

Schedules of Flexible Manufacturing Cells and Their Timed Colored Petri Net Models

W.M. Zuberek

Department of Computer Science
Memorial University of Newfoundland
St. John's, Canada A1C-5S7

A b s t r a c t

It is shown that flexible manufacturing cells can conveniently be represented and evaluated by colored Petri nets in which transitions represent operations performed by the machines of the cell and the transporting robot, tokens represent parts transported by the robot from one machine to another, and the colors are used to indicate different scheduling policies for the same cell.

1. INTRODUCTION

In colored Petri nets [3], information can be associated with individual tokens. These token attributes are called 'token colors'. Token colors can be quite complex, for example, they can describe the value of a variable or the contents of a message package. Token colors can be modified by (firing) transitions and also the conditions enabling transitions can be different for different colors.

Colored Petri nets can be used to integrate several models representing different variants of the same system; in this case the colors are used to indicate different versions of the model, and color-sensitive transitions are used for those activities which differ from one version to another.

Flexible manufacturing systems are discrete-event systems which are composed of a set of versatile machines, an automatic transportation system, and a decision-making system which determines *what* has to be done, *when* and *where*. Machines are often grouped into manufacturing cells (or robotic cells), in which a robot performs sequences of pickup, move, load, unload and drop operations, transporting the manufactured parts from one machine of the cell to another [7]. The behavior of flexible manufacturing cells is represented by 'events' and 'activities'; an activity corresponds to an operation performed by a machine or by the robot while an event corresponds to a change of activities. Different sets of activities determine the states of the system. In each state, several activities can occur concurrently, for example, several machines can perform their operations simultaneously and the robot can also transport a part. Petri nets [6, 5] provide a simple and convenient formalism for modeling systems of events which

can occur concurrently, but there are constraints on precedence and frequency of such occurrences. On a number of occasions Petri nets have been used for modeling and analysis of production systems [2, 8].

In order to study performance aspects of Petri net models, the duration of activities must also be taken into account and included into model specifications. Several types of Petri nets 'with time' have been proposed by assigning 'firing times' to the transitions or places of a net. In timed nets [10], transition firings are 'real-time' events, i.e., tokens are removed from input places at the beginning of the firing period, and they are deposited to the output places at the end of this period (sometimes this is also called a 'three-phase' firing mechanism). The firing times may be either deterministic or stochastic, i.e., described by some probability distribution function. In both cases the concepts of state and state transitions has been formally defined and used in derivation of different performance characteristics of the model [10].

Analysis of net models can be based on their behavior (i.e., the space of reachable states) or on the structure of the net; the first case is called reachability analysis and the second the structural approach. Invariant analysis is the most popular example of structural analysis [4]. Invariant analysis decomposes the net model into a number of simpler subnets and derives the properties of the model from properties of its components. In the case of net models of manufacturing cells, invariant analysis is used to determine the cycle time (or the throughput) of a cell from cycle times of its invariant subnets.

The throughput of a cell depends on the sequence of robot's activities as well as on the sequence in which different parts enter the cell [1]. The sequence of robot's activities is called the *schedule* of a manufacturing cell. Two types of schedules can be identified for analysis of the throughput of manufacturing cells. For simple schedules, exactly one part enters and one leaves the cell in each cycle. For composite schedules, several parts enter and leave the cell in each cycle. It appears that models of composite schedules can be obtained by combining several simple schedules [11]. Consequently, the approach used for analysis of simple schedules can also be used (with straightforward modifications) to analyze composite schedules.

In colored Petri net models of manufacturing cells,

colors are used to represent different schedules of the same cell, so, analysis of several schedules can be performed simultaneously.

The main contributions of this paper include a systematic development of a colored Petri net model of a complete family of schedules for a manufacturing cell composed of m machines, and analysis of the developed colored net model by net invariants. The result of this analysis determines the optimal schedule with respect to the throughput of the cell. Several possible extensions of the proposed approach are also discussed.

2. TIMED COLORED PETRI NETS

A convenient concept of ‘multisets’ is frequently used in the definitions which follow. A multiset (or a bag) X over a (nonempty) set A is any function $A \rightarrow \mathcal{N}$, $\mathcal{N} = \{0, 1, 2, \dots\}$. Intuitively, a multiset is a ‘set’ which can contain multiple occurrences of the same elements; if $X = A \rightarrow \mathcal{N}$ and $a \in A$, then $X(a)$ denotes the number of occurrences of a in X .

A colored Petri net is a 6-tuple $\mathbf{N} = (P, T, A, C, w, m_0)$ where:

P is a finite (nonempty) set of places,

T is a finite (nonempty) set of transitions,

A is a (nonempty) set of directed arcs which connect places with transitions and transitions with places, $A \subseteq P \times T \cup T \times P$, such that there are no isolated places or transitions; moreover, for each $t \in T$, $Inp(t)$ denotes the set of places which are connected by arcs directed to t , and $Out(t)$ the set of places which are connected by arcs directed from t ; $Inp(p)$ and $Out(p)$ are defined similarly,

C is a finite (nonempty) set of colors,

w is the arc function which associates, with each arc of the net, a function from the set of (occurrence) colors into a multiset of token colors, $w : A \rightarrow C \rightarrow C \rightarrow \mathcal{N}$, i.e., for each arc $a \in A$, $w(a) : C \rightarrow C \rightarrow \mathcal{N}$,

m_0 is the initial marking function which assigns multisets of token colors (or colored tokens) to places of a net, $m_0 : P \rightarrow C \rightarrow \mathcal{N}$.

This definition is a slightly modified version of a colored Petri net matrix [3]; the modification is made in order to simplify the definition, and to emphasize the relationship between colored nets and ‘ordinary’ nets (i.e., non-colored nets [6]). It should be observed that ordinary nets correspond to such colored nets in which: (i) the set of colors C contains just one color, and (iii) the arc function assigns the weights equal to 1 to all arcs of the net.

Let any function m that maps P into multisets of token colors, $m : P \rightarrow C \rightarrow \mathcal{N}$, be called a marking of the net \mathbf{N} .

An occurrence o (or occurrence color) of a transition $t \in T$, $o \in C$, is enabled at the marking m if and only if

$$\forall (p \in Inp(t)) \ w(p, t)(o) \leq m(p),$$

where $w(p, t)(o)$ denotes the application of the arc function w of the arc (p, t) to the occurrence color o , and \leq denotes element-wise comparison of multisets.

If an occurrence o of the transition t is enabled at the marking m_i , it can fire; firing o of t transforms m_i into another marking m_j which is directly (t, o) -reachable (i.e., reachable in ‘one step’) from m_i

$$\forall (p \in P) \ m_j(p) = m_i(p) - \sum_{t \in Out(p)} w(p, t)(o) + \sum_{t \in Inp(p)} w(t, p)(o)$$

where \sum is used for element-wise addition of multisets. During o of t ’s firing, tokens are removed from t ’s input places in numbers corresponding to the (input) arc functions applied to o , and tokens are added to t ’s output places in numbers corresponding to the (output) arc functions applied to o .

A colored net is *conflict-free* iff no two enabled occurrences of transitions share the same place color, i.e., iff

$$\forall (t_i, t_j \in T) \ \forall (o_\ell, o_k \in C) \ p \in Inp(t_i) \cap Inp(t_j) \Rightarrow \forall (c \in C) \ w(p, t_i)(o_\ell)(c) * w(p, t_j)(o_k)(c) = 0.$$

Only conflict-free colored nets are considered in this paper.

In timed colored nets, a ‘firing time’ is associated with each occurrence color of each transition. This firing time may be deterministic or it can be a random variable with some distribution function, for example, negative exponential distribution. Only deterministic firing times are considered in this paper.

In timed colored nets, the firing of an occurrence o of a transition t can be considered as a three-phase event; first, the (colored) tokens are removed from t ’s input places (in numbers corresponding to the input arc functions) and are transformed into occurrence colors of the firing transitions, the second phase is the firing time period when the firing (initiated in the phase one) continues, and in the last phase, occurrence colors are transformed into colored tokens of t ’s output places (in numbers corresponding to the output arc functions). If a transition occurrence becomes enabled while the transition is firing, a new independent firing cycle begins. All firings are initiated in the same instant of time in which the corresponding occurrences become enabled.

A conflict-free timed colored net is a pair, $\mathbf{T} = (\mathbf{N}, f)$ where

\mathbf{N} is a conflict-free colored net, $\mathbf{N} = (P, T, A, C, w, m_0)$,

f is a firing-time function which assigns a nonnegative firing time to each occurrence of each transition of the net, $f : T \rightarrow C \rightarrow \mathbf{R}^+$, where \mathbf{R}^+ denotes the set of nonnegative real numbers.

The behavior of timed colored nets can be described by a sequence of states and state transitions [9].

3. NET INVARIANTS

The arc functions are mappings $C \rightarrow C \rightarrow \mathcal{N}$, or $(C \times C) \rightarrow \mathcal{N}$; the second form can be represented quite conveniently by a rectangular array indexed by occurrence colors (columns) and token colors (rows). The incidence (or connectivity) matrix of a colored net is a $k \times \ell$ matrix A , where k is the number of places and ℓ the number of transitions, and:

$$A[k, \ell] = w(t_k, p_\ell) - w(p_k, t_\ell)$$

i.e., each element of A is a $C \times C \rightarrow \mathcal{N}$ mapping, and the difference is componentwise (for each occurrence color and each token color), so if the arc functions $w(p, t)$ and $w(t, p)$ are represented as $(C \times C)$ matrices of elements of \mathcal{N} , the difference is in the sense of the corresponding elements of these matrices.

A place invariant of a colored net is a k -element (column) vector of multisets $C \rightarrow \mathcal{N}$ such that

$$transp(A) * I = 0$$

where $transp(A)$ is the transpose of A , and the operation “*” is a componentwise application of elements of A to multisets of I (and “0” is the ℓ -element vector of zero multisets $C \rightarrow \{0\}$):

$$\forall (1 \leq j \leq \ell) \sum_{1 \leq i \leq k} A[i, j](I[i]) = 0$$

and the sum is performed componentwise (on multisets):

$$\forall (1 \leq j \leq \ell) \forall (c_t \in C) \forall (c_p \in C) \sum_{1 \leq i \leq k} A[i, j][c_p, c_t] * I[i](c_p) = 0$$

A colored net $\mathbf{N} = (P, T, A, C, w, m_0)$ is *decoupled* iff the occurrences of transitions do not ‘mix’ the colors, i.e., iff there is a partition $\mathcal{P}(C)$ of the set of colors C (and an implied equivalence relation H_{eq} on C), such that all occurrences of transitions have their (nonzero) input and output arc mappings in the same equivalence classes of \mathcal{P} :

$$\exists (H_{eq} \subset C \times C) \forall (t \in T) \forall (o \in C) \forall (p_i \in Inp(t)) \forall (p_j \in Out(t)) \forall (c_\ell, c_k \in C) w(p_i, t)(o)(c_\ell) > 0 \wedge w(t, p_j)(o)(c_k) > 0 \Rightarrow (c_\ell, c_k) \in H_{eq}.$$

It should be observed that if a net is decoupled, it can be analyzed independently for each equivalent class of colors because different classes of colors never interfere one with another.

4. SCHEDULES AND THEIR MODELS

A simple manufacturing cell composed of three machines and a robot is sketched in Fig.1; the machines are denoted by M_1, M_2 and M_3 , In represents a conveyor bringing the parts in while Out a conveyor for outgoing parts. Let, for simplicity, all parts follow the same path from In to M_1 , from M_1 to M_2 , from M_2 to M_3 , and finally from M_3 to Out .

The parts are moved between machines by the robot which follows a cyclical pattern of actions called its schedule. It is known that for an m -machine cell there are $m!$ different simple schedules [7]. The six

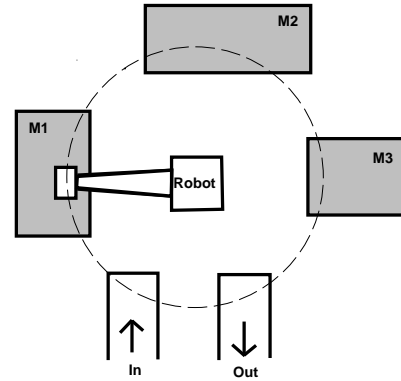


Fig.1. Layout of a three-machine cell.

simple schedules for a three-machine cell are as follows (the robot moves from X to Y are denoted by $X \Rightarrow Y$ if the robot carries a part, and by $X \rightarrow Y$ otherwise):

- A: $In \Rightarrow M_1 \Rightarrow M_2 \Rightarrow M_3 \Rightarrow Out \rightarrow In$
- B: $In \Rightarrow M_1 \Rightarrow M_2 \rightarrow M_3 \Rightarrow Out \rightarrow M_2 \Rightarrow M_3 \rightarrow In$
- C: $In \Rightarrow M_1 \rightarrow M_3 \Rightarrow Out \rightarrow M_1 \Rightarrow M_2 \Rightarrow M_3 \rightarrow In$
- D: $In \Rightarrow M_1 \rightarrow M_2 \Rightarrow M_3 \rightarrow M_1 \Rightarrow M_2 \rightarrow M_3 \Rightarrow Out \rightarrow In$
- E: $In \Rightarrow M_1 \rightarrow M_2 \Rightarrow M_3 \Rightarrow Out \rightarrow M_1 \Rightarrow M_2 \rightarrow In$
- F: $In \Rightarrow M_1 \rightarrow M_3 \Rightarrow Out \rightarrow M_2 \Rightarrow M_3 \rightarrow M_1 \Rightarrow M_2 \rightarrow In$

Examples of Petri net models of simple schedules are shown in Fig.2 and Fig.3 [11] (the schedules are shown in the instant of time when the robot is going to pick a part from the input conveyor). In timed models, transitions represent operations and places ‘conditions’ (in the most general sense); the three machines (or rather machine operations) are represented by t_1, t_2 and t_3 , each transition with its input and output place (for ‘part loaded’ and ‘machine operation finished’ conditions). The firing times associated with these transitions (the same for all occurrences) $f(t_1) = o_1, f(t_2) = o_2$ and $f(t_3) = o_3$ represent the (average) times of performing the operations on machines M_1, M_2 and M_3 , respectively. The remaining parts of the models represent the robot’s activities. It is assumed that there is always an available part in In and that Out removes manufactured parts sufficiently quickly, so In and Out are not shown.

A colored Petri net model of all six schedules is shown in Fig.4. There are six basic colors representing the schedules (and also denoted A, B, ..., F), and five auxiliary colors which are used for elimination of potential conflicts when the models of different schedules are combined together. These auxiliary colors are needed for schedules B, C, D, E and F, and are denoted by b, c, d, e i f; they are used as token colors only, so there are eleven token colors and only six occurrence colors (formally, the arc functions are partial functions which are undefined for the occurrence auxiliary colors).

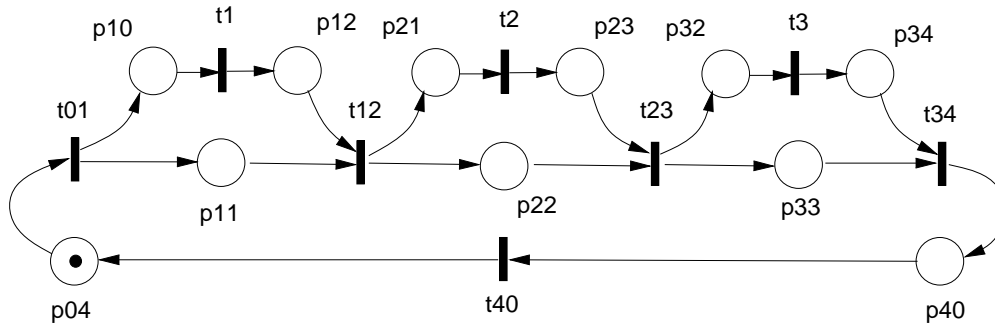


Fig.2. Petri net model of the simple schedule A.

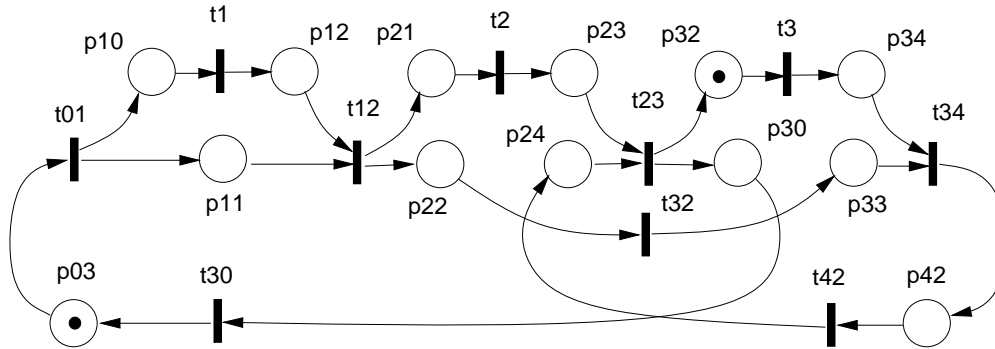


Fig.3. Petri net model of the simple schedule B.

The transitions correspond to the following actions:

	<i>robot's operation</i>	<i>exec time</i>
t_1	M_1 operation	o_1
t_2	M_2 operation	o_2
t_3	M_3 operation	o_3
t_{01}	pick from <i>In</i> , move to M_1 , load	$u + w + y$
t_{11}	move from M_1 to M_2 [D,E]	y
t_{12}	unload M_1 , move to M_2 , load	$v + w + y$
t_{13}	move from M_1 to M_3 [C,F]	$2y$
t_{20}	move from M_2 to <i>In</i> [E,F]	$2y$
t_{22}	move from M_2 to M_3 [B,D]	y
t_{23}	unload M_2 , move to M_3 , load	$v + w + y$
t_{30}	move from M_3 to <i>In</i> [B,C]	$2y$
t_{31}	move from M_3 to M_1 [D,F]	$2y$
t_{34}	unload M_3 , move to <i>Out</i> , drop	$v + x + y$
t_{40}	move from <i>Out</i> to <i>In</i> [A,D]	y
t_{41}	move from <i>Out</i> to M_1 [C,E]	$2y$
t_{42}	move from <i>Out</i> to M_2 [B,F]	$2y$

the occurrence colors – or the schedules – are indicated in square brackets for those transitions for which only some occurrence colors are used.

The *execution times* are the same for all transition occurrences, and are given assuming that ‘ u ’ denotes the (average) pickup time, ‘ v ’ the (average) unload time, ‘ w ’ the (average) load time, ‘ x ’ the (average) drop time and ‘ y ’ the average ‘travel’ time between adjacent machines; it is assumed, for simplicity, that the travel time the same for all adjacent machines, and also the same for M_3 to *Out*, *Out* to *In* and *In* to M_1 moves).

The arc functions w are mappings $C \rightarrow C \rightarrow \mathcal{N}$; for most of arcs, the arc functions are (partial) identity

functions for the basic colors A, B, ..., F, i.e., for an arc a , an occurrence color $o \in C$ and a token color $c \in C$:

$$w(a)(o)(c) = \begin{cases} 1, & \text{if } o, c \in \{A, B, C, D, E, F\} \wedge o = c, \\ 0, & \text{if } o, c \in \{A, B, C, D, E, F\} \wedge o \neq c, \\ \text{undefined,} & \text{otherwise.} \end{cases}$$

The definitions of all ‘non-standard’ arc functions are as follows (the occurrence colors correspond to columns and each entry is a function $g : C \rightarrow \mathcal{N}$ which is shown using a simplified notation “ $X : i$ ” denoting $g(c) = i$ if $c = X$ and 0 otherwise):

<i>arc</i>	A	B	C	D	E	F
(t_{01}, p_{11})	A:1	B:1	c:1	d:1	e:1	f:1
(t_{12}, p_{22})	A:1	b:1	C:1	d:1	e:1	f:1
(t_{23}, p_{33})	A:1	b:1	c:1	d:1	E:1	f:1
(t_{34}, p_{40})	A:1	b:1	c:1	D:1	e:1	f:1
(p_{11}, t_{11})	—	—	—	d:1	e:1	—
(p_{11}, t_{13})	—	—	c:1	—	—	f:1
(p_{22}, t_{20})	—	—	—	—	e:1	f:1
(p_{22}, t_{22})	—	b:1	—	d:1	—	—
(p_{33}, t_{30})	—	b:1	c:1	—	—	—
(p_{33}, t_{31})	—	—	—	d:1	—	f:1
(p_{40}, t_{41})	—	—	c:1	—	e:1	—
(p_{40}, t_{42})	—	b:1	—	—	—	f:1

It can be observed that, in addition to the representation of machines, there is a systematic structure of the net model shown in Fig.4:

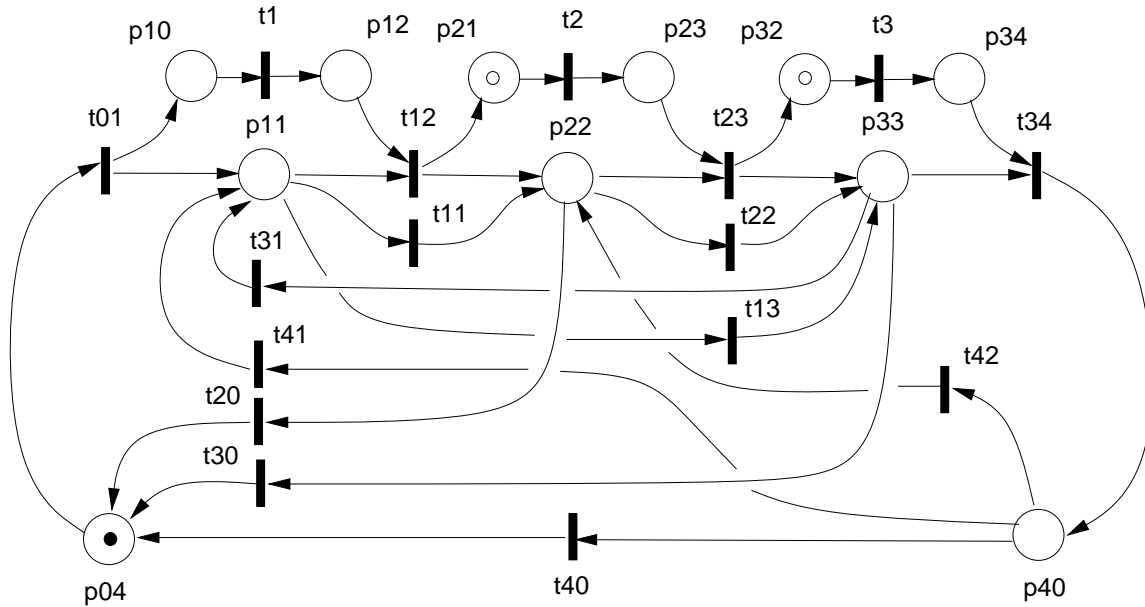


Fig.4. Colored net model of a three-machine cell.

- there is a place corresponding to each of the machines (p_{11} , p_{22} and p_{33} in Fig.4), and single places corresponding to *Input* and *Output* (p_{40} and p_{04} in Fig.4); in general, for an m -machine cell, there are $m+2$ such places;
- places representing *Input* and all machines (i.e., p_{04} and p_{11} , p_{22} , p_{33}) have three input transitions each (or m input transitions in general) representing moves from the ‘other’ machines and *Output* (and from *Input* in the case of M_1); the ‘other’ machines do not include the ‘next’ machine (with *Output* as ‘the next machine’ for M_3); so, for p_{04} the input arcs are from M_2 , M_3 and *Output*, for p_{11} the input arcs are from M_3 , *Output* and *Input*, for p_{22} the input arcs are from *Output*, M_1 (carrying a part) and M_1 (without a part), and for p_{33} the input arcs are from M_1 , M_2 (carrying a part) and M_2 (without a part);
- places representing *Output* and all machines have three output transitions each (or m output transitions in general) representing moves to ‘other’ machines and *Input* (and *Output* in the case of M_3); the ‘other’ machines do not include the ‘previous’ machine (with *Input* as ‘the previous machine’ for M_1); so for p_{11} the output arcs are to M_2 (carrying a part), M_2 (without a part) and M_3 , for p_{22} the output arcs are to M_3 (carrying a part), M_3 (without a part) and *Input*, for p_{33} the output arcs are to *Output*, *Input* and M_1 , and for p_{40} the output arcs are to *Input*, M_1 and M_2 ;
- the place representing *Input* (p_{04}) has only one output transition (t_{01} , representing ‘pick a part, move and load M_1 ’);
- the place representing *Output* (p_{40}) has only one input transition (t_{34} , representing ‘unload M_3 , move and drop),

- the total number of ‘scheduling’ transitions, resulting from the above rules, is equal to (counting either the input or output arcs) $1 + m * (m + 1)$, so for a three-machine cell there are 13 transitions modeling the possible robot schedules (see Fig.4); a net model of a four-machine cell needs 21 transitions (and 6 places) to represent all possible robot’s schedules, and a model of a five-machine cell, needs 31 such transitions (and 7 places).

The colored net shown in Fig.4 is decoupled and the partition of the set of colors is as follows:

$$\mathcal{P}(C) = \{\{A\}, \{B, b\}, \{C, c\}, \{D, d\}, \{E, e\}, \{F, f\}\}$$

Consequently, the invariants are grouped in sections corresponding to different colors (i.e., different schedules). There are 8 invariants for color/schedule A, 5 invariants for color/schedule B, 6 invariants for color/schedule C, etc.; total number of place-invariants for this model is 33. The minimum cycle time of each schedule is determined by the invariant subnet with the maximum total cycle time. Since all invariant subnets are simple cyclic nets, each subnet cycle time is equal to the sum of firing times assigned to all transitions of the (invariant) subnet. The cycle times of the six schedules are as follows (since the sets of transitions of some invariant subnets are subsets of those of other invariant subnets, not all invariants are used in the formulas)

schedule	cycle time
A	$\tau_A = \tau_1$
B	$\tau_B = \max(\tau_9, \tau_{10}, \tau_{12})$
C	$\tau_C = \max(\tau_{14}, \tau_{16}, \tau_{17}, \tau_{19})$
D	$\tau_D = \max(\tau_{20}, \tau_{21}, \tau_{22}, \tau_{23}, \tau_{24})$
E	$\tau_E = \max(\tau_{25}, \tau_{26}, \tau_{28})$
F	$\tau_F = \max(\tau_{30}, \tau_{31}, \tau_{32}, \tau_{33})$

where

$$\begin{aligned}
\tau_1 &= o_1 + o_2 + o_3 + u + 3v + 3w + x + 5y \\
\tau_9 &= o_1 + o_2 + u + 2v + 3w + 5y \\
\tau_{10} &= o_1 + u + 3v + 3w + x + 9y \\
\tau_{12} &= o_3 + v + w + 5y \\
\tau_{14} &= o_1 + o_2 + u + 2v + 2w + 5y \\
\tau_{16} &= o_2 + o_3 + 3v + 2w + x + 5y \\
\tau_{17} &= o_2 + u + 3v + 3w + x + 7y \\
\tau_{19} &= u + 3v + 3w + x + 10y \\
\tau_{20} &= o_1 + o_2 + o_3 + u + 3v + 3w + x + 5y \\
\tau_{21} &= o_1 + u + 2v + 2w + x + 5y \\
\tau_{22} &= o_2 + 2v + 2w + 4y \\
\tau_{23} &= o_3 + u + 2v + 2w + x + 5y \\
\tau_{24} &= u + 2v + 2w + x + 8y \\
\tau_{25} &= o_1 + u + v + 2w + 4y \\
\tau_{26} &= o_2 + o_3 + 3v + 2w + x + 5y \\
\tau_{28} &= o_3 + u + 3v + 3w + x + 9y \\
\tau_{30} &= o_1 + u + v + 2w + 4y \\
\tau_{31} &= o_2 + 2v + 2w + 4y \\
\tau_{32} &= o_3 + 2v + w + x + 4y \\
\tau_{33} &= u + 3v + 3w + x + 12y
\end{aligned}$$

Because the optimal schedule is the schedule with the minimum cycle time, so:

$$\tau_{opt} = \min(\tau_A, \tau_B, \tau_C, \tau_D, \tau_E, \tau_F).$$

6. CONCLUDING REMARKS

It has been shown that timed Petri nets can conveniently model (at least a large class of) flexible manufacturing cells. A number of different schedules can be analyzed simultaneously within one model by using token attributes (i.e., colors). For simple schedules, the modeling nets are conflict-free nets, so the cycle time (or throughput) can easily be determined from cycle times (or throughputs) of the invariant subnets. In fact, a symbolic solution can be derived, with times of basic operations (like ‘unload’, ‘move’) as parameters. This symbolic solution can than be used for very efficient comparisons and selections of optimum schedules for any given sets of parameter values.

Composite schedules, i.e., schedules in which several parts enter (and leave) the cell in each cycle, can be obtained by combining several simple schedules [11]. Since all simple schedules are modeled by the same (colored) net (as shown in Fig.4), the same modeling net will also represent the composite schedules of a manufacturing cell. For example, to model a composite schedule AB obtained by combining schedules A and B (Fig.2 and 3), two additional basic colors Ab , aB and an auxiliary color ab should be introduced with the following occurrences of transitions (the occurrences of Ab correspond to part A of the schedule AB, while occurrences aB to part B of this schedule):

arc	...	Ab	aB
(t_{12}, p_{22})	...	aB:1	ab:1
(t_{23}, p_{33})	...	Ab:1	ab:1
(t_{34}, p_{40})	...	Ab:1	ab:1
(p_{22}, t_{22})	...	—	ab:1
(p_{33}, t_{30})	...	—	ab:1
(p_{40}, t_{42})	...	ab:1	—

so that the complete robot’s path is $(p_{04}, t_{01}, p_{11}, t_{12}, p_{22}, t_{22}, p_{33}, t_{34}, p_{40}, t_{42}, p_{22}, t_{23}, p_{33}, t_{30}, p_{04}, t_{01}, p_{11}, t_{12}, p_{22}, t_{23}, p_{33}, t_{34}, p_{40}, t_{40}, p_{04})$. Systematic generation of such composite schedules and their analysis need to be investigated in greater detail.

A number of simplifying assumptions were used in the discussion presented in previous sections, e.g., all parts are identical, robot ‘travel times’ are equal, etc. It should be noted, however, that all these assumptions were made to simplify the discussion and they can easily be removed by straightforward modifications and extensions of the presented approach.

Acknowledgement

The Natural Sciences and Engineering Research Council of Canada partially supported this research through Research Grant A8222.

References

- [1] Dixon, C., Hill, S.D.: “Work-cell cycle-time analysis in a flexible manufacturing system”; Proc. Pacific Conf. on Manufacturing, Sydney–Melbourne, Australia, vol.1, pp.182–189, 1990.
- [2] Hillion, H.P.: “Timed Petri nets and application to multi-stage production system”; in: Advances in Petri Nets 1989 (Lecture Notes in Computer Science 424); pp. 281–305, Springer Verlag 1989.
- [3] K. Jensen, “Coloured Petri nets”; in: “Advanced Course on Petri Nets 1986” (Lecture Notes in Computer Science 254), G. Rozenberg (ed.), pp.248–299, Springer Verlag 1987.
- [4] Krueckeberg, F., Jaxy, M.: “Mathematical methods for calculating invariants in Petri nets”; in: “Advances in Petri Nets 1987” (Lecture Notes in Computer Science 266), G. Rozenberg (ed.), pp.104–131, Springer Verlag 1987.
- [5] Murata, T.: “Petri nets: properties, analysis and applications”; Proceedings of IEEE, vol.77, no.4, pp.541–580, 1989.
- [6] Reisig, W.: “Petri nets - an introduction” (EATCS Monographs on Theoretical Computer Science 4); Springer Verlag 1985.
- [7] Sethi, S.P., Sriskandarajah, C., Sorger, G., Blazewicz, J., Kubiak, W.: “Sequencing of parts and robot moves in a robotic cell”; Int. Journal of Flexible Manufacturing Systems, vol.4, pp.331–358, 1992.
- [8] Silva, M., Valette, R.: “Petri nets and flexible manufacturing”; in: “Advances in Petri nets 1989” (Lecture Notes in Computer Science 424), pp. 374–417, Springer Verlag 1989.
- [9] Zuberek, W.M., “Performance evaluation using timed colored Petri nets”, Proc. 33-rd Midwest Symp. on Circuit and Systems (Special Session on Petri Net Models), Calgary, Alberta, pp.779–782, 1990.
- [10] Zuberek, W.M.: “Timed Petri nets – definitions, properties and applications”; Microelectronics and Reliability (Special Issue on Petri Nets and Related Graph Models), vol.31, no.4, pp.627–644, 1991.
- [11] Zuberek, W.M., Kubiak, W., “Timed Petri net models of flexible manufacturing cells”; Proc. 36-th Midwest Symp. on Circuits and Systems, Detroit, MI, pp.922–925, 1993.