PAPER

# Scheduling for a Large-Scale Production System Based on a Continuous and Timed Petri-Net Model

**YoungWoo KIM**[†], *Nonmember*, **Akio INABA**[††], **Tatsuya SUZUKI**[†††], *Regular Members*, *and* **Shigeru OKUMA**[†††], *Nonmember*

**SUMMARY** This paper presents a new hierarchical scheduling method for a large-scale manufacturing system based on the hybrid Petri-net model, which consists of CPN (Continuous Petri Net) and TPN (Timed Petri Net). The study focuses on an automobile production system, a typical large-scale manufacturing system. At a high level, CPN is used to represent continuous flow in the production process of an entire system, and LP (Linear Programming) is applied to find the optimal flow. At a low level, TPN is used to represent the manufacturing environment of each sub-production line in a decentralized manner, and the MCT algorithm is applied to find feasible semi-optimal process sequences for each sub-production line. Our proposed scheduling method can schedule macroscopically the flow of an entire system while considering microscopically any physical constraints that arise on an actual shop floor.

***key words:*** *hybrid control, large-scale production system scheduling, hybrid Petri-net model, RTA* algorithm, LP*

## 1. Introduction

In a manufacturing system, scheduling is a typical combinatorial optimization problem, and one that decides the starting times and allocations of jobs to be processed. A desirable scheduling method must include two characteristics: (1) Easy formulation of a problem. (2) Quick identification of semi-optimal solutions. To meet these requirements, numerous approaches have been proposed. Model-based approaches have clear advantages over other ones, since they make it easy (1) to consider the various practical constraints that arise in production environments, (2) to monitor the current situation of a production system, and (3) to utilize powerful search algorithms. Among several modeling tools, the Petri Net (PN) is considered powerful because it can be understood graphically and features algebraic manipulability.

The PN model has served as the basis for several types of manufacturing systems: Schemes for deadlock avoidance were reported in [1]–[4] ; performance evaluation of given schedules was carried out by using a stochastic Petri net in [5]; and a combination with a heuristic search was developed in [6]–[8].

These previous studies considered various realistic constraints and transformed them into precedence constraints and capacity constraints in the PN model. The manufacturing environment of a real shop floor, however, is much more complex. The previous studies did not address adaptability to unexpected changes in the manufacturing environment or in production requests. From these points of view, the authors have proposed the MCT (Minimization of total Completion Time) algorithm based on the timed Petri-net model and the reactive fast graph search algorithm [8]. MCT has demonstrated superior computational performance and adaptability for job shop scheduling in FMS and has satisfied various constraints arising from the physical characteristics of the production system and the logical specification of production processes.

A real production system usually consists of many subsystems, and the direct application of the MCT algorithm to such a system is still difficult. On the other hand, if the MCT algorithm is applied to each subsystem independently, the results may bring about a bottleneck in the entire system and may cause an unreasonably large amount of inventory to build up. Therefore, for efficient scheduling in large-scale production, the process flow of the entire system should be captured in cooperation with the decision-making for the process sequence at each local subsystem.

In recent years, there have been some attempts to grasp overall process behavior analytically by employing fluid approximation models such as Continuous Petri Net [9], Hybrid Petri Net [10]–[13], Batch Petri Net [14], [15], Hybrid Flow Net [16], [17], High Level Petri Net [18]–[20] and Stochastic Petri Net [21], [22]. The basic idea in these fluid approximation models is to extend the modeling ability of the Petri net by allowing the place and/or transition to handle continuous variables in addition to discrete logical variables. F. Baluzzi et al. formulated a general problem to find the continuous process flow in a production system by means of combining First-Order Hybrid Petri Net (FOHPN) with Linear Programming (LP) [13]. In that work, the optimal process speed at each machine was found "myopically" by applying LP. Although these approaches based on fluid approxima-

Manuscript received May 27, 2002.
Manuscript revised October 12, 2002.

[†]The author is with Graduate School of Electrical Engineering, Nagoya University, Nagoya-shi, 464–8603 Japan.
[††]The author is an engineer at the Gifu Prefectural Research Institute of Manufactural Information Technology, Kakamigahara-shi, 509–0108 Japan.
[†††]The authors are with the Department of Electrical Engineering, Nagoya University, Nagoya-shi, 464–0814 Japan.

tion models could provide systematic planning schemes for high-level decision-making, the way in which they could be made to cooperate with low-level scheduling, which generates concrete feasible process sequences, has yet to be fully discussed.

This paper presents a new scheduling method for a large-scale manufacturing system based on a hierarchical Petri net model consisting of the Continuous Petri Net (CPN) and the Timed Petri Net (TPN). An application to an automobile production system, a typical large-scale production system, is studied. At the high level, CPN is used to represent continuous flows, which correspond to process speeds in each of the subsystems and the transfer speeds between production lines in the entire production system; LP is applied to find the optimal flow. Also, a sampling period is introduced in the high-level model, and the flow is supposed to be constant during two successive sampling instants. LP is carried out at each instance. The LP solution in the high-level model is regarded as a reference for the process speed in each sub-production line, which is modeled on the low level. On the low level, TPN is used to represent the manufacturing environment of each sub-production line in a decentralized manner, and the MCT algorithm is applied to find a feasible semi-optimal process sequence for each sub-production line. Our proposed scheduling method macroscopically tries to find an optimal flow of the process for the entire system and to generate microscopically a processing sequence that accommodates the physical constraints occurring on a real shop floor. An advantage of the hierarchical scheduling method is its adaptability to unexpected changes in the production environment, such as processing failures, conveyance delays and so on. The usefulness of the hierarchical scheduling method is demonstrated for a large-scale manufacturing system through some numerical experiments.

## 2. Hierarchical Modeling Based on CPN and TPN

This section outlines the production environment of the automobile manufacturing system that is the focus of this paper, and it introduces the hierarchical modeling method based on CPN and TPN. The CPN model adopts the same basic dynamics as the continuous part of the FOHPN model [13].

### 2.1 Outline of Automobile Production System

An automobile production system is generally composed of the following six sub-production lines.

(1) Body assembly line (production line 1: P1): press process, body assembly, painting/coating and interior finishing.

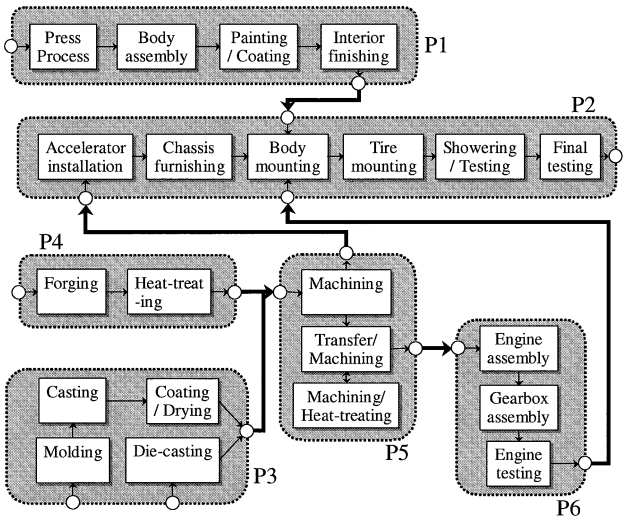



**Fig. 1** Automobile production system.

(2) Main assembly line (production line 2: P2): accelerator installation, chassis furnishing, body mounting, tire mounting, showering/testing and final testing.

(3) Metalworking line 1 (production line 3: P3): molding, casting, coating/drying and die-casting.

(4) Metalworking line 2 (production line 4: P4): forging and heat-treating.

(5) Machining line (production line 5: P5): machining, heat-treating and transfer/machining.

(6) Engine assembly line (production line 6: P6): engine assembly, gearbox assembly and engine testing.

The entire system includes the confluence and diffluence of all of the production lines, as shown in Fig. 1. Each line may include serial and/or parallel sequences.

### 2.2 High-Level Modeling by CPN

In high-level modeling, the macroscopic behavior of an automobile production system is described by CPN, which helps us to understand the process flow of an entire system.

#### 2.2.1 Structure of CPN

The CPN model of the automobile production system is defined as follows:

$$N_C = (P_C, T_C, v, I_C^-, I_C^+, M_C^0) \quad (1)$$

where components $P_C$, $T_C$, $v$, $I_C^+$, $I_C^+$ and $M_C^0$ are given in the following:

• Set of continuous places $P_C$
Continuous places represent load stations or unload stations in each production line. Subsets of $P_C$ are defined as follows:

$P_C^1$ : set of load stations
$P_C^2$ : set of unload stations

Then the set of continuous places $P_C$ is given by

$$P_C = P_C^1 \cup P_C^2. \tag{2}$$

• Set of continuous transitions $T_C$
Continuous transitions represent the flow of work from a macroscopic point of view. There are two kinds of transitions: (1) a transition corresponding to a production line (from load station(s) to the unload station in each production line) and (2) a transition corresponding to a transfer line (from the unload station to load station(s) in the next production line). Subsets of $T_C$ are defined as follows:

$$T_C^1 = \{\ t \mid {}^\bullet t \subset P_C^1, t^\bullet \subset P_C^2\}$$
$$T_C^2 = \{\ t \mid {}^\bullet t \subset P_C^2, t^\bullet \subset P_C^1\}$$

Here, ${}^\bullet t (t^\bullet)$ denotes the preset (postset) of a transition. Then, the set of continuous transitions $T_C$ is given by

$$T_C = T_C^1 \cup T_C^2. \tag{3}$$

• The firing speed of continuous transition $v$
Function $v(t_l, \tau)$ specifies the firing speed assigned to continuous transition $t_l$ at time $\tau$. $v(t_l, \tau)$ must satisfy

$$0 \leqq v(t_l, \tau) \leqq v_M(t_l) \tag{4}$$

where $v_M(t_l)$ indicates the maximum firing speed of the continuous transition $t_l$.

• Functions $I_C^-$ and $I_C^+$
Functions $I_C^-(p,t)$ and $I_C^+(p,t)$ specify the backward and forward incidence relationships between the continuous transition $t$ and the continuous place $p$ that precedes or follows the transition, respectively. $I_C^+$ and $I_C^-$ represent the number of works that flow along the corresponding arc during a unit time.

• Initial marking $M_C^0$
$M_C^0$ is the initial marking.

• Virtual net dynamics
The net dynamics of CPN is supposed to be represented by a first-order differential equation for each continuous place $p \in P_C$, as follows:

$$\frac{dm_c(p,\tau)}{d\tau} = \sum_{t_i \in I_e} C(p, t_i) \cdot v(t_i, \tau) \tag{5}$$

where $m_c(p,\tau)$ is the marking of continuous place $p$ at time $\tau$, $C(p,t)$ is the incidence relationship given by $C(p,t) = I_C^+(p,t) - I_C^-(p,t)$, and $I_e = \{t_1, \cdots, t_n\}$ is the set of enabled continuous transitions $t_i$. Also, $m_c(p,\tau)$ is supposed to have its maximum value $m_M(p)$.

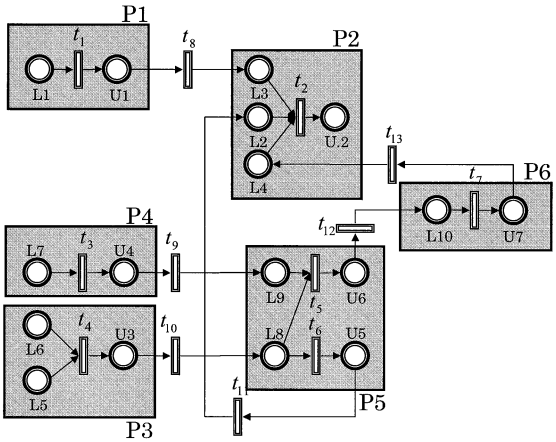



**Fig. 2** CPN model of automobile production system.

Equation (5) is easily transformed into its discrete version (6) supposing that $v(t_i, \kappa T_s)$ is constant during two successive sampling instants.

$$m_c(p, (\kappa+1)T_s) = m_c(p, \kappa T_s) + \sum_l C(p, t_l) \cdot v(t_l, \kappa T_s) \cdot T_s \tag{6}$$

where $\kappa$ is a number of cycles and $T_s$ is a sampling period. Note that this virtual net dynamics is used only for estimating the overall behavior of an entire system, and that the actual marking of a CPN model is decided by the evolution in the TPN model at each sampling instant (See Sect. 3).

Figure 2 shows the CPN model of the automobile production system in Fig. 1. In Fig. 2, $t_\alpha (\alpha = 1, \cdots, 7)$ and $t_\beta (\beta = 8, \cdots, 13)$ indicate the production line and the transfer line, respectively. Also, 'L$i$' and 'U$j$' indicate load station $i$ and unload station $j$, respectively. Note that each continuous transition $t_i$ can take only one output place $p \in t_i{}^\bullet$ in this model.

### 2.3 Low-Level Modeling by TPN

In the low-level modeling, the microscopic behavior in each production line is described by TPN. TPN enables us to take into consideration the various physical constraints that occur on a real shop floor and to introduce some powerful search engines to find (semi-) optimal process sequences.

#### 2.3.1 Structure of TPN

A TPN model $l (l = 1, \cdots, 7)$ is allocated to continuous transition $t_l$ and is defined as follows:

$$N_{D_l} = (P_{D_l}, T_{D_l}, C_{D_l}, \theta_{D_l}, I_{D_l}^-, I_{D_l}^+, M_{D_l}^0) \tag{7}$$

• Set of discrete places $P_{D_l}$
The discrete places represent the resources, and the

subsets of $P_{D_l}$ are defined as follows:

$P^1_{D_l}$ : set of load stations
$P^2_{D_l}$ : set of unload stations
$P^3_{D_l}$ : set of input buffers
$P^4_{D_l}$ : set of output buffers
$P^5_{D_l}$ : set of machines
$P^6_{D_l}$ : set of paths

The capacities of the discrete place for machines and paths are supposed to be 1, and the capacities of the other placescan be more than 1. The set of discrete places $P_{D_l}$ is given by

$$P_{D_l} = \bigcup_{i=1}^{6} P^i_{D_l}. \tag{8}$$

• Set of discrete transitions $T_{D_l}$
Discrete transitions represent the movement of the work (discrete token) from the microscopic point of view, that is, they represent the real mechanical action in the process. There are eight kinds of transitions, as follows:

$T^1_{D_l}=\{\ t \mid {}^\bullet t \subset P^1_{D_l},\ t^\bullet \subset P^3_{D_l}\}$
$T^2_{D_l}=\{\ t \mid {}^\bullet t \subset P^1_{D_l},\ t^\bullet \subset P^6_{D_l}\}$
$T^3_{D_l}=\{\ t \mid {}^\bullet t \subset P^1_{D_l} \cup P^5_{D_l}, t^\bullet \subset P^4_{D_l}\}$
$T^4_{D_l}=\{\ t \mid {}^\bullet t \subset P^3_{D_l},\ t^\bullet \subset P^5_{D_l}\}$
$T^5_{D_l}=\{\ t \mid {}^\bullet t \subset P^4_{D_l}, t^\bullet \subset P^6_{D_l}\}$
$T^6_{D_l}=\{\ t \mid {}^\bullet t \subset P^6_{D_l},\ t^\bullet \subset P^3_{D_l}\}$
$T^7_{D_l}=\{\ t \mid {}^\bullet t \subset P^6_{D_l},\ t^\bullet \subset P^6_{D_l}\}$
$T^8_{D_l}=\{\ t \mid {}^\bullet t \subset P^4_{D_l} \cup P^6_{D_l}, t^\bullet \subset P^2_{D_l}\}$

Then, the set of discrete transitions $T_{D_l}$ is given by

$$T_{D_l} = \bigcup_{i=1}^{8} T^i_{D_l}. \tag{9}$$

• Set of color information $C_{D_l}$ for token $c_{D_l}$
Each token has color information to represent the status of the corresponding work. The color of a token is defined as follows:

$$c_{D_l} = (c_t, c_o), \tag{10}$$
$$c_t \in C_{D_t}, c_o \in C_{D_o},$$

where
$C_{D_t}$ : set of indices for type of job
$C_{D_o}$ : set of indices for operation stage

• Functions $I^-_{D_l}$ and $I^+_{D_l}$
Functions $I^-_{D_l}$ and $I^+_{D_l}$ specify the backward and forward incidence relationships between the discrete transition and the discrete place that proceeds or follows the transition, respectively. They are defined as follows:

(1) If $t \in T^1_{D_l}, T^2_{D_l}, T^4_{D_l}, T^5_{D_l}, T^6_{D_l}, T^7_{D_l}, T^8_{D_l}$,
${}^\bullet t = \{p_1\}, t^\bullet = \{p_2\}$, $card\{{}^\bullet t\} = 1$,
then $I^-_{D_l}(p_1, t) = \{(c_{t_1}, c_{o_1})\}$
$I^+_{D_l}(p_2, t) = \{(c_{t_1}, c_{o_1})\}$

(2) If $t \in T^3_{D_l}, {}^\bullet t = \{p_1, p_2, p_3\}, t^\bullet = \{p_4\}$,
$p_1, p_2 \in P^1_{D_l}, p_3 \in P^5_{D_l} p_4 \in P^4_{D_l}$,
then $I^-_{D_l}(p_i, t) = \{(c_{t_i}, c_{o_i})\}$ for all $i$ that satisfy $1 \leqq i \leqq 3$
$I^+_{D_l}(p_4, t) = \{(c_{t_i}, c_{o_i} + 1)\}$

(3) If $t \in T^3_{D_l}, {}^\bullet t = \{p_1\}, t^\bullet = \{p_2\}, p_1 \in P^5_{D_l}$,
$p_2 \in P^4_{D_l}$,
then $I^-_{D_l}(p_1, t) = \{(c_{t_1}, c_{o_1})\}$
$I^+_{D_l}(p_2, t) = \{(c_{t_1}, c_{o_1} + 1)\}$

(4) If $t \in T^5_{D_l}, {}^\bullet t = \{p_1, p_2, p_3\},\ t^\bullet = \{p_4\}$,
$p_1, p_2, p_3 \in P^4_{D_l} p_4 \in P^6_{D_l}$,
then $I^-_{D_l}(p_i, t) = \{(c_{t_i}, c_{o_i})\}$ for all $i$ that satisfy $1 \leqq i \leqq 3$
$I^+_{D_l}(p_4, t) = \{(c_{t_i}, c_{o_i})\}$

(5) If $t \in T^5_{D_l}, {}^\bullet t = \{p_1, p_2\}, t^\bullet = \{p_3\}, p_1, p_2 \in P^4_{D_l}$,
$p_3 \in P^6_{D_l}$,
then $I^-_{D_l}(p_i, t) = \{(c_{t_i}, c_{o_i})\}$ for all $i$ that satisfy $1 \leqq i \leqq 2$
$I^+_{D_l}(p_3, t) = \{(c_{t_i}, c_{o_i})\}$

(6) If $t \in T^8_{D_l}, {}^\bullet t = \{p_1, p_2\},\ t^\bullet = \{p_3\}$,
$p_1 \in P^4_{D_l}, p_2 \in P^6_{D_l}, p_3 \in P^2_{D_l}$,
then $I^-_{D_l}(p_i, t) = \{(c_{t_i}, c_{o_i})\}$ for all $i$ that satisfy $1 \leqq i \leqq 2$
$I^+_{D_l}(p_4, t) = \{(c_{t_i}, c_{o_i})\}$

• Initial marking $M^0_{D_l}$ and firing time $\theta_{D_l}$
$M^0_{D_l}$ is the initial marking and $\theta_{D_l}$ is the firing time.

The formulation about the backward and forward relationships between the discrete transition and discrete place can be applied to all transitions in TPN models. For more detail, see [23]. Figure 3 shows a TPN model of production line 1 for the jobs listed in Table 1. A double and single circle in Fig. 3 indicate a place at which capacity is more than 1 and only 1, respectively. Also, white and black transitions are timed transitions at which the firing time is more than 1 and only 1, respectively.

In Fig. 3, L$n$, U$n$, IB$n$, OB$n$, M$n$ and P$n$ indicate load station, unload station, input buffer, output buffer, machine and path, respectively.

The TPN model represents the physical constraints and the behaviors explicitly, and enables us to analyze the status of the production line visually. The same formulation is applied to other production lines, as shown in Fig. 3 to Fig. 8.

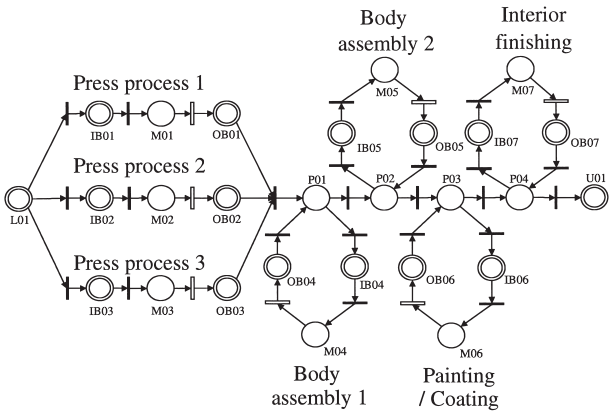



**Fig. 3** TPN model of production line 1.

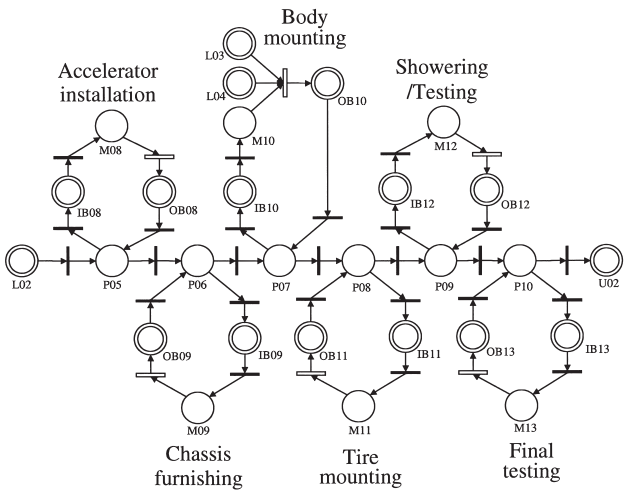



**Fig. 4** TPN model of production line 2.

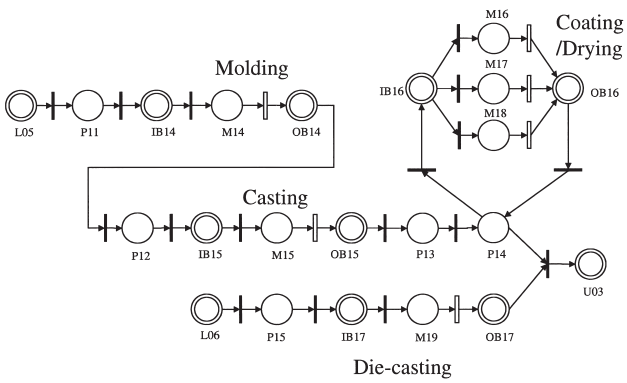



**Fig. 5** TPN model of production line 3.

## 2.4 Interaction between CPN and TPN

**Definition 1:** $m_{in}(t_l, \kappa T_s)$ is an *internal marking* for the transition $t_l$ in the CPN at sampling instant $\kappa T_s$. This value is equal to the number of discrete tokens in the TPN of the corresponding production line, excluding load station $P^1_{D_l}$ and unload station $P^2_{D_l}$. □

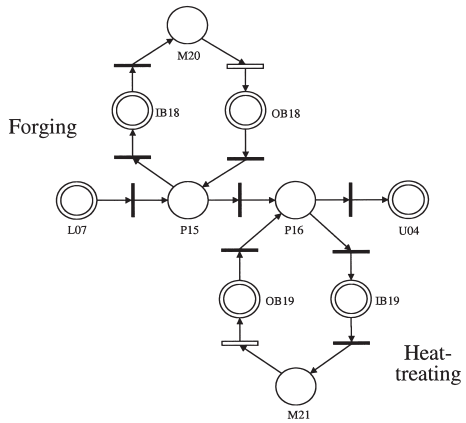



**Fig. 6** TPN model of production line 4.

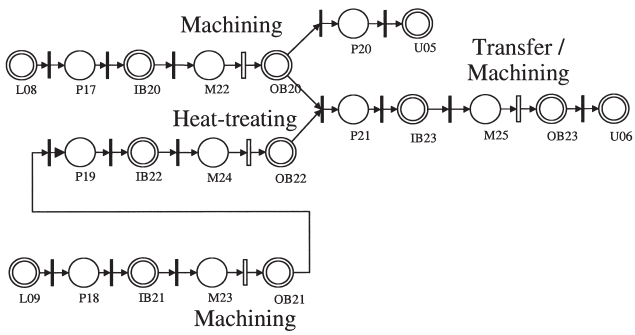



**Fig. 7** TPN model of production line 5.

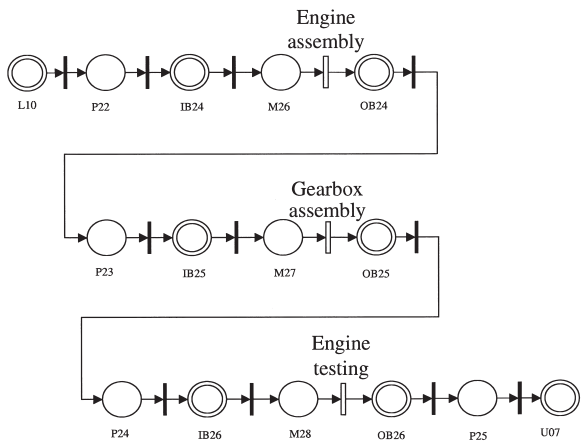



**Fig. 8** TPN model of production line 6.

By the introduction of the *internal marking*, the CPN model is consistent with the TPN model. Roughly speaking, the internal marking describes the amount of continuous token that is under the 'transit' state at the continuous transition $t_l$.

The CPN model and the TPN models interact with each other at every sampling instant $\kappa \cdot T_s$ by a "*marking converter*(MC)" and a "*reference generator*(RG)." The *marking converter* generates the marking of the CPN model $m_c(p, \kappa T_s)$ $(p \in P_C)$

**Table 1** Machine routings and processing times for production line 1.

| Operation Number | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Job 1 | M1 (4) | M4 (3) | M5 (2) | M6 (3) | M7 (3) |
| Job 2 | M2 (3) | | | | |
| Job 3 | M3 (2) | | | | |

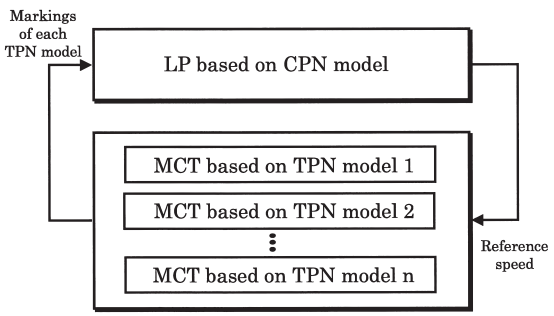



**Fig. 9** Outline of proposed scheduling method.

and internal marking $m_{in}(t_l, \kappa T_s)$ based on the markings of the TPN models, $m_d(p, \kappa T_s)$ $(p \in P_{D_l})$. The function of MC can be summarized as follows:

(1) The marking of $P^1_{D_l}$ and $P^2_{D_l}$ (i.e. load station and unload station) in each TPN model is directly referred to as the marking of the corresponding place in the CPN model.

(2) The total amount of marking of other places in each TPN model is referred to as the internal marking for the corresponding transition in the CPN model.

By the introduction of internal marking and MC, consistency between the CPN model and TPN model can be guaranteed. Also, a *reference generator* calculates the number of works $v(t_l, \kappa T_s) \cdot T_s \rfloor$ to be processed at the next cycle in each TPN model, cutting off decimals in order to make this number an integer. The interaction between the CPN model and the TPN models is depicted in Fig. 9.

## 3. Hierarchical Scheduling Based on Hierarchical Modeling

This section presents a hierarchical scheduling method based on the hierarchical Petri-net model developed in the previous section. At the high level, continuous variables such as the production speeds in each production line and transfer speeds in each transfer line are analytically planned at every sampling instant by applying LP to the CPN model based on $m_c(p, \kappa T_s)$ and $m_{in}(t, \kappa T_s)$ obtained by the marking converter. At the low level, the process sequence in each TPN model is scheduled by applying the MCT algorithm so as to process the number of works obtained by the reference generator. An outline of the proposed scheduling method is depicted in Fig. 10

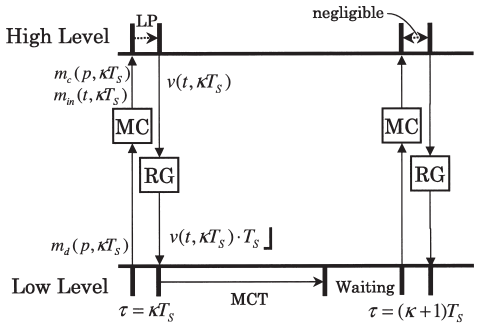



**Fig. 10** Interaction between high- and low-level scheduling.

### 3.1 High-Level Scheduling

LP enables us to maximize or minimize the linear objective function satisfying the linear inequality constraint and the non-negativity condition for variables. In our research, the simplex method, which is one of the most well known methods, is used.

Although the firing speed $v(t_l, \kappa T_s)$ for each transition has its maximum value $v_M(t_l)$, the actual feasible maximum speed depends on the distribution of continuous marking at $\kappa T_s$. In the following, another constraint on $v(t_l, \kappa T_s)$ is introduced, taking into account the situation around $t_l$.

**Definition 2:** $m_I(t_l, \kappa T_s)$ is an *input potential marking* for the input place $p_i \in {}^\bullet t_l$ of continuous transition $t_l$ at the sampling instant $\kappa T_s$. The $m_I(t_i, \kappa T_s)$ is defined as follows:

$$m_I(t_l, \kappa T_s) = \min_i \left\{ \frac{m_c(p_i, \kappa T_s) + I^+_C(p_i, t_k) \cdot v(t_k, \kappa T_s) \cdot T_s}{I^-_C(p_i, t_l)} \right\} + m_{in}(t_l, \kappa T_s) \quad (11)$$

where $t_k \in {}^\bullet p_i$. □

**Definition 3:** $m_O(t_l, \kappa T_s)$ is *output potential marking* for the output place $p_o \in t_l{}^\bullet$ of continuous transition $t_l$ at the sampling instant $\kappa T_s$. $m_O(t_l, \kappa T_s)$ is defined as follows:

$$m_O(t_l, \kappa T_s) = \frac{m_M(p_o) - m_c(p_o, \kappa T_s)}{I^+_C(p_o, t_l)} + \sum_k \frac{I^-_C(p_o, t_k) \cdot v(t_k, \kappa T_s) \cdot T_s}{I^+_C(p_o, t_l)} \quad (12)$$

where $t_k \in p_O^\bullet$, and $m_M(p_o)$ is the maximum capacity of place $p_o$. □

The $m_I(t_l, \kappa T_s)$ and $m_O(t_l, \kappa T_s)$ include not only

the amount of continuous token in $t_l$ at sampling instant $\kappa T_s$, but also the potential inflow to the input place $p_i$ and outflow from output place $p_o$, respectively. The virtual net dynamics in CPN (6) is used only to calculate $m_I$ and $m_O$, i.e. to estimate the potential flow of tokens.

**Definition 4:** In each transition $t_l$ ($t_l \in I_t(\kappa T_s)$), $v_I(t_l, \kappa T_s)$ and $v_O(t_l, \kappa T_s)$ correspond to the firing speeds taking into account the potential marking $m_I(t_l, \kappa T_s)$ and $m_O(t_l, \kappa T_s)$, respectively. They are defined as follows:

$$\begin{aligned} v_I(t_l, \kappa T_s) &= \frac{m_I(t_l, \kappa T_s)}{T_s}, \ \forall p_i \in {}^\bullet t_l \\ &= \min_i \left\{ \frac{m_c(p_i, \kappa T_s)}{I_C^-(p_i, t_l) \cdot T_s} + \frac{I_C^+(p_i, t_k) \cdot v(t_k, \kappa T_s)}{I_C^-(p_i, t_l)} \right\} \\ &\quad + \frac{m_{in}(t_l, \kappa T_s)}{T_s} \end{aligned} \tag{13}$$

and

$$\begin{aligned} v_O(t_l, \tau T_s) &= \frac{m_O(t_l, \kappa T_s)}{T_s}, \ p_o \in t_l{}^\bullet \\ &= \frac{m_M(p_o) - m_c(p_o, \kappa T_s)}{I_C^+(p_o, t_l) \cdot T_s} \\ &\quad + \sum_k \frac{I_C^-(p, t_k) \cdot v(t_k, \kappa T_s)}{I_C^+(p_o, t_l)}. \end{aligned} \tag{14}$$

□

From (4),(13) and (14), $v(t_l, \kappa T_s)$ must meet the following equation in order to find the 'reasonable' firing speed for $t_l$ at sampling instant $\kappa T_s$.

$$\begin{aligned} &v(t_l, \kappa T_s) \leqq v_M(t_l) \\ &v(t_l, \kappa T_s) \leqq v_I(t_l, \kappa T_s) \\ &v(t_l, \kappa T_s) \leqq v_O(t_l, \kappa T_s) \end{aligned} \tag{15}$$

**Example :** Consider the CPN described in Fig. 11. The marking is supposed to be given as follows:

$$\begin{aligned} &m_c(p_5, \kappa T_s) = 0, m_c(p_6, \kappa T_s) = 8, \\ &m_c(p_7, \kappa T_s) = 22, m_c(p_8, \kappa T_s) = 4, \\ &m_c(p_9, \kappa T_s) = 25 \end{aligned} \tag{16}$$

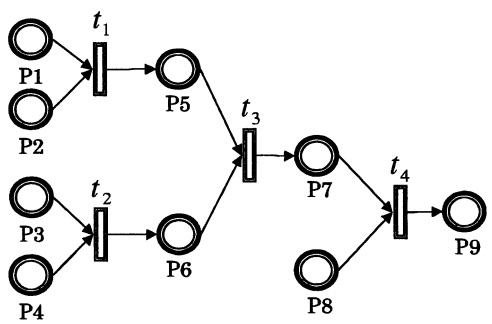



**Fig. 11** Example of CPN.

Here, maximum capacities $m_M$ for all places are supposed to be 30, maximum firing speeds $v_M$ for all transitions are 5, the sampling period $T_s = 4$, $I_C^+(p,t) = 1$, $I_C^-(p,t) = 1$ and $m_{in}(t_3, \kappa T_s) = 2$. The firing speed of each continuous transition is supposed to be given as follows:

$$\begin{aligned} &v(t_1, \kappa T_s) = 4, v(t_2, \kappa T_s) = 4, \\ &v(t_4, \kappa T_s) = 3. \end{aligned} \tag{17}$$

In this case, from ${}^\bullet t_3 = \{p_5, p_6\}$, ${}^\bullet p_5 = t_1$ and ${}^\bullet p_6 = t_2$, $M_I(t_3, \kappa T_s)$, and $m_O(t_3, \kappa T_s)$, $v_I(t_3, \kappa T_s)$ and $v_o(t_3, \kappa T_s)$ are calculated as follows.

$$\begin{aligned} &m_I(t_3, \kappa T_s) = \min \Big\{ (0 + 1 \times 4 \times 4), \\ &\qquad\qquad (8 + 1 \times 4 \times 4) \Big\} + 2 = 18 \\ &m_O(t_3, \kappa T_s) = 30 - 22 + 1 \times 3 \times 4 = 20, \\ &v_I(t_3, \kappa T_s) = 18/5 = 3.5, \\ &v_O(t_3, \kappa T_s) = 20/5 = 4. \end{aligned} \tag{18}$$

□

The inequality constraints of (15) for continuous transition $t_l$ can be applied to all transitions and transformed into equality constraints like (20) by introducing slack variables considering a condition of non-negativity for the objective function of (19). In (19), $c_l$ is the cost coefficient for transition $t_l$.

$$\max \sum_{t_l \in T_C} c_l \cdot v(t_l, \kappa T_s) \tag{19}$$

$$\begin{cases} v(t_1, \kappa T_s) & = v_I(t_1, \kappa T_s) + s_{1,1}(\kappa T_s) \\ \quad \vdots & \quad \vdots \\ v(t_L, \kappa T_s) & = v_I(t_L, \kappa T_s) + s_{1,L}(\kappa T_s) \\ v(t_1, \kappa T_s) & = v_O(t_1, \kappa T_s) + s_{2,1}(\kappa T_s) \\ \quad \vdots & \quad \vdots \\ v(t_L, \kappa T_s) & = v_O(t_L, \kappa T_s) + s_{2,L}(\kappa T_s) \\ v(t_1, \kappa T_s) & = v_M(t_1) + s_{3,1}(\kappa \Gamma) \\ \quad \vdots & \quad \vdots \\ v(t_L, \kappa T_s) & = v_M(t_L) + s_{3,L}(\kappa T_s) \end{cases} \tag{20}$$

This optimization problem belongs to a class of Linear Programming Problems (LPP), and some powerful algorithms, such as the simplex method, have been developed.

The optimal firing velocity, $v_{opt}(t_l, \kappa T_s)$, acts as the reference value of the production speed (or the transferring speed) for a production line (or transfer line) $l$, that is, the production line (or transfer line) tries to process (or convey) $v_{opt}(t_l, \kappa T_s) \cdot T_s$ works between $\kappa T_s$ and $(\kappa+1)T_s$. $v_{opt}(t_l, \kappa T_s)$ remains constant during a sampling interval.

In the proposed planning scheme, the macroscopic optimal flow of works is found at every sampling instant. This also enables the production system to flexibly accommodate changes in the system environment

such as process failures, conveyance delays, changes in process requirements and so on. For example, in case a process failure occurs in production line $l$ in the middle of the $\kappa$th sampling cycle, and the line recovers in the middle of the $(\kappa+n)$th sampling cycle, LP is applied by specifying $v_M(t_l) = 0$ between the $(\kappa + 1)$th sampling and the $(\kappa + n)$th sampling.

### 3.2 Decision of Discrete Process Sequence by MCT

The MCT [8] algorithm was developed for the time-optimal scheduling of a production system. This algorithm interleaves the search and execution in a reactive manner. In our framework, the MCT is applied to each production line (low-level TPN model) independently. The search engine of the MCT consists of the RTA* algorithm and a rule-based supervisor. The RTA* algorithm was originally developed in the field of artificial intelligence and is a reactive graph search method. It is adaptable to abrupt changes in the search space of a given problem, whereas conventional graph searches have the following drawbacks: (1) the search procedure must be completed before the execution and (2) when an unexpected change of search space (caused by a problem such as machine trouble) occurs, the remaining process must be rescheduled. In the RTA* algorithm, however, the quality of the solution may be inferior to those of the conventional algorithms. In order to overcome this problem, a rule-based supervisor is incorporated with the RTA* into the MCT algorithm. The role of the supervisor is to reduce the search space of RTA* by pruning off obviously undesirable branches in a graph search. In our application, the following rules are adopted as the supervisory control law:

(1) In the execution of all tasks in each production or conveyance line, maximum parallel execution has a priority.

(2) A job loading at the load station or output buffer is controlled so as to avoid an overflow in the input buffer.

Rule (1) enables the MCT to find a semi-optimal solution. Also, rule (2) prevents the production line from getting stacked in deadlock. In the RTA* algorithm, a heuristic function, which estimates the cost from the current state to the goal, plays an important role. In our study, the following heuristic function is adopted.

$$h(m,l) = \sum_{i=1}^{n} \sum_{j=p_c(s_i)+1}^{p_e(s_i)} \sigma_p(J(s_i), j, m, l) \qquad (21)$$

where $n$ is a number of tokens at marking $m$, $s_i$ is a label assigned to each token at marking $m$, $l$ is an index representing the line number, $J(s_i)$ is an index representing the type of the job corresponding to $s_i$, $p_c(s_i)$ is an index representing the operation stage of the job corresponding to $s_i$, $p_e(s_i)$ is an index representing the final operation for job $J(s_i)$, and $\sigma_p(i, j, m, l)$ is a processing time for the operation stage $j$ of the job $i$.

Note that the MCT always finds the time-optimal solution and does not try to control the production speed of each line explicitly. Once the low-level scheduler for each line receives the reference production speed $v_o(t_l, \kappa T_s)$ from the high-level scheduler, it runs the MCT, finally terminating the MCT after the last transition (i.e. the transition preceding the unload station) fires $v_o(t_l, \kappa T_s)T_s$ times or the sampling interval $T_s$ elapses. If no works exist in line $l$, then MCT is not executed until some works are imported from another line.

The proposed hierarchical scheduling method is summarized as follows:

Step 1 *(In marking converter)* Determine the continuous markings $m_c(p, \kappa T_s)$ for the continuous places $p \in P_C$ and initial marking $m_{in}(t, \kappa T_s)$ for the continuous transition $t \in T_C$ by checking the markings of each TPN model

Step 2 *(At high level)* Find the optimal firing speed $v_{opt}(t, \kappa T_s)$ of each continuous transition $t \in T_C$ at $\kappa T_s$ by applying LP.

Step 3 *(In reference generator)* Cut off decimals of $v_{opt}(t, \kappa T_s) \cdot T_s$ found in step 2 and give the resulting number to the correspondin production line as a reference production speed between $\kappa T_s$ and $(\kappa + 1)T_s$

Step 4 *(At low level)* Apply the MCT algorithm to each TPN model in a distributed manner.

Step 5 Repeat the above procedure, steps 1 to 4, until all works are completed.

## 4. Numerical Experiments

In order to confirm the usefulness of the proposed method, numerical experiments were performed. The manufacturing environments are shown in Fig. 1, and the capacities of all load stations and unload stations

**Table 2** Maximum capacities of load/unload stations.

| Load station | L.1 | L.2 | L.3 | L.4 |
|---|---|---|---|---|
| $m_M$ | 300 | 20 | 20 | 20 |
| Load station | L.5 | L.6 | L.7 | L.8 |
| $m_M$ | 200 | 200 | 100 | 20 |
| Load station | L.9 | L.10 | | |
| $m_M$ | 20 | 20 | | |
| Unload station | U.1 | U.2 | U.3 | U.4 |
| $m_M$ | 20 | 100 | 20 | 20 |
| Unload station | U.5 | U.6 | U.7 | |
| $m_M$ | 20 | 20 | 20 | |

**Table 3** Machine routings and processing times for works to be processed at each production line.

| Line | Work | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | M1 (4) | M4 (3) | M5 (2) | M6 (3) | M7 (3) | |
| 1 | 2 | M2 (3) | | | | | |
| 1 | 3 | M3 (2) | | | | | |
| 2 | 1 | M8 (5) | M9 (4) | M10 (6) | M11 (5) | M12 (3) | M13 (3) |
| 3 | 1 | M14 (2) | M15 (4) | M16 M17 M18 (5) | | | |
| 3 | 2 | M19 (3) | | | | | |
| 4 | 1 | M20 (5) | M21 (5) | | | | |
| 5 | 1 | M22 (2) | | | | | |
| 5 | 2 | M23 (3) | M24 (2) | M25 (1) | | | |
| 6 | 1 | M26 (6) | M27 (5) | M28 (4) | | | |

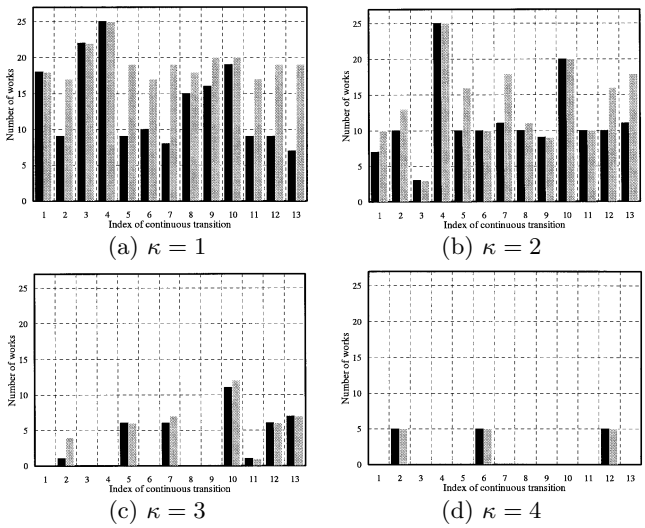
are listed in Table 2, while the capacities of all input buffers and output buffers in the low-level model are supposed to be 5. The machine routings and processing times for the works are listed in Table 3. The maximum firing speed and conveyance speed of each line are also listed in Table 4. These values can be easily found by applying the MCT algorithm several times, varying the number of works committed to each line.

### 4.1 Case Study

To begin with, the results of a sample scheduling are shown in Fig. 12 in a case in which 25 cars are produced. In Fig. 12, the reference number and the actual number of works processed in each continuous transition (line) at $\kappa$th cycle are shown. The gray bars represent the reference number of products to be produced in line $l$, $v_{opt}(t_l, \kappa T_s) \cdot T_s$, which was decided by LP in the high-level scheduler. Meanwhile, the black bars represent the number of works that actually flow through the corresponding continuous transition during the sampling interval. This value coincides with the number of firings of the transition that follows the load station in each TPN model. In the first cycle ($\kappa = 1$), the reference value almost coincides with $v_M$. This is because all continuous places except for L.1, L.5, L.6 and L.7 are initially empty, and $v_I(t_l, 1T_s)$ and $v_O(t_l, 1T_s)$ in (13) and (14) take almost the same value as $v_M$. As a result, the high-level scheduler (LP) tries to process as many works as possible. This yields a discrepancy between the reference number and the actual number

**Table 4** $v_M \cdot T_s$ of continuous transition.

| Index of continuous transition | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ |
|---|---|---|---|---|---|---|---|
| $v_M(t_i) \cdot T_s$ | 18 | 17 | 22 | 25 | 36 | 50 | 22 |
| Index of continuous transition | $t_8$ | $t_9$ | $t_{10}$ | $t_{11}$ | $t_{12}$ | $t_{13}$ | |
| $v_M(t_i) \cdot T_s$ | 20 | 20 | 20 | 20 | 20 | 20 | |




(a) $\kappa = 1$ (b) $\kappa = 2$ (c) $\kappa = 3$ (d) $\kappa = 4$

**Fig. 12** Scheduling example by the proposed method.

**Table 5** Evaluation of adaptability to machine failure.

| Failure condition | Final cycle |
|---|---|
| Without breakdown | 7 |
| With breakdown from 1st to 4th periods | 7 |
| With breakdown from 1st to 5th periods | 8 |
| With breakdown from 1st to 6th periods | 9 |
| With breakdown from 1st to 7th periods | 10 |

of works processed. In the second cycle ($\kappa = 2$), however, this discrepancy goes down. In this cycle, as for $t_3$, the actual number of works coincides with its reference, $v_{opt}(t_3, 2T_s)$. In this case study, all production processes have been completed at the fourth cycle.

### 4.2 Evaluation of Adaptability to Unexpected Changes in Production Environment

The adaptability to an unexpected change in the production environment is investigated, assuming that a machine failure occurs in the middle of production. Table 5 shows the result in the case where a machine (number 26) in production line 6 breaks down in the

**Table 6** Comparison with other strategies.

| Cycle | Processed number by proposed / A / B | Sum of sojourn time for all works by proposed / A / B |
|---|---|---|
| 1 | 3 / 2 / 3 | 6144 / 5904 / 6145 |
| 2 | 10 / 11 / 10 | 9056 / 14038 / 9057 |
| 3 | 10 / 11 / 10 | 9983 / 17574 / 9984 |
| 4 | 10 / 10 / 10 | 11101 / 17832 / 10981 |
| 5 | 10 / 11 / 10 | 10388 / 13184 / 9515 |
| 6 | 10 / 8 / 10 | 8883 / 9826 / 9248 |
| 7 | 7 / 12 / 10 | 5983 / 11111 / 9622 |
| 8 | 10 / 8 / 10 | 6086 / 6004 / 9119 |
| 9 | 10 / 12 / 10 | 4107 / 6393 / 9731 |
| 10 | 10 / 12 / 10 | 4172 / 3834 / 5222 |
| 11 | 10 / 3 / 7 | 1001 / 642 / 1405 |
| Total | 100 / 100 / 100 | 76904 / 106342 / 90029 |

first cycle, and recovers after several more cycles. In this experiment, 50 cars were supposed to be produced. In any case, no significant difference in the production time is found. This implies that the proposed method can flexibly accommodate unexpected changes in the manufacturing environment.

### 4.3 Comparison with Other Strategies

The usefulness of the proposed scheduling system is compared in the following two strategies:

Strategy A: Centralized MCT,

Strategy B: Constant reference speed and decentralized MCT.

In strategy A, the MCT algorithm is directly applied in a centralized manner to the TPN model, which integrates the models in Fig. 3 and Fig. 8. Although this strategy obviously has some difficulties, especially in a large-scale system, it does not require a flow model for the entire system. In strategy B, the constant reference production speed $v_M(t_i) \cdot T_s$ is adopted at every cycle, instead of using the varying optimal reference speeds obtained by LP in the proposed strategy. This strategy implies that the scheduler must try to process as many works as possible at every cycle regardless of the distribution of works in the entire system.

Table 6 shows the results for a case in which 100 cars are produced. In Table 6, 'Processed number' indicates the number of processed works at production line 2 (final line) at each cycle, and 'Sum of sojourn time of all works' indicates the summation of each work's sojourn time. In this numerical experiment, the maximum flow of transition $t_{13}$, $v_M(t_{13})T_s$ was reduced by 50% from the value listed in Table 4 so as to get almost the same production time by three strategies (an 11th cycle was required to complete production in any strategy). Machine failure in a subsystem can occur in operating a large-scale production system, and sometimes it will create a bottle-neck affecting the entire system. From this table, the proposed strategy achieves a shorter sojourn time than the other two strategies. This implies that, with the proposed strategy, the overall system will be less affected by a bottlenecked subsystem. This is one of the desired characteristics of large-scale production system scheduling. This clearly shows the advantage of introducing potential markings that can take into account the flow through the entire system.

## 5. Conclusions

This paper presented a new hierarchical scheduling method for a large-scale manufacturing system based on the hybrid Petri-net model, which consists of CPN and TPN. At a high level, the CPN has been used to represent continuous flow in the production process of an entire system, and LP has been applied to find the optimal flow. Meanwhile, at a low level, TPN has been used to represent the manufacturing environment of each sub-production line in a decentralized manner, and the MCT algorithm has been applied to find a feasible semi-optimal process sequence for each sub-production line. The key distinguishing features of our proposed method can be stated as follows. (1) The CPN model and the TPN model interact with each other at every sampling instant through the marking converter and the reference generator. (2) The macroscopic behavior of an entire system is scheduled by a high-level scheduler so that the total sojourn time is reduced. (3) The microscopic evolution of each sub-system is scheduled by a low-level scheduler taking into account the various physical constraints. By these characteristics, the applicability to the large-scale system and the adaptability to unexpected changes in production environments are increased. Finally, the usefulness of our method has been confirmed through numerical experiments. In future work, the behavior of the AGVs used for the transfer should be taken into account.

#### References

[1] F. Chu and X. Xie, "Deadlock analysis of Petri nets using Siphons and mathematical programming," IEEE Trans. Robot. Autom., vol.13, no.6, pp.793–804, 1997.

[2] N. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored petri net model," IEEE Trans. Syst., Man. Cybern. C, Appl. Rev., vol.29, no.2, pp.192–204, 1999.

[3] Z.A. Banaszak and B.H. Krogh, "Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows," IEEE Trans. Robot. Autom., vol.6, no.6, pp.724–734, 1990.

[4] N. Viswanadham, Y. Narahari, and T.L. Johnson, "Deadlock prevention and deadlock avoidance in flexible manufacturing systems using petrinet models," IEEE Trans. Robot. Autom., vol.6, no.6, pp.713–723, 1990.

[5] R.T. Al-Jaar and A.A. Desrochers, "Performance evaluation of automated manufacturing systems using generalized stochastic petri nets," IEEE Trans. Robot. Autom., vol.6,

no.6, pp.621–639, 1990.
[6] D.Y. Lee and F. DiCesare, "Scheduling flexible manufacturing systems using petri nets and heuristic search," IEEE Trans. Robot. Autom., vol.10, no.2, pp.123–132, 1994.
[7] D.Y. Lee and F. DiCesare, "Integrated scheduling of flexible manufacturing system employing automated guided vehicles," IEEE Trans. Ind. Electron., vol.41, no.6, pp.602–610, 1994.
[8] Y.W. Kim, A. Inaba, T. Suzuki, and S. Okuma, "FMS scheduling based on timed Petri net model and RTA* algorithm," Proc. IEEE ICRA, pp.848–853, Seoul, Korea, March 2001.
[9] R. David, "Modeling of hybrid systems using continuous and hybrid petri nets," Proc. Petri Nets and Performance Models PNPM '97, Saint Malo, France, June 1997.
[10] J. Le Bail, H. Alla, and R. David, "Hybrid Petri nets," Proc. 1st European Control Conference on Decision and Control, pp.108–113, 1997.
[11] H. Allam and H. Alla, "From Petri nets to hybrid automata," J.E.S.A., vol.32, 1998.
[12] S. Pettersson and B. Lennartson, "Hybrid modeling focused on hybrid Petri nets," Proc. 2nd European Workshop on Real-time and Hybrid Systems, pp.303–309, Grenoble, France, 1995.
[13] F. Balduzzi, A. Giua, and G. Menga, "First-order hybrid Petri nets: A model for optimization and control," IEEE Trans. Robot. Autom., vol.16, no.4, pp.382–399, 2000.
[14] I. Demongodin and F. Prunet, "Batch Petri nets," Proc. 7th Annual European Computer Conference, pp.29–37, Paris, May 1993.
[15] M. Caradec and F. Prunet, "Modeling of hybrid flexible production systems by coloured-batches Petri nets," J.E.S.A., vol.32, 1998.
[16] J.M. Flaus, "Hybrid flow nets for batch processes modeling and simulation," Proc. 2nd IMACS MATHOD Conference, pp.211–216, Vienna, 1997.
[17] J.M. Flaus, "Hybrid supervisor synthesis for a class of hybrid systems," Proc. [ZAY98], pp.185–192, 1998.
[18] A. Giua and E. Usai, "High-level hybrid Petri nets: A definition," Proc. 35th Conference on Decision and Control, Kobe, Japan, 1996.
[19] A. Giua and E. Usai, "Modeling hybrid systems by high-level Petri nets," J.E.S.A., vol.32, 1998.
[20] D. Vibert, C. Valentin-Toubinet, and E. Niel, "A modeling method to take into account fluctuations of continuous variables in a class of hybrid system," Proc. European Control Conference 97, Brussels, Belgium, 1997.
[21] R. Balduzzi, A. Giua, and G. Menga, "Hybrid stochastic Petri nets: Firing speed computation and FMS modeling," Proc. 4th Workshop on Discrete Event Systems (WODES '98), pp.432–438, Cagliari, Italy, Aug. 1998.
[22] K.S. Trivedi and V.G. Kulkarni, "FSPNs: Fluid stochastic Petri nets," Proc. 14th Int. Conf. on Applications and Theory of Petri Nets. ser. Lecture Notes in Computer Science, M. A. Marsan, ed., vol.691, pp.24–31, Springer-Verlag, Heidelberg, 1993.
[23] Y.W. Kim, A. Inaba, T. Suzuki, S. Okuma, T. Zamma, and F. Fuziwara, "Reactive FMS scheduling with RTA* and supervisor — An approach based on integrated Petri net model," Trans. Soc. Instrum. Control Eng., vol.37, no.5, 2001.

**YoungWoo Kim** was born in TaeGu, Korea, in 1973. He received the B.E. degree in Electrical Engineering from YeungNam University, Korea in 1997 and the M.S. degree from Nagoya University, Japan in 2000. Now, he is the PhD candidate of Electrical Engineering of Nagoya University, Japan. His research interests include Manufacturing Scheduling, Discrete Event System and Hybrid Control System.

**Akio Inaba** received the BE degree in Electronic Engineering from Gifu University, Japan in 1984. He received the Ph.D degree from Nagoya University, Japan in 1999. From 1984 to 1991, he was with Gifu Prefectural Industrial Technology Research Center. From 1991 to 1999, he was with Gifu Prefectural Metal Research Institute. Since 1999, he has been with Gifu Prefectural Research Institute of Manufacturing Information Technology. His current research interests are in discrete-event-system and the areas of robotics. He is a member of the Society of Instrument and Control Engineers of Japan, the institute of Electronics, Information and Communication Engineers, the Information Processing Society of Japan, and the Robotic Society of Japan.

**Tatsuya Suzuki** was born in Aichi, Japan, in 1964. He recieved the BE, ME and PhD degrees in Electronic Mechanical Engineering from Nagoya University, Japan in 1986, 1988 and 1991, respectively. From 1998 to 1999, he was a visiting researcher of the Mechanical Engineering Department of U.C.Berkeley. Currently, he is an associate Professor of the Department of Electrical Engineering of Nagoya University. He won the paper award from IEEJ in 1995. His current research interests are in the areas of motion control systems, discrete event systems, hybrid dynamical systems and intelligent control systems. Dr. Suzuki is a member of the IEEJ, SICE, ISCIE, RSJ, JSME and IEEE.

**Shigeru Okuma** was born in Gifu, Japan, in 1948. He received the B.E., M.E., and Ph.D. degrees in electronic-mechanical engineering from Nagoya University, Nagoya, Japan, in 1970, 1972, and 1978, respectively. He also received the M.E. degree in systems engineering from Case Western Reserve University, Cleveland, OH, in 1974. Since December 1990, he has been a Professor in the Department of Electrical Engineering, Nagoya University. His research interests are power electronics, robotics, and evolutionary soft computing. Dr. Okuma was the recipient of the IEEE IECON'92 Best Paper Award, in addition to paper awards from the Japan Society for Precision Engineering and Institute of Electrical Engineers of Japan.