Such systems with a large number of machines and tools that deal with a tool sharing environment cannot be solved with black token Petri net models. Furthermore, in comparison to simulation and agent-based modeling, PN offers benefits, such as better graphical representation. In PN, work-in-process (WIP) control is more systematic while simulations are based on trials and errors. In addition, PN offers optimal solutions in most of the cases, whereas dynamic agent coordination is one of the challenging tasks in multi-agent research. CPNs also provide appropriate mathematical verification for the description, construction, and analysis of distributed and concurrent systems prior to its implementation [15].

II. PROBLEM STATEMENT

A. General statement

For flexible manufacturing system, there are some problems existing in practice when controlling it such as how jobs are sequenced to minimize job completion time, how production rates to be maximized, how jobs are dispatched to individual machines because Flexible Manufacturing System is a group of machines. Together all above issues to get the optimal solution is difficult. So Petri net is introduced to deal those problems.

In big FMSs, the graphical model becomes complex due to multiple parameters, such as tools, parts, and machines. Each tool will have its separate loop in the Petri net model which not only exponentially increases the resultant elementary circuits but also increases the complexity of the graphical representation of the model. The graphical complexity has considerably been reduced by introducing the idea of colored tokens. Colors are assigned to various parameters of the system that includes tools, machines, and parts, which made the model very much simpler, understandable, and demonstrable with reduced graphical complexities.

The key objective of the work is to plan the given Manufacturing System, modeling, optimizing and sequencing with Petri nets and Colored Petri nets and comparing both results in order to arrive at the best option with reduced analysis time, easily understandable, minimum conflict of events and physical representation. As per process plan each job has particular processing time, particular machining procedure in a particular way, which complicates the scheduling, so we employ Petri Nets to model, sequence and optimize the problem.

B. The case study

The case study is a simple example describes the proposed approach. The sample project consists of two parts, Two stations with station 1 having 6 machines and station two having 4 machines will be utilized for manufacturing the parts. Three (3) number of tools were utilized during the process as reported in **Table I**.

III. METHODOLOGY

A. Process Plan

Process plan is the main input to the Petri Net model for a manufacturing system. The process plan includes information about the type of parts to be manufactured, total number of processes, operations to be processed on each part, processing times of each part on each machine and the total number of tools to be used by the system and machines for producing those parts. The graphical Petri net model is being developed with the help of process plan, the model means transforming a Manufacturing System into a logical topology which represents parts, tools and workstations. Using the INA software all the invariants and total number of possible elementary circuits will be find out. The output which we get from the INA software will be then carefully entered in the Excel worksheet. The Excel worksheet is to be designed in a way that to calculate the total number of tokens in system, processing time of each part and cycle time of each elementary circuit. Afterward, according to critical circuit rules the constraints used in Lingo will be created. At last, to optimize the Work in Process (WIP) and Tool Inventory linear programming technique will be used. For the linear programming Lingo software will be used; the input data to the Lingo program will be the constraints from the Excel sheet and to get the desired parameters based on those constraints the Lingo will give us the optimal values. To recalculate the cycle time of each elementary circuit in the Excel sheet the output from Lingo will be used. The model is then analyzed for different sequences to find the effect of sequencing on total Work in Process and total number of tools.

Table I Process Plan: Two (2) Parts Two (2) Stations (P1P2)

Part	Station 1 (6)	Station 2 (4)
P1	t_{11} (10 min)	t_{12} (8 min)
	TL_{111}	TL_{121}, TL_{122}
P2	t_{21} (6 min)	t_{22} (5 min)
	TL_{213}	TL_{221}, TL_{222}

B. Model Illustration

The methodology is illustrated in three steps, in the first step the system is modeled as a black token timed Petri net model in which black tokens are used for the parts, machines and tools. In the next step, the tool loops are represented through colored tokens and being analyzed and then compared with the black token PN model. Finally the graphical model is further simplified by using colored tokens for the tool, machines and parts as well. The three models are then analyzed and compared.

Figure I represents the PN model of an example FMS with a process plan given in *Table I*, the part place P_{11} contains the first part that has been unloaded from the load/unload station and is ready for the first operation, this part after finishing will be unloaded and will leave the system after t_{12}. Second part will follow the same pattern i.e. it will be processed by t_{21} when it is unloaded on place P_{21} and will leave the system after being processed on machine t_{22}.

The transition t_{11} can start processing a part if there is at least one token in the places P_{11}, T_{111}, T_{112}, T_{113} and C_{11}, which are the input places to t_{11}. The token in the place P_{11} represent a part ready for processing. Token in the places T_{111}, T_{112} and T_{113} represent the availability of tools, similarly, token in the place C_{11} represent the availability of machine at station 1. The number of tokens in the loop C_{11}, t_{11}, C_{21} and t_{21} represents the number of identical machines or servers of the station 1.

After the t11 fires, token will be added to each of its output places i.e. one token each to the places P_{12}, C_{21}, T_{121}, T_{122}, and T_{213}. The three tools are now available for other servers and can be shared among machine t_{12}, t_{21}, and t_{22}. In the next step transitions t_{12} and t_{21} can fire simultaneously, as input conditions to both the transitions are fulfilled. According to the process plan transition t_{12} needs tool T_{121}, T_{122} i.e. tool number 1 and 2, while transition t_{21} needs tool T_{213} i.e. tool number 3, hence both the transition will fire at the same time. After t_{12} and t_{21}fires, tokens will be added to each of its output places i.e. one token each to the place P_{11}, P_{22}, C_{11} and C_{22}, tokens for the first and second tool will be added to the place T_{221} and T_{222}, similarly token for tool number three will be added the place T_{113} i.e. starting place of the tool. In the last transition t_{21} will fire the tokens in the place T_{221}, T_{222}, P_{22} and C_{22} and the system will approach to the start point and will start processing the parts in the same manner as it is cyclic in behavior.

In case a tool is requested by two machines at the same time, then the service discipline used in the model is FIFO i.e. First In First Serve basis. In addition to that, the system is also capable of keeping track of the tool life of each tool in terms of the total processing time a tool has served in processing the work pieces at each station. This can be done by adding the total processing times of the transitions the tool has passed through. Such as, for the given process plan the total processing time of tool number one is 18 minutes in one cycle.

C. Elementary Circuits

An elementary circuit γc, in a Petri Net model is a directed path that starts from one node, place or transition and comes back to this same node in such a manner that no other nodes are repeated and always progress in the direction of the arcs. Elementary circuits can easily be identified, however, mixed circuits that pass through part and sequencing and or pass through part/sequencing and tools are very complex to find and need an organized approach to be identified. Some of the elementary circuits can be easily found
in the generic model.

- Part circuit ($\gamma^{c}{}_{p}$) Pij tij
- Tool circuit ($\gamma^{c}{}_{T}$) Tijk tij
- Part/Tool circuit ($\gamma^{c}{}_{PT}$) Pij tij Tijk
- Sequencing circuit ($\gamma^{c}{}_{s}$) Cij tij

The prefix "c" in γ^{c} represent the color of the token in that particular circuit, since each γ will have different cycle time for each color. The whole transition time $\tau(\gamma^{c})$ in each elementary circuit γ can be obtained as the sum of the transition firing times $t^{c}{}_{ij}$ in that circuit. M(γ) refers to the total sum of tokens in the elementary circuit γ. The cycle time C(γ) of every one elementary circuit is the ratio τ(γ)/M(γ). The biggest cycle time $C(\gamma_s)$ of the sequencing circuits γs determine the capacity of the system for a given number of identical machines represented by $M(\gamma_s)$. τ(γs) represents the total processing time required from a single machine during one production cycle in which one unit of each part type *i* is machined. **Fig.3** shows all the elementary circuits for **Fig.1**. These elementary circuits are also known as the semi positive invariants of the PN in the literature.

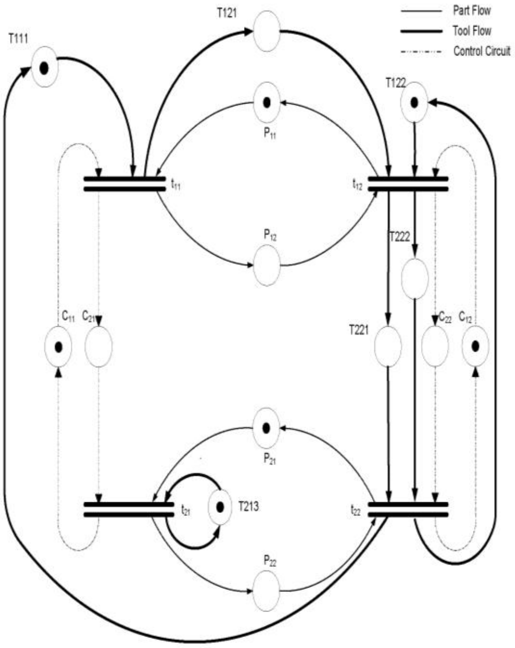
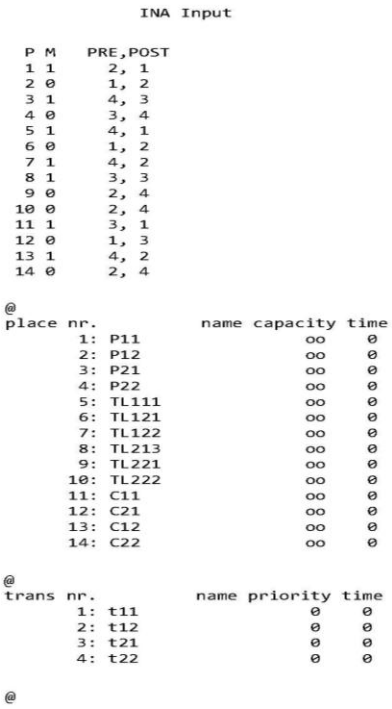


Fig.1 Petri Net Model Diagram

D. Integrated Net Analyzer (INA)

After having the Petri Net model, the next step is to determine all the possible elementary circuits that are called Invariants. In case of small system it is easy to find the invariants manually by selecting the possible circuits from the Petri Net diagram, but for complex systems having many machines with many parts, it become very difficult to do the same manually. For this purpose different softwares are available that can be used to find out the total number of possible invariants in a Petri Net model. INA is one of the freely available software that can solve such problems.

INA software will be used to find out the possible number of invariants, from the Petri Net diagram, a notepad file is being prepared as shown in **Fig.2** below which will be used as an INPUT file to the INA software.



```
                INA Input

  P M    PRE,POST
  1 1      2, 1
  2 0      1, 2
  3 1      4, 3
  4 0      3, 4
  5 1      4, 1
  6 0      1, 2
  7 1      4, 2
  8 1      3, 3
  9 0      2, 4
 10 0      2, 4
 11 1      3, 1
 12 0      1, 3
 13 1      4, 2
 14 0      2, 4

@
place nr.            name capacity time
       1: P11                oo     0
       2: P12                oo     0
       3: P21                oo     0
       4: P22                oo     0
       5: TL111              oo     0
       6: TL121              oo     0
       7: TL122              oo     0
       8: TL213              oo     0
       9: TL221              oo     0
      10: TL222              oo     0
      11: C11                oo     0
      12: C21                oo     0
      13: C12                oo     0
      14: C22                oo     0

@
trans nr.           name priority time
       1: t11                0     0
       2: t12                0     0
       3: t21                0     0
       4: t22                0     0

@
```

Fig.2 INA input file

The output from INA showing possible elementary circuits is given in ***Fig.3****.*

Rules for the proposed algorithm can be formulated as shown below:

$\forall\ (P_{ij}\ ,\ C_{ij})\ \in\ y \qquad y\ \in\ elementary\ circuit$

For j = 1, 2, 3, ….. n

And IF

$\sum\ P_{ij} = 0\ \rightarrow\ C_{ij} = \overline{C}_{ij}$ yields

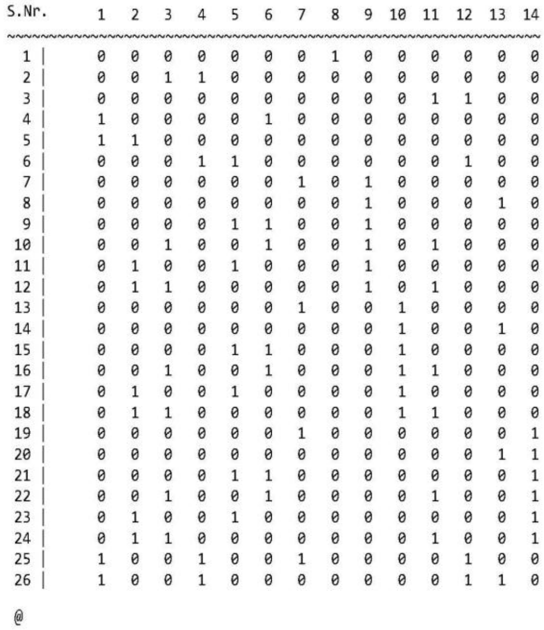
If pre and post values of both transitions are changed

$\sum\ P_{ij} = 0\ \rightarrow\ C_{i1} = \overline{C}_{i1}$ yields

If pre and post values of both transitions are changed

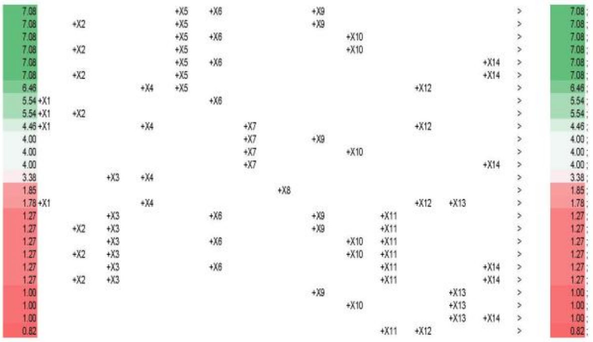
$\sum\ P_{ij} = 0\ \rightarrow\ C_{i2} = \overline{C}_{i2}$ yields

If pre and post values of both transitions are changed



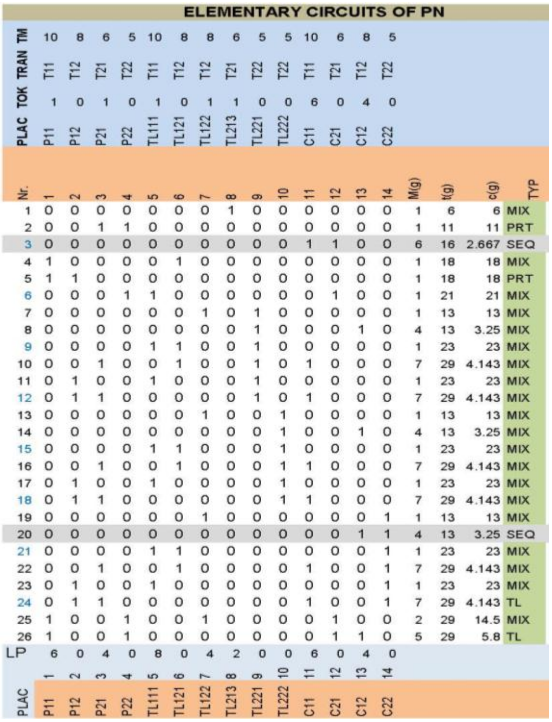
```
S.Nr.      1  2  3  4  5  6  7  8  9 10 11 12 13 14
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   1 |     0  0  0  0  0  0  0  1  0  0  0  0  0  0
   2 |     0  0  1  1  0  0  0  0  0  0  0  0  0  0
   3 |     0  0  0  0  0  0  0  0  0  0  1  1  0  0
   4 |     1  0  0  0  0  1  0  0  0  0  0  0  0  0
   5 |     1  1  0  0  0  0  0  0  0  0  0  0  0  0
   6 |     0  0  0  1  1  0  0  0  0  0  0  1  0  0
   7 |     0  0  0  0  0  0  1  0  1  0  0  0  0  0
   8 |     0  0  0  0  0  0  0  0  1  0  0  0  1  0
   9 |     0  0  0  0  1  1  0  0  1  0  0  0  0  0
  10 |     0  0  1  0  0  1  0  0  1  0  1  0  0  0
  11 |     0  1  0  0  1  0  0  0  1  0  0  0  0  0
  12 |     0  1  1  0  0  0  0  0  1  0  1  0  0  0
  13 |     0  0  0  0  0  0  1  0  0  1  0  0  0  0
  14 |     0  0  0  0  0  0  0  0  0  1  0  0  1  0
  15 |     0  0  0  0  1  1  0  0  0  1  0  0  0  0
  16 |     0  0  1  0  0  1  0  0  0  1  1  0  0  0
  17 |     0  1  0  0  1  0  0  0  0  1  0  0  0  0
  18 |     0  1  1  0  0  0  0  0  0  1  1  0  0  0
  19 |     0  0  0  0  0  0  1  0  0  0  0  0  0  1
  20 |     0  0  0  0  0  0  0  0  0  0  0  0  1  1
  21 |     0  0  0  0  1  1  0  0  0  0  0  0  0  1
  22 |     0  0  1  0  0  1  0  0  0  0  1  0  0  1
  23 |     0  1  0  0  1  0  0  0  0  0  0  0  0  1
  24 |     0  1  1  0  0  0  0  0  0  0  1  0  0  1
  25 |     1  0  0  1  0  0  1  0  0  0  0  1  0  0
  26 |     1  0  0  1  0  0  0  0  0  0  0  1  1  0

@
```

Figure III INA output showing possible number of invariants



7.08					+X5	+X6			+X9						>	7.08 ;
7.08		+X2			+X5				+X9						>	7.08 ;
7.08					+X5	+X6				+X10					>	7.08 ;
7.08		+X2			+X5					+X10					>	7.08 ;
7.08					+X5	+X6								+X14	>	7.08 ;
7.08		+X2			+X5									+X14	>	7.08 ;
6.46				+X4	+X5							+X12			>	6.46 ;
5.54	+X1					+X6									>	5.54 ;
5.54	+X1	+X2													>	5.54 ;
4.46	+X1			+X4			+X7					+X12			>	4.46 ;
4.00							+X7		+X9						>	4.00 ;
4.00							+X7			+X10					>	4.00 ;
4.00							+X7							+X14	>	4.00 ;
3.38			+X3	+X4											>	3.38 ;
1.85								+X8							>	1.85 ;
1.78	+X1			+X4								+X12	+X13		>	1.78 ;
1.27			+X3			+X6			+X9		+X11				>	1.27 ;
1.27		+X2	+X3						+X9		+X11				>	1.27 ;
1.27			+X3			+X6				+X10	+X11				>	1.27 ;
1.27		+X2	+X3							+X10	+X11				>	1.27 ;
1.27			+X3			+X6					+X11			+X14	>	1.27 ;
1.27		+X2	+X3								+X11			+X14	>	1.27 ;
1.00									+X9				+X13		>	1.00 ;
1.00										+X10			+X13		>	1.00 ;
1.00													+X13	+X14	>	1.00 ;
0.82											+X11	+X12			>	0.82 ;

Fig.4 Variables sheet for lingo model formation



ELEMENTARY CIRCUITS OF PN

TM	10	8	6	5	10	8	8	6	5	5	10	6	8	5				
TRAN	T11	T12	T21	T22	T11	T12	T12	T21	T22	T22	T11	T21	T12	T22				
TOK	1	0	1	0	1	0	1	1	0	0	6	0	4	0				
PLAC	P11	P12	P21	P22	TL111	TL121	TL122	TL213	TL221	TL222	C11	C21	C12	C22				
Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	M(g)	t(g)	c(g)	TYP
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	6	6	MIX
2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	11	11	PRT
3	0	0	0	0	0	0	0	0	0	0	1	1	0	0	6	16	2.667	SEQ
4	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	18	18	MIX
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	18	18	PRT
6	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	21	21	MIX
7	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	13	13	MIX
8	0	0	0	0	0	0	0	0	1	0	0	0	1	0	4	13	3.25	MIX
9	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	23	23	MIX
10	0	0	1	0	0	1	0	0	1	0	1	0	0	0	7	29	4.143	MIX
11	0	1	0	0	1	0	0	0	1	0	0	0	0	0	1	23	23	MIX
12	0	1	1	0	0	0	0	0	1	0	1	0	0	0	7	29	4.143	MIX
13	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	13	13	MIX
14	0	0	0	0	0	0	0	0	0	1	0	0	1	0	4	13	3.25	MIX
15	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	23	23	MIX
16	0	0	1	0	0	1	0	0	0	1	1	0	0	0	7	29	4.143	MIX
17	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	23	23	MIX
18	0	1	1	0	0	0	0	0	0	1	1	0	0	0	7	29	4.143	MIX
19	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	13	13	MIX
20	0	0	0	0	0	0	0	0	0	0	0	0	1	1	4	13	3.25	SEQ
21	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	23	23	MIX
22	0	0	1	0	0	1	0	0	0	0	1	0	0	1	7	29	4.143	MIX
23	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	23	23	MIX
24	0	1	1	0	0	0	0	0	0	0	1	0	0	1	7	29	4.143	TL
25	1	0	0	1	0	0	1	0	0	0	0	1	0	0	2	29	14.5	MIX
26	1	0	0	1	0	0	0	0	0	0	0	1	1	0	5	29	5.8	TL
LP	6	0	4	0	8	0	4	2	0	0	6	0	4	0				
	1	2	3	4	5	6	7	8	9	10	11	12	13	14				
PLAC	P11	P12	P21	P22	TL111	TL121	TL122	TL213	TL221	TL222	C11	C21	C12	C22				

Fig.5 Resultant excel sheet showing the cycle times

E. Excel sheet formulation

To analyze the cycle time of each invariant (i.e. all the elementary circuits), the output from INA software is imported to the Microsoft Excel. The Microsoft Excel file is then programmed accordingly so that it can automatically determine the total number of tokens, total processing times and the cycle times of each invariant. Resultant Microsoft Excel file showing the cycle times of all the invariants is shown in **Fig.4**. Another sheet is programmed for lingo optimization model variable part as shown in **Fig.5**.

F. Lingo formulation

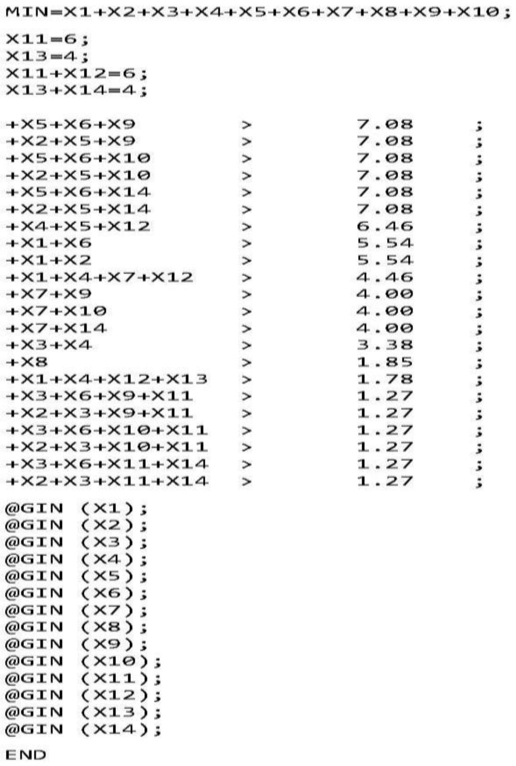
An optimization model consists of three parts:

Objective function – This is single formula that describes exactly what the model should optimize. A general manufacturing example of an objective function would be to minimize the cycle time for a given product.

Variables – These are the quantities that can be changed to produce the optimal value of the objective function. For example, when driving a car, the duration of the trip (t) and the speed at which it is taken (v) determine the distance (d) that can be traveled.

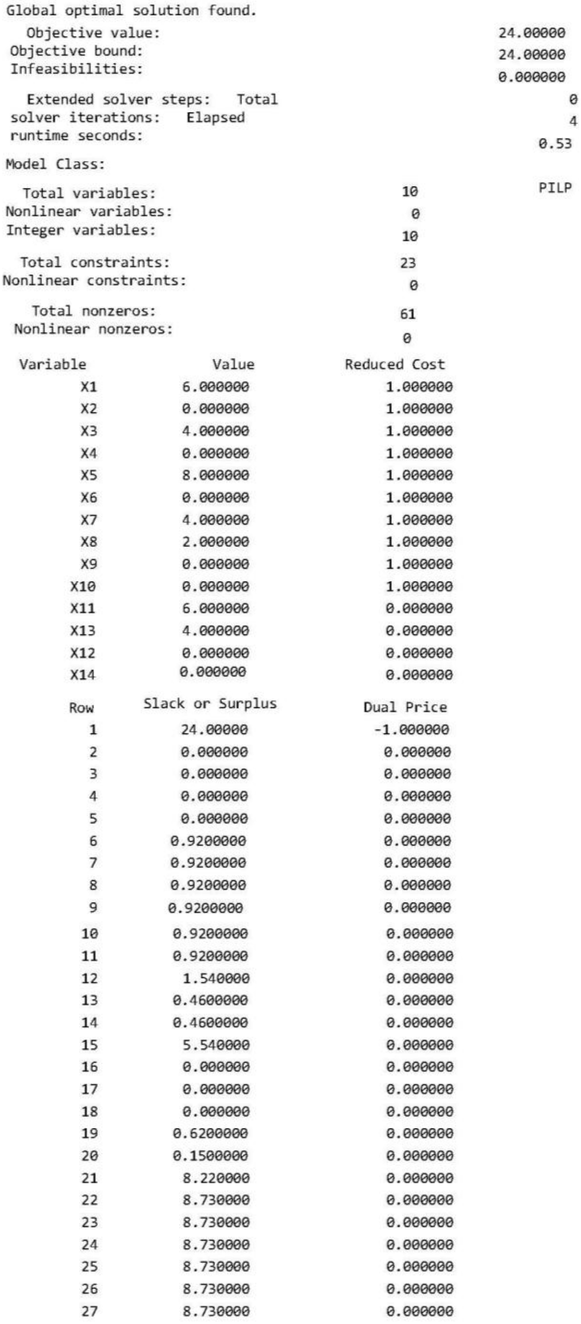
Constraints – These are formulas that define the limits on the values of the variables. If an ice cream store is determining how many flavors it should offer, only a positive number of flavors is feasible

Optimization model shown in **Fig.6**, consist of first Part as objective function, second part is our constraints and third part is variables. @GIN command is used for rounding off the data.



```
MIN=X1+X2+X3+X4+X5+X6+X7+X8+X9+X10;

X11=6;
X13=4;
X11+X12=6;
X13+X14=4;

+X5+X6+X9          >        7.08     ;
+X2+X5+X9          >        7.08     ;
+X5+X6+X10         >        7.08     ;
+X2+X5+X10         >        7.08     ;
+X5+X6+X14         >        7.08     ;
+X2+X5+X14         >        7.08     ;
+X4+X5+X12         >        6.46     ;
+X1+X6             >        5.54     ;
+X1+X2             >        5.54     ;
+X1+X4+X7+X12      >        4.46     ;
+X7+X9             >        4.00     ;
+X7+X10            >        4.00     ;
+X7+X14            >        4.00     ;
+X3+X4             >        3.38     ;
+X8                >        1.85     ;
+X1+X4+X12+X13     >        1.78     ;
+X3+X6+X9+X11      >        1.27     ;
+X2+X3+X9+X11      >        1.27     ;
+X3+X6+X10+X11     >        1.27     ;
+X2+X3+X10+X11     >        1.27     ;
+X3+X6+X11+X14     >        1.27     ;
+X2+X3+X11+X14     >        1.27     ;

@GIN (X1);
@GIN (X2);
@GIN (X3);
@GIN (X4);
@GIN (X5);
@GIN (X6);
@GIN (X7);
@GIN (X8);
@GIN (X9);
@GIN (X10);
@GIN (X11);
@GIN (X12);
@GIN (X13);
@GIN (X14);

END
```

Fig.6 Lingo input file

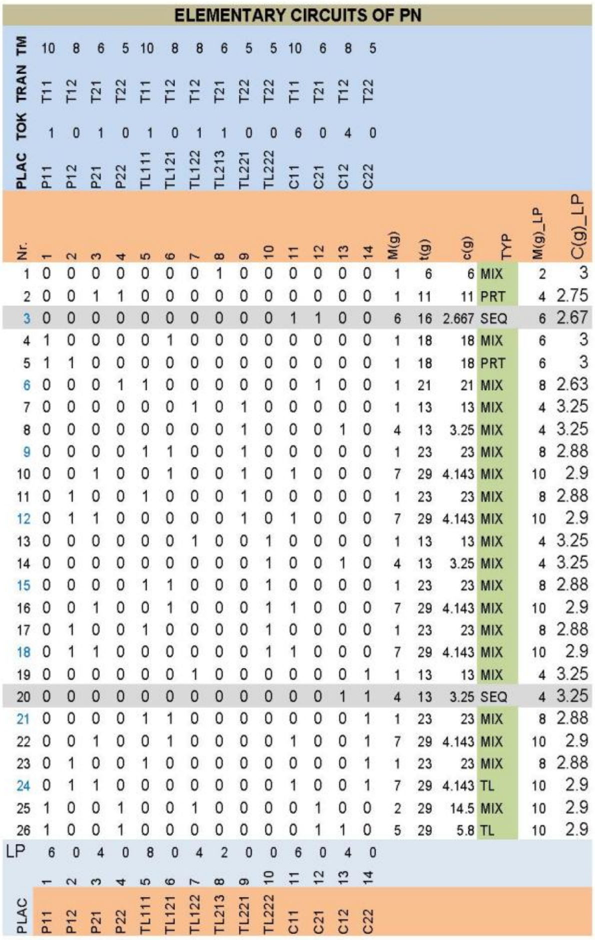


```
Global optimal solution found.
  Objective value:                                          24.00000
 Objective bound:                                           24.00000
 Infeasibilities:                                           0.000000

  Extended solver steps:   Total                                   0
 solver iterations:   Elapsed                                      4
 runtime seconds:                                               0.53
Model Class:

  Total variables:                       10                     PILP
Nonlinear variables:                      0
Integer variables:                       10

  Total constraints:                     23
Nonlinear constraints:                    0

   Total nonzeros:                       61
 Nonlinear nonzeros:                      0

  Variable          Value          Reduced Cost
        X1        6.000000          1.000000
        X2        0.000000          1.000000
        X3        4.000000          1.000000
        X4        0.000000          1.000000
        X5        8.000000          1.000000
        X6        0.000000          1.000000
        X7        4.000000          1.000000
        X8        2.000000          1.000000
        X9        0.000000          1.000000
       X10        0.000000          1.000000
       X11        6.000000          0.000000
       X13        4.000000          0.000000
       X12        0.000000          0.000000
       X14        0.000000          0.000000

       Row     Slack or Surplus      Dual Price
         1        24.00000          -1.000000
         2        0.000000           0.000000
         3        0.000000           0.000000
         4        0.000000           0.000000
         5        0.000000           0.000000
         6        0.9200000          0.000000
         7        0.9200000          0.000000
         8        0.9200000          0.000000
         9        0.9200000          0.000000
        10        0.9200000          0.000000
        11        0.9200000          0.000000
        12        1.540000           0.000000
        13        0.4600000          0.000000
        14        0.4600000          0.000000
        15        5.540000           0.000000
        16        0.000000           0.000000
        17        0.000000           0.000000
        18        0.000000           0.000000
        19        0.6200000          0.000000
        20        0.1500000          0.000000
        21        8.220000           0.000000
        22        8.730000           0.000000
        23        8.730000           0.000000
        24        8.730000           0.000000
        25        8.730000           0.000000
        26        8.730000           0.000000
        27        8.730000           0.000000
```

Fig.7 Lingo output with optimized values

The input file shown in **Fig.6** has been executed and the resultant output file from the LINGO software is shown in **Fig.7**. The value of the objective function i.e. the WIP is found as 10 and tools as 14 and the optimum number of tokens at P_{ij}.

After getting the optimum number of tokens at P_{ij}, these results are being inputted in the LP row of the Microsoft Excel file and the cycle times are recalculated with this optimum number of tokens for each invariant. The optimum cycle time is represented as C(g)_LP.



ELEMENTARY CIRCUITS OF PN

TM	10	8	6	5	10	8	8	6	5	5	10	6	8	5						
TRAN	T11	T12	T21	T22	T11	T12	T12	T21	T22	T22	T11	T21	T12	T22						
TOK	1	0	1	0	1	0	1	1	0	0	6	0	4	0						
PLAC	P11	P12	P21	P22	TL111	TL121	TL122	TL213	TL221	TL222	C11	C21	C12	C22						
Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	M(g)	t(g)	c(g)	TYP	M(g)_LP	C(g)_LP
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	6	6	MIX	2	3
2	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	11	11	PRT	4	2.75
3	0	0	0	0	0	0	0	0	0	0	1	1	0	0	6	16	2.667	SEQ	6	2.67
4	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	18	18	MIX	6	3
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	18	18	PRT	6	3
6	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	21	21	MIX	8	2.63
7	0	0	0	0	0	0	1	0	1	0	0	0	0	0	1	13	13	MIX	4	3.25
8	0	0	0	0	0	0	0	0	1	0	0	0	1	0	4	13	3.25	MIX	4	3.25
9	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	23	23	MIX	8	2.88
10	0	0	1	0	0	1	0	0	1	0	1	0	0	0	7	29	4.143	MIX	10	2.9
11	0	1	0	0	1	0	0	0	1	0	0	0	0	0	1	23	23	MIX	8	2.88
12	0	1	1	0	0	0	0	0	1	0	1	0	0	0	7	29	4.143	MIX	10	2.9
13	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	13	13	MIX	4	3.25
14	0	0	0	0	0	0	0	0	0	1	0	0	1	0	4	13	3.25	MIX	4	3.25
15	0	0	0	0	1	1	0	0	0	1	0	0	0	0	1	23	23	MIX	8	2.88
16	0	0	1	0	0	1	0	0	0	1	1	0	0	0	7	29	4.143	MIX	10	2.9
17	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	23	23	MIX	8	2.88
18	0	1	1	0	0	0	0	0	0	1	1	0	0	0	7	29	4.143	MIX	10	2.9
19	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	13	13	MIX	4	3.25
20	0	0	0	0	0	0	0	0	0	0	0	0	1	1	4	13	3.25	SEQ	4	3.25
21	0	0	0	0	1	1	0	0	0	0	0	0	0	1	1	23	23	MIX	8	2.88
22	0	0	1	0	0	1	0	0	0	0	1	0	0	1	7	29	4.143	MIX	10	2.9
23	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1	23	23	MIX	8	2.88
24	0	1	1	0	0	0	0	0	0	0	1	0	0	1	7	29	4.143	TL	10	2.9
25	1	0	0	1	0	0	1	0	0	0	0	1	0	0	2	29	14.5	MIX	10	2.9
26	1	0	0	1	0	0	0	0	0	0	0	1	1	0	5	29	5.8	TL	10	2.9
LP	6	0	4	0	8	0	4	2	0	0	6	0	4	0						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14						
PLAC	P11	P12	P21	P22	TL111	TL121	TL122	TL213	TL221	TL222	C11	C21	C12	C22						

Fig.8 Resultant Excel file with optimized cycle times

G. Corresponding CPN *Model*

The drawback of using black token PN model for the tools is that for each additional tool, machine or part we have to include a new loop for every new addition in the model, hence an increase in the tools, machine or part will make the model more and more complex. This can be simplified by assigning colors to the tokens, which reduces the number of loops. **Fig.9** shows the corresponding CPN model for the same process plan with the colors assigned to tools only. Methodology for optimization in CPN is same as in Black Petri Nets.

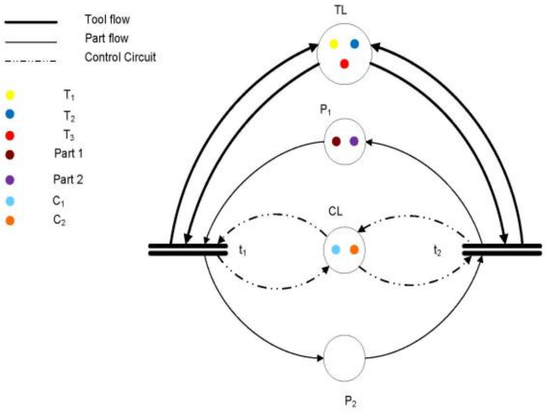
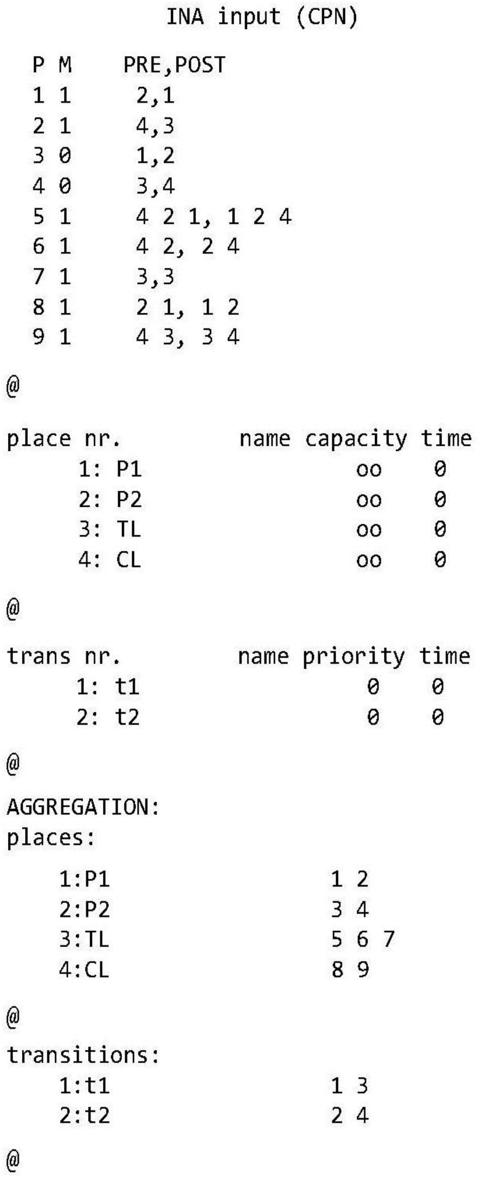
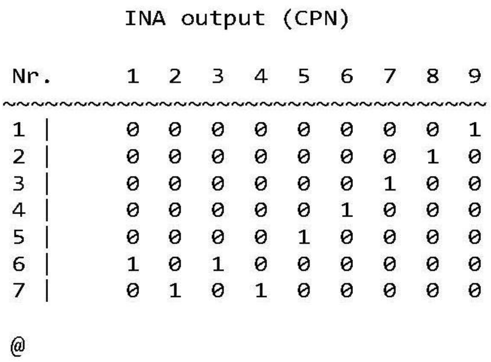


Fig.9 Color Petri Net Model



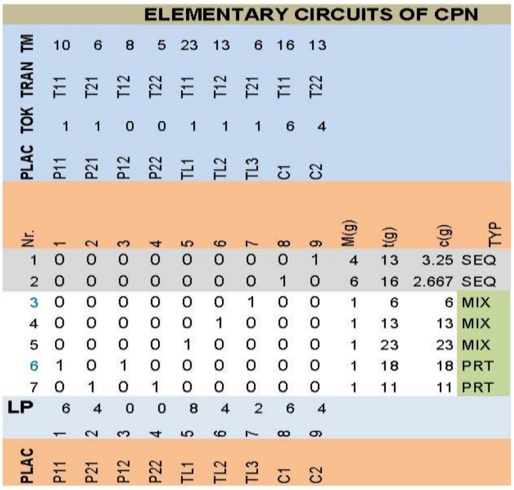
```
                INA input (CPN)
P M     PRE,POST
1 1      2,1
2 1      4,3
3 0      1,2
4 0      3,4
5 1      4 2 1, 1 2 4
6 1      4 2, 2 4
7 1      3,3
8 1      2 1, 1 2
9 1      4 3, 3 4

@

place nr.         name capacity time
     1: P1                 oo    0
     2: P2                 oo    0
     3: TL                 oo    0
     4: CL                 oo    0

@

trans nr.         name priority time
     1: t1                  0    0
     2: t2                  0    0

@

AGGREGATION:
places:

    1:P1                1 2
    2:P2                3 4
    3:TL                5 6 7
    4:CL                8 9

@

transitions:
    1:t1                1 3
    2:t2                2 4

@
```

Fig.10 INA Input



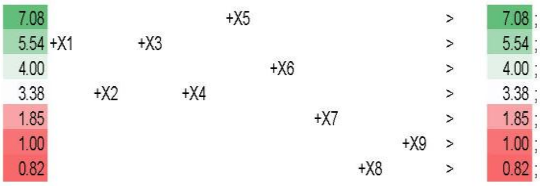
```
        INA output (CPN)

Nr.      1  2  3  4  5  6  7  8  9
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
1 |      0  0  0  0  0  0  0  0  1
2 |      0  0  0  0  0  0  0  1  0
3 |      0  0  0  0  0  0  1  0  0
4 |      0  0  0  0  0  1  0  0  0
5 |      0  0  0  0  1  0  0  0  0
6 |      1  0  1  0  0  0  0  0  0
7 |      0  1  0  1  0  0  0  0  0

@
```

Fig.11 INA output file



ELEMENTARY CIRCUITS OF CPN													
TM	10	6	8	5	23	13	6	16	13				
TRAN	T11	T21	T12	T22	T11	T12	T21	T11	T22				
TOK	1	1	0	0	1	1	1	6	4				
PLAC	P11	P21	P12	P22	TL1	TL2	TL3	C1	C2				
Nr.	1	2	3	4	5	6	7	8	9	M(g)	t(g)	c(g)	TYP
1	0	0	0	0	0	0	0	0	1	4	13	3.25	SEQ
2	0	0	0	0	0	0	0	1	0	6	16	2.667	SEQ
3	0	0	0	0	0	0	1	0	0	1	6	6	MIX
4	0	0	0	0	0	1	0	0	0	1	13	13	MIX
5	0	0	0	0	1	0	0	0	0	1	23	23	MIX
6	1	0	1	0	0	0	0	0	0	1	18	18	PRT
7	0	1	0	1	0	0	0	0	0	1	11	11	PRT
LP	6	4	0	0	8	4	2	6	4				
	1	2	3	4	5	6	7	8	9				
PLAC	P11	P21	P12	P22	TL1	TL2	TL3	C1	C2				

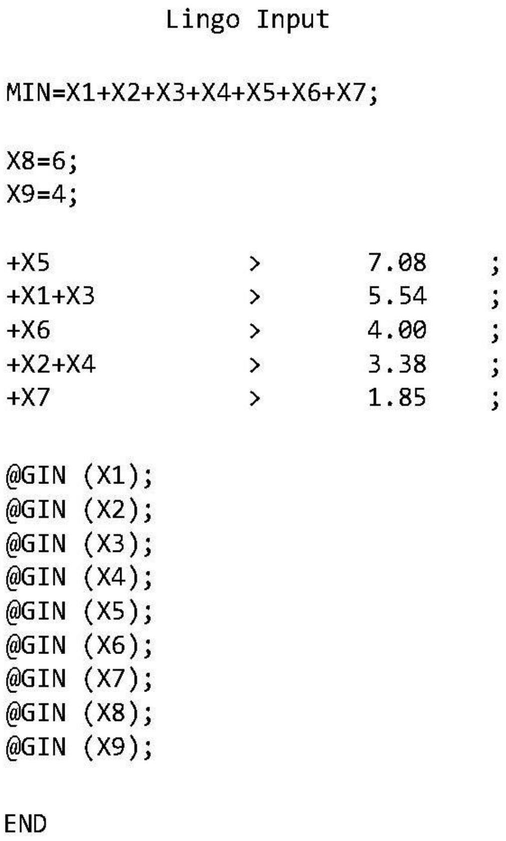
Fig.12 Resultant excel sheet showing the cycle times



7.08					+X5					>	7.08 ;
5.54	+X1		+X3							>	5.54 ;
4.00						+X6				>	4.00 ;
3.38		+X2		+X4						>	3.38 ;
1.85							+X7			>	1.85 ;
1.00									+X9	>	1.00 ;
0.82								+X8		>	0.82 ;

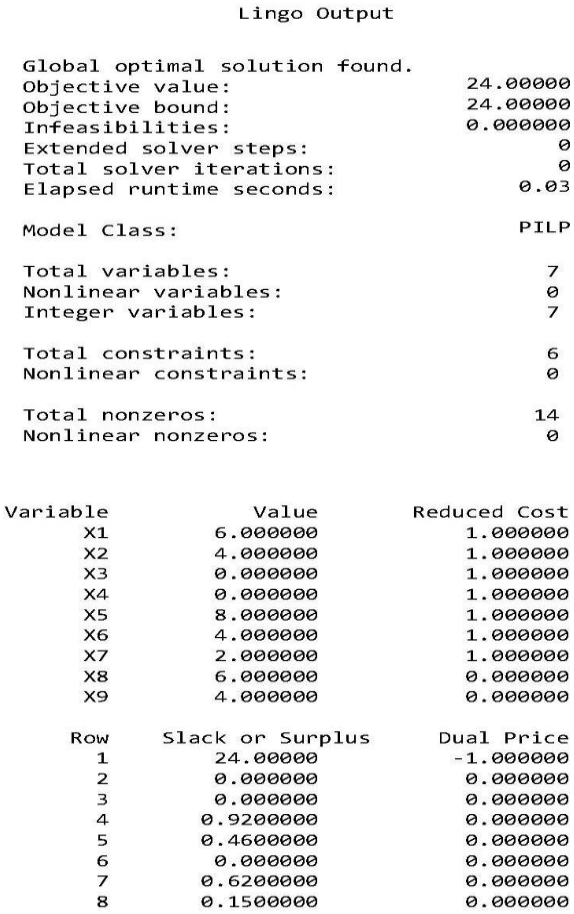
Fig.13 Variables sheet for lingo model formation

Fig.15 Lingo output with optimized values



```
Lingo Input

MIN=X1+X2+X3+X4+X5+X6+X7;

X8=6;
X9=4;

+X5                >        7.08    ;
+X1+X3             >        5.54    ;
+X6                >        4.00    ;
+X2+X4             >        3.38    ;
+X7                >        1.85    ;

@GIN (X1);
@GIN (X2);
@GIN (X3);
@GIN (X4);
@GIN (X5);
@GIN (X6);
@GIN (X7);
@GIN (X8);
@GIN (X9);

END
```

Fig.14 Lingo software input



```
                    Lingo Output

 Global optimal solution found.
 Objective value:                              24.00000
 Objective bound:                              24.00000
 Infeasibilities:                              0.000000
 Extended solver steps:                               0
 Total solver iterations:                             0
 Elapsed runtime seconds:                          0.03

 Model Class:                                      PILP

 Total variables:                      7
 Nonlinear variables:                  0
 Integer variables:                    7

 Total constraints:                    6
 Nonlinear constraints:                0

 Total nonzeros:                      14
 Nonlinear nonzeros:                   0

                       Variable           Value        Reduced Cost
                             X1        6.000000            1.000000
                             X2        4.000000            1.000000
                             X3        0.000000            1.000000
                             X4        0.000000            1.000000
                             X5        8.000000            1.000000
                             X6        4.000000            1.000000
                             X7        2.000000            1.000000
                             X8        6.000000            0.000000
                             X9        4.000000            0.000000

                            Row    Slack or Surplus      Dual Price
                              1        24.00000           -1.000000
                              2        0.000000            0.000000
                              3        0.000000            0.000000
                              4        0.9200000           0.000000
                              5        0.4600000           0.000000
                              6        0.000000            0.000000
                              7        0.6200000           0.000000
                              8        0.1500000           0.000000
```

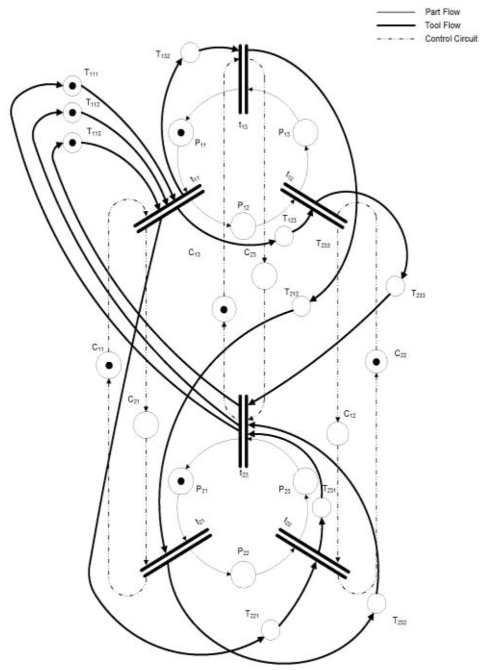


Fig.16 PN Graphical model for 2 parts 3 machines

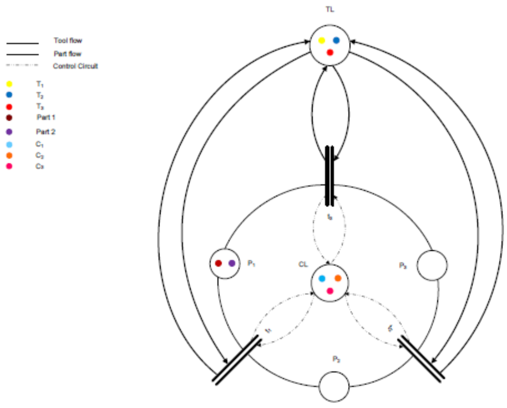


Fig.17 CPN Graphical model for 2 parts 3 machines

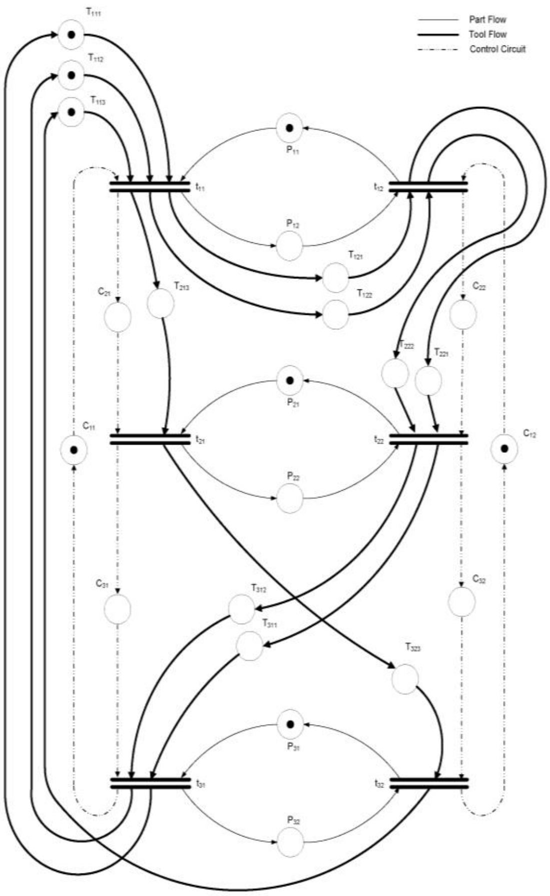


Fig.17 PN Graphical model for 3 parts 2 machines

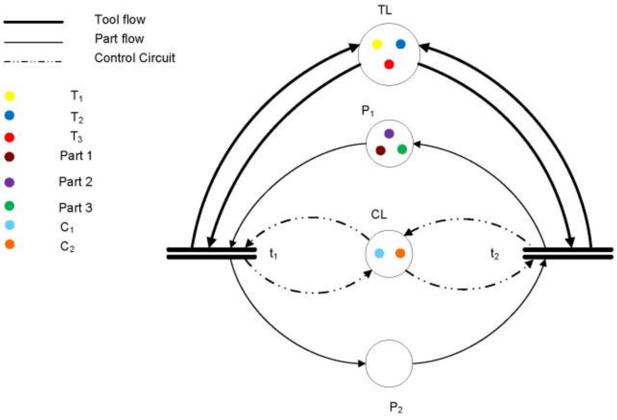


Fig.18 CPN Graphical model for 3 parts 2 machines

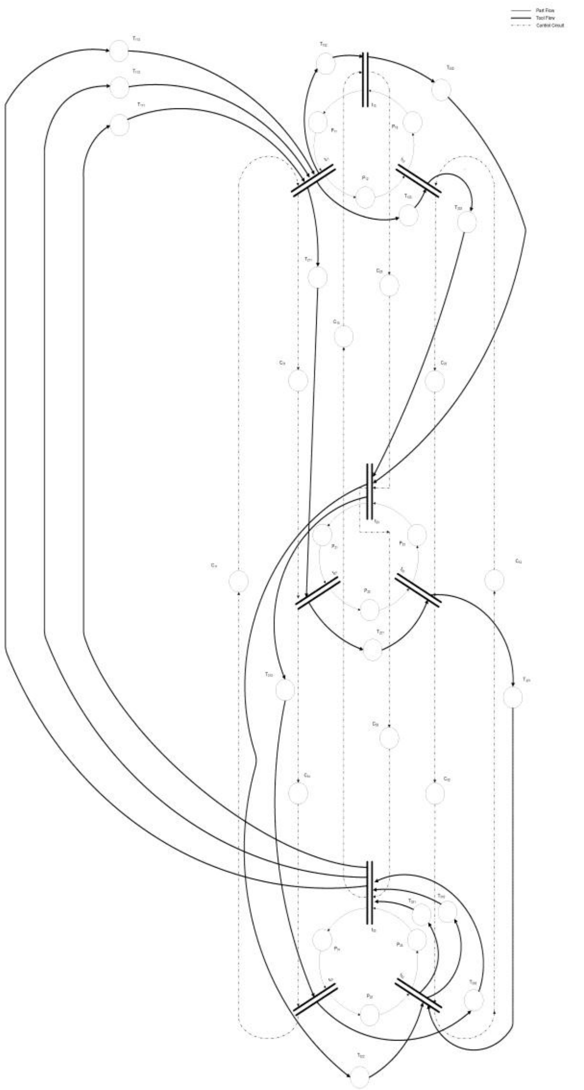


Fig.19 PN Graphical model for 3 parts 3 machines

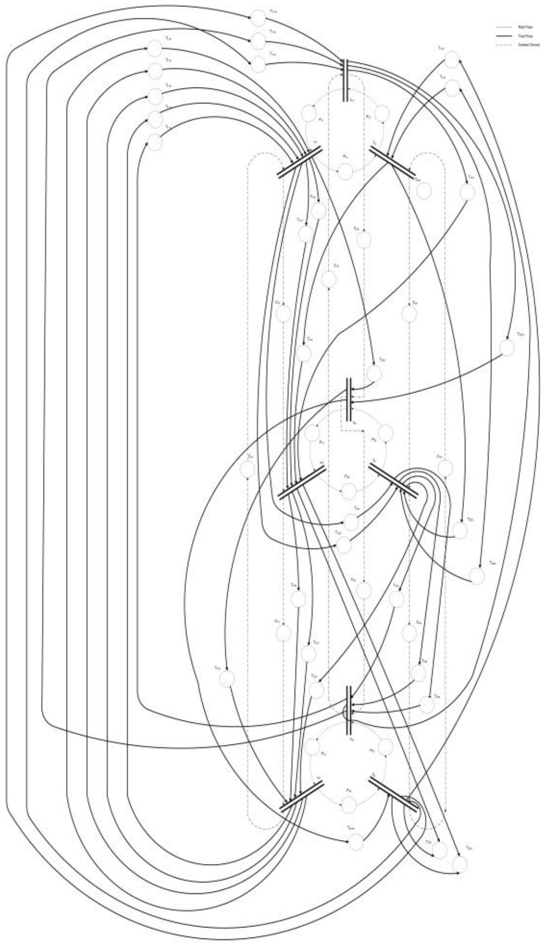


Figure PN Graphical model for 3 parts 3 machines & 10 tools

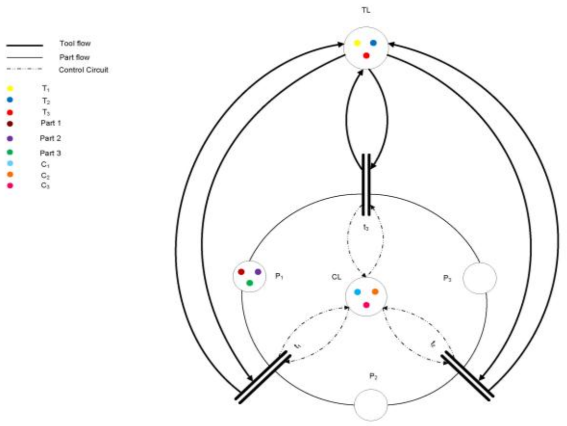


Fig.20 CPN Graphical model for 3 parts 3 machines

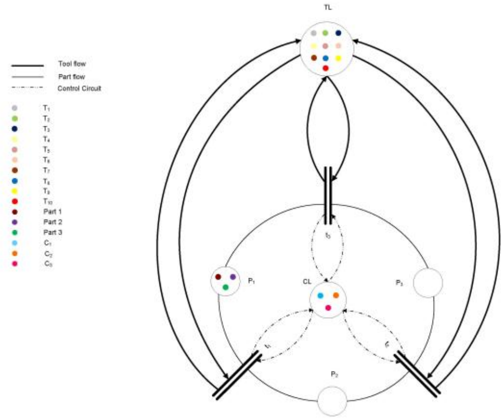


Fig.21 CPN Graphical model for 3 parts 3 machines & 10 tools

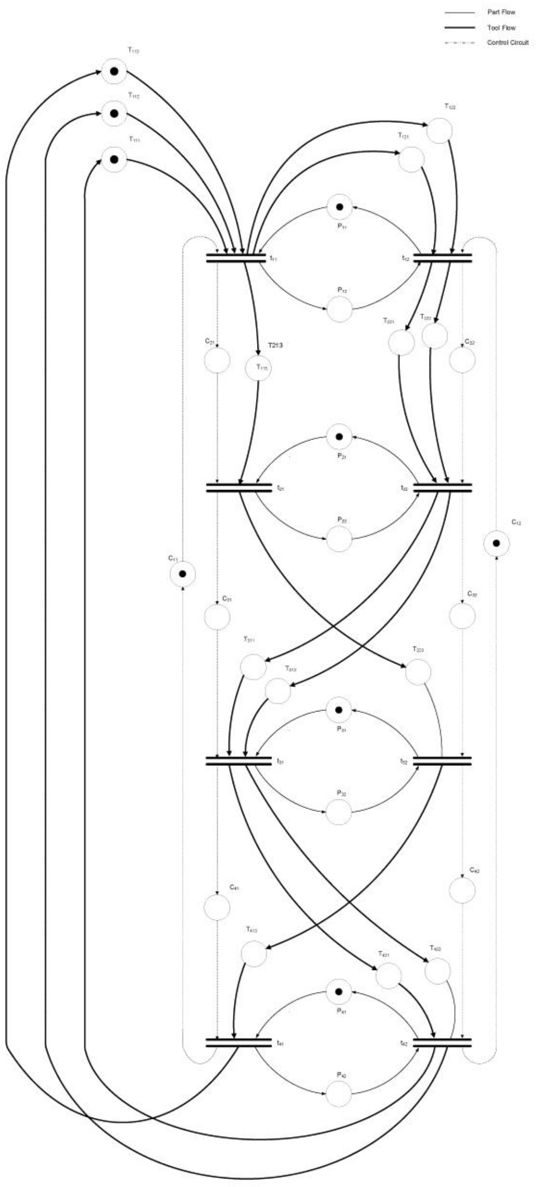


Fig.22 PN Graphical Model for 4 Parts 2 Machines and 3 Tools

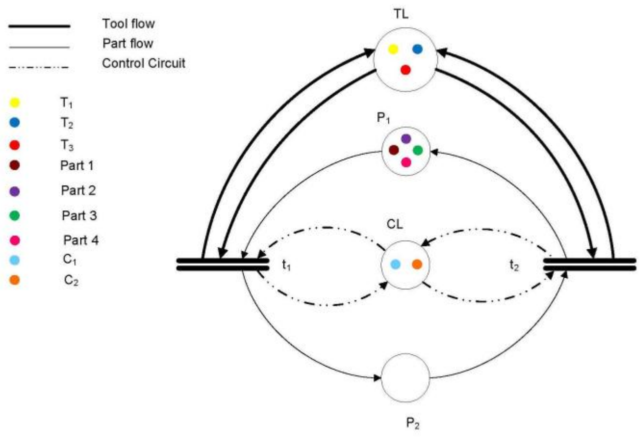


Fig.23 CPN Graphical Model for 4 Parts 2 Machines and 3 Tools

H. Conclusion

The parameters for the comparison are physical representation, Work in Process, Number of Tools and Number of Invariants. ***Here our main focus is on physical representation and analysis time of invariants***.

It seems very clear from the first four cases that the number of invariants reduces considerably if colored methodology is used. Still, the models being discussed are not too large in giving any hard time to INA in finding the invariants. It is clear that the models are equivalent and giving the same performance for the same sequence solved by a new approach using CPN and results are 100 % identical, which shows that our approach will give no error in any type of manufacturing system.

Criteria to evaluate the performance of black token and color petri nets includes graphical model, number of invariants, the complexity of lingo program and the parameters after lingo optimization. Some of the main advantages of CPN are as follow:

- Physical representation of CPN is simpler then black token PN as presented in the physical models of example manufacturing system
- Easy to understand because of simpler graphical representation
- No conflicts of events due to colored tokens
- Same graphical model of CPN can be used for different sequences
- Analysis time of CPN with INA is less than black token PN, because of the elementary circuits reduction i.e. reduced from 26 to 7.

This paper demonstrates a method to simplify the Petri net model of a manufacturing system in order to reduce the total number of invariants so as to decrease the analysis time of systems during investigations. In general, the addition of a tool, machine, or workpiece in a system exponentially increases the elementary circuits during analysis, thereby increasing the computational times. Organization of the proposed generic model permits the user to cluster several tools which considerably reduce the number of places within the model, thereby decreasing the graphical complexity. The CPN methodology if implemented not only clarifies the graphical illustration but also considerably decreases the resultant invariants. In response, time will be reduced within the integrated net analyzer in evaluating the system.

The proposed model is also investigated for different sequencing of the parts. It can be clearly seen in the given Figures that CPN models are more simple with no conflict of events wherein the case of Black token Petri Nets the Graphical model becomes more and more complex with the addition of a single tool or part. Moreover, for a change in sequence one has to develop another graphical model. But when the problem is solved by using color petri nets the change in sequence has no effect on graphical model as the colors of the tokens can be distinguished in the process.

Calculating number of invariants or elementary circuits is performed with the help of INA software. With increase in number of parts, tools or machines give rise to number of elementary circuits, the calculation time in case of CPN is much less as compared to black token petri nets.

Table II Number of Places and Invariants comparison in PN & CPN

S. No.		**Places**		**Invariants**	
		PN	**CPN**	**PN**	**CPN**
1	2P2M	14	9	26	7
2	2P3M	22	12	125	8
3	3P2M	23	11	286	8
4	3P3M	33	15	1643	9
5	3P3M10T	48	22	12465	16
6	4P2M	30	13	1279	9

I. Future Recommendation

The model will be further developed to be implemented on a real manufacturing system.

REFERENCES

[1] Yisheng An, Naiqi Wu , Xiangmo Zhao, Xuan Li and Pei Chen “Hierarchical Colored Petri Nets for Modeling and Analysis of Transit Signal Priority Control Systems” Appl. Sci. 2018, 8, 141; doi:10.3390/app8010141

[2] Maurizio Bevilacquaa , Filippo Emanuele Ciarapicaa , Mazzuto Giovannia “Timed coloured petri nets for modelling and managing processes and projects” 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '17

[3] Tsukada TK, Shin KG (1998) Distributed tool sharing in flexible manufacturing systems. IEEE Trans Robot Autom 14(3):379‑389

[4] Xu Z, Randhawa S (1998) Evaluation of scheduling strategies for a dynamic job shop in a tool-shared, flexible manufacturing environment. Prod Plan Control 9(1):74‑86

[5] Roh HK, Kim YD (1997) Due-date based loading and scheduling methods for a flexible manufacturing system with an automatic tool transporter. Int J Prod Res 35(11):2989‑3004

[6] Kashyap AS, Khator SK (1996) Analysis of tool sharing in an FMS: a simulation study. Comput Ind Eng 30(1):137‑145

[7] Tetzlaff UAW (1996) A queueing network model for flexible manufacturing systems with tool management. IIE Trans 28(4): 309‑317

[8] Amoako-Gyampah K, Meredith JR (1996) A simulation study of FMS tool allocation procedures. J Manuf Syst 15(6):419‑431

[9] Atan TS, Pandit R (1996) Auxiliary tool allocation in flexible manufacturing systems. Eur J Oper Res 89(3):642‑659

[10] Li XN, Cheung EHM, Chuah KB (1996) Increase the efficiency of an FMS by improving the tool scheduling strategies. J Mater Process Technol 61(1‑2):213‑218

[11] Rau KR, Chetty OVK (1996) Production planning of FMS under tool magazine constraints: a dynamic programming approach. Int J Adv Manuf Technol 11(5):366‑371

[12] Macchiaroli R, Riemma S (1996) Clustering algorithms to optimize the tool handling system management in an FMS. Int J Flex Manuf Syst 8(2):183‑201

[13] Grieco Aet al (1995) Simulation of tool and part flowinFMSs. Int J Prod Res 33(3):643‑658

[14] Barkaoui K, Ben-Abdallah I (1993) Modeling and performance evaluation of tool sharing management in FMS using stochastic Petri nets. In: Systems, man and cybernetics, 1993. ‘Systems engineering in the service of humans’, conference proceedings, international conference on 1993

[15] 11. Culea G, Cristea I, Puiu PG (2015) Modeling and analysis of processing flow in a flexible manufacturing system using synchronized Petri nets. Appl Mech Mater 809–810:1474–1479

[16] Sanjib Kumar Saren, Florin Blaga, and Tiberiu Vesselenyi “State space properties of Flexible Manufacturing Cell based on Hierarchical Timed Colored Petri Nets” MATEC Web of Conferences 137, 01012 (2017), DOI: 10.1051/matecconf/201713701012

[17] F. Yang, “Optimal O ne-Wafer Cyclic Scheduling of Time-Constrained Hybrid Multicluster Tools via Petri Nets,” IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(11), pp. 2920 – 2932, Nov. 2017. Unpublished

[18] Mohamed Torky and Aboul Ella Hassanein “Orbital Petri Nets: A Novel Petri Net Approach” arXiv:1806.03267v1 [cs.AI] 8 Jun 2018

[19] Liu HC, You JX, Li Z, Tian G. Fuzzy Petri nets for knowledge representation and reasoning: A literature review. Engineering Applications of Artificial Intelligence. 2017 Apr 1;60:45-56

[20] Q. H. Zhu., “Optimal Scheduling of Complex Multi-cluster Tools based on Timed Resource-oriented Petri Nets,” IEEE Acces s, vol. 4, pp. 2096 – 2109, May 2016. *Unpublished*

[21] Baruwa OT, Piera MA, Guasch A. Deadlock-free scheduling method for flexible manufacturing systems based on timed colored Petri nets and anytime heuristic search. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2015 May;45(5):831-46

[22] Shaukat Ali Shah, Erik L.J. Bohez, Roongrat Pisuchpen, "New modeling and performance evaluation of tool sharing control in FMS using colored Petri nets", Assembly Automation, Vol. 31 Issue: 2, pp.137-152 (2011)

[23] Liu H, Xing K, Wu W, Zhou M, Zou H. Deadlock prevention for flexible manufacturing systems via controllable siphon basis of Petri nets. IEEE Transactions on Systems, Man, and Cybernetics: Systems. 2015 Mar;45(3):519-29

[24] Du Y, Tan W, Zhou M. Timed compatibility analysis of web service composition: A modular approach based on Petri nets. IEEE Transactions on Automation Science and Engineering. 2014 Apr;11(2):594-606

[25] Hamid Ullah, (2011) "Petri net versus queuing theory for evaluation of FMS", Assembly Automation, Vol. 31 Issue: 1, pp.29-37.

[26] Tashnizi, E.S., Farahani, S.N. and Nahrekhalaji, A.R.F. (2008), “Production process optimization in flexible manufacturing system using petri nets”, Proceedings of the World Congress on Engineering and Computer Science, WCECS, San Francisco, CA, 22-24 October.

[27] Hamid, U. (2007), “*Feature based sequence planning and assembly system design using petri net*”, PhD dissertation, Asian Institute of Technology, Bangkok.

[28] Jain, A., Jain, P.K. and Singh, I.P. (2006), “Performance modeling of FMS with flexible process plans – a petri net approach”, International Journal of Simulation Modeling, Vol. 5 No. 3, pp. 101-13.

[29] Mandavilli, P. (November 05’ 2004). *Control synthesis in color condition.* Lexington, Kentucky: University of Kentucky

[30] Adrien leitold, Brigitta márczi, anna ibolya pózna , and miklós gerzson; Monitoring and diagnosis of manufacturing systems using timed colored petri nets; Hungarian journal of industry and chemistry veszprém vol. 41(1) pp. 17–26 (2013)

[31] Fermandez, R. and Zerhouni, N. (2001), “Modeling and analysis of disassembly systems using continuous Petri nets”, Proceedings of the 4th IEEE International Symposium on Assembly and Task Planning, pp. 232-7

[32] Jensen, K., Kristensen, L.M.: Coloured Petri Nets - Modelling and Validation of Concurrent Systems. Springer (2009)

[33] Freedman, Paul, “Time, Petri Nets, and Robotics,” IEEE Transactions on Robotics and Automation, Vol. 7, No. 4, August 1991, pp. 417-433.