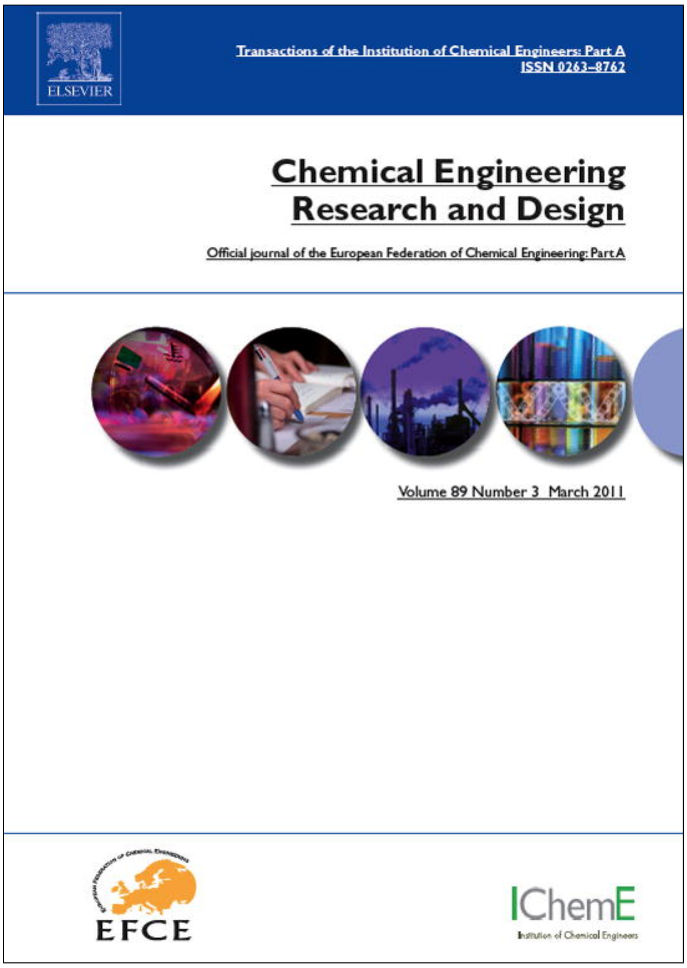

ELSEVIER

Transactions of the Institution of Chemical Engineers: Part A
ISSN 0263-8762

Chemical Engineering
Research and Design

Official journal of the European Federation of Chemical Engineering: Part A

Volume 89 Number 3 March 2011

EFCE

IChemE
Institution of Chemical Engineers

Contents lists available at ScienceDirect

Chemical Engineering Research and Design

journal homepage: www.elsevier.com/locate/cherd

Petri-net based scheduling strategy for semiconductor manufacturing processes

Yin-Hsuan Lee[a], Chuei-Tin Chang[a,*], David Shan-Hill Wong[b], Shi-Shang Jang[b]

[a] Department of Chemical Engineering, National Cheng Kung University, Tainan 70101, Taiwan, ROC
[b] Department of Chemical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan, ROC

ABSTRACT

Semiconductor manufacturing is a highly automated and capital-intensive industrial process. The operating cost of a wafer processing plant is in general closely related to the design and management of its process flows. Traditionally, the task of production scheduling is performed manually on the basis of past experiences. There are thus real incentives to develop a systematic approach to construct a mathematical programming model in order to reduce the chance of human errors and to ensure operational efficiency in implementing the resulting schedules. To this end, the Petri nets are adopted in this work to accurately model the semiconductor manufacturing activities. The token movements in a Petri net are represented with the well-established scheduling model for batch chemical processes, and the optimal schedule of the given semiconductor process can then be determined accordingly. The feasibility and effectiveness of this scheduling strategy is demonstrated in the present paper with three examples, i.e., the final test process, the re-entrant flow process, and the photolithography-etching process.

Keywords: Petri net; Scheduling; Semiconductor; MILP; MINLP

1. Introduction

As a result of the highly automated and capital-intensive nature of modern semiconductor manufacturing facilities, the design and management of their production processes have become an important research issue in recent years. Usually different types of products are processed simultaneously in a typical campaign. Each product must go through more than one stage and the operations in every stage can be performed in one out of several available tools. Traditionally, the production schedules in such an environment are stipulated manually in an ad hoc fashion on the basis of past experiences. Owing to the high level of process complexity, this scheduling task is often time-consuming, laborious and error-prone. To circumvent these drawbacks, there are obvious incentives to develop a computer-aided approach to facilitate systematic generation of optimal schedules for the semiconductor manufacturing operations.

There have been many scheduling related studies reported in the literature. A few examples are given below: Uzsoy et al. (1991) tried to use disjunctive graph to model the final test procedure; Johri (1993) and Duenyas et al. (1994) outlined the major challenges in planning and scheduling semiconductor processes; Chen et al. (1995) constructed an integer programming model to describe the test process in IC production and then solved this model with the Lagrangian relaxation technique; Lu and Kumar (1991) and Narahari and Khan (1996) applied FBFS (first-buffer–first-serve) and LBFS (last-buffer–first-serve) strategies to the re-entrant processes; Hsieh et al. (2001) explored the feasibility of applying the ordinal optimization-based simulation technique to efficiently select good scheduling rules for wafer fabrication; Hwang and Chang (2003) described the design of a two-level hierarchical production scheduling engine, which could be used to manage the mass production activities in semiconductor fabrication factories; Kumar et al. (2004) analyzed the re-entrant wafer production schemes on the basis of queuing theory; Gupta and Sivakumar (2006) presented a brief review of the scheduling techniques for semiconductor manufacturing processes, which include dispatching heuristics, mathematical programming models, neighborhood search methods, and AI methodologies; Pfund et al. (2008) modeled the semiconduc-

[*] Corresponding author. Fax: +886 6 234 4496.
E-mail address: ctchang@mail.ncku.edu.tw (C.-T. Chang).

Received 20 October 2009; Received in revised form 7 June 2010; Accepted 7 July 2010
0263-8762/$ – see front matter © 2010 The Institution of Chemical Engineers. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.cherd.2010.07.005

tor wafer fabrication process as a complex job shop, and adopted the Modified Shifting Bottleneck Heuristic (MSBH) to facilitate the multi-criteria optimization of makespan, cycle time, and total weighted tardiness using a desirability function; Sourirajan and Uzsoy (2007) proposed a rolling-horizon heuristic that decomposes the shop into smaller sub-problems that can be solved sequentially over time using a work center-based decomposition heuristic; Chou et al. (2008) studied the dynamic scheduling problem of semiconductor burn-in operations; Chen and Wang (2009) proposed a nonlinear scheduling rule with a fuzzy-neural remaining cycle time estimator to improve operation performance in a semiconductor manufacturing facility.

Due to the extreme complexity of multi-product, multi-stage and multi-tool processes, it is clear from the above studies that the wafer production schedule of any given system cannot be successfully synthesized without an appropriate model of its manufacturing activities. Since the Petri net has long been demonstrated to be suitable for characterizing and analyzing the discrete-event systems (Peterson, 1981; David and Alla, 1994), it was widely adopted for modeling and simulating the semiconductor production processes. Cavalieri et al. (1997) used the colored Petri nets to describe the semiconductor production sequences and, then, identified the minimum-cost schedules according to the corresponding reachability trees with a trial-and-error approach; Lin and Huang (1998) modeled the furnace in an IC fab also with colored-timed Petri net; Allam and Alla (1998) simulated and analyzed the assembling and testing processes on the basis of hybrid Petri nets; Zhou and Jeng (1998) performed a literature review on Petri-net applications concerning various semiconductor manufacturing systems; Jeng et al. (1998) modeled the etching area in an IC lot fabrication system with Petri nets, and also evaluated its performance accordingly; Xiong and Zhou (1998) built the Petri-net model for semiconductor test facility and applied heuristic search algorithm to identify test schedules; Jeng et al. (2000) developed Markovian timed Petri nets for performance analysis of semiconductor manufacturing systems; Kuo and Huang (2003) constructed colored-timed Petri-net models to design flexible processing routes for multiple products in IC fabs; Chiang et al. (2006) and Chien and Chen (2007) proposed Petri-net based scheduling models and solved with genetic algorithm (GA).

It can be observed from the aforementioned publications that, although the Petri nets are quite useful for modeling and simulating the semiconductor manufacturing activities, the numerical algorithms needed to identify a proper schedule (e.g., GA) may not be efficient enough. On the other hand, it should be noted that the scheduling methods used for batch chemical processes have advanced significantly in the last two decades. A large number of generalized models have been developed and applied successfully. For example, Pagageorgaki and Reklaitis (1990) proposed a MINLP model to generate the optimal schedules for multi-product batch processes; Kondili et al. (1993) solved this problem with a MILP program; Kim and Moon (2000) synthesized the multi-purpose schedules with Symbolic Model Verifier (SMV). It is worth noting that the mathematical program reported in Kondili et al. (1993) was formulated with discrete-time representation according to a graphic model of the batch process, i.e., the so-called state-task network (STN). Ierapetritou and Floudas (1998a,b) later proposed a STN-based MILP model for producing an optimal batch schedule. An equivalent Resource Task Network (RTN) representation was proposed by Pantelides (1994). Zhang and Sargent (1996, 1998) provided a unified mathematical formulation to determine the optimal operating conditions of RTN in continuous-time representation. The STN-based model later became a popular choice for many scheduling applications due to its capability to capture the equipment-sharing possibilities. Various other mathematical programs have also been constructed accordingly, e.g., Shah et al. (1993) and Maravelias and Grossman (2003). An overview of the continuous-time versus discrete-time approaches for scheduling multi-product and/or multi-purpose batch processes was presented by Floudas and Lin (2004). Finally, a STN-based cyclic scheduling strategy has also been proposed by Wu and Ierapetritou (2004). They essentially modified the aforementioned short-term model (Ierapetritou and Floudas, 1998a,b) with additional constraints to accommodate the unique requirements in periodic operations.

From the above discussions, it is obvious that a specific state-task network must be created before constructing the schedule-generating models. However, since this representation is too simple to capture all dynamic features embedded in semiconductor manufacturing processes, it is often difficult to build an appropriate STN model and check its correctness in a systematic and efficient manner. On the other hand, notice that the Petri net is not only a suitable model of the wafer production operations but also a convenient simulation tool. The latter capability can be quite useful in validating the model and also verifying the feasibility of any given schedule. Therefore, it is the intention of this work to develop a Petri-net based scheduling procedure. The main steps of this procedure can be summarized as follows:

- Construction of a Petri-net model for characterizing all manufacturing activities in the given semiconductor process,
- formulation of a mathematical programming model for describing the token movements in Petri net, and
- generation of the optimal production schedules from the solution of the above-mentioned mathematical program.

2. Petri-net models

A generalized model construction method is described here. It should be first noted that the Petri nets used in this study are colorless but timed. The places in a net can be classified into two general types, i.e., tool states and buffer states, while the transitions represent operation stages exclusively. The token number assigned to each place can only assume positive integer values and the arc weights are all one. Real time delays can be assigned to the transitions to signify processing times needed in the corresponding operation stages.

With the aforementioned conventions, a Petri-net model can be constructed easily according to the given process data. Two examples are provided below to illustrate this construction method.

2.1. *Example 1*

After completing the IC packaging operations, a series of final tests (FTs) have to be performed on the finished products. A typical FT process may consist of several different operation stages. The hardware facilities needed in each stage include: tester, handler and other auxiliary equipments, and these facilities can be viewed as the components of a *work center*. Xiong and Zhou (1998) have developed the Petri-net models of

Table 1 – Work centers and their processing times in Example 1.

Stage	Job 1	Job 2	Job 3	Job 4
1	(M1, 2)	(M3, 4)	(M1, 3)	(M2, 3)
2	(M2, 3)	(M1, 2)	(M3, 5)	(M3, 4)
3	(M3, 4)	(M2, 2)	(M2, 3)	(M1, 3)

(1 time unit = 10 min).

final test (FT) *jobs*. In one of their examples, four distinct jobs were considered. To complete each job, three different stages are needed and each must be carried out in a dedicated work center. The work centers and the corresponding processing times required for implementing the operation stages in every job are listed in Table 1. Let us further assume that these work centers do not share common components and thus they can be operated independently.

On the basis of Table 1, the corresponding Petri-net model can be constructed easily according to the proposed conventions (see Fig. 1). In this model, transition t_{ij} represents the operation in stage j of job i ($i = 1, 2, 3, 4; j = 1, 2, 3$); place P_{ik} denotes the buffer state k of job i ($i = 1, 2, 3, 4; k = 1, 2, 3, 4$); place M_l denotes the state of lth machine or work center (i.e., tool state). Finally notice that the processing times are used as the delay times of the transitions.

2.2. *Example 2*

In a semiconductor manufacturing plant, the wafers are often processed in many different stages by following re-entrant flow patterns, in which the operations in different stages may be carried out with the same tool. Due to the complexity of re-entrant flow processes, it is highly desirable to identify the bottlenecks in a production schedule in advance. Petri-net model is an ideal vehicle for this purpose. As an example, let us consider Fig. 2, which is a slightly modified version of the system studied by Odrey et al. (2001). Notice that the wafers follow a re-entrant route among three work centers. Center 1 is used for performing the operations in four different stages, center 2 is for three stages, and center 3 is for two stages. There are also buffers between work centers to store the wafers waiting in queues.

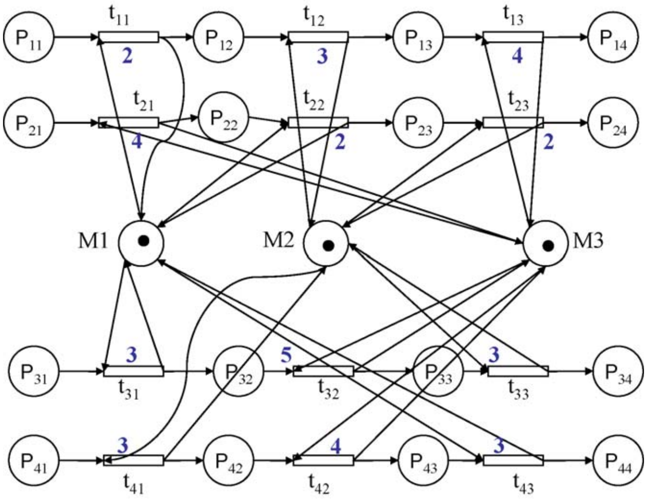

Fig. 1 – Petri-net model of a FT process with 4 jobs, 3 stages and 3 work centers.

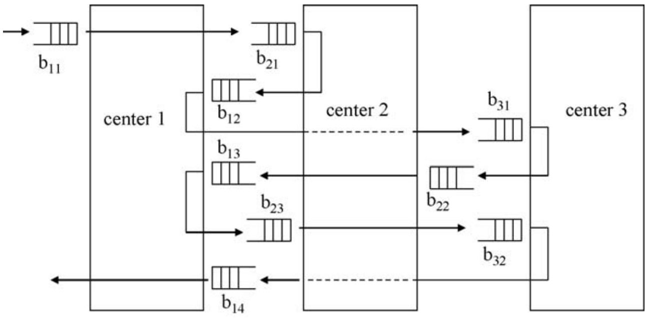

Fig. 2 – An open-loop re-entrant flow process with 3 work centers.

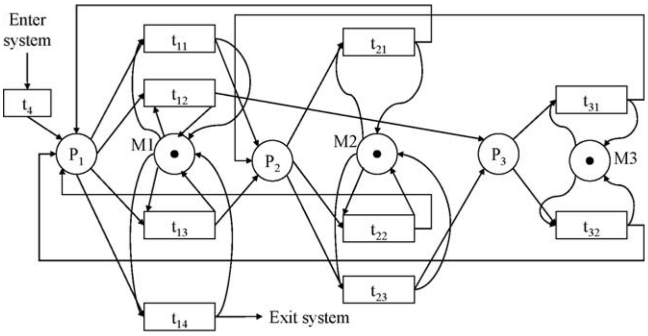

Fig. 3 – Petri-net model of a re-entrant flow process.

The corresponding Petri net can be found in Fig. 3, where P_i ($i = 1, 2, 3$) denotes the state of buffer i, M_j ($j = 1, 2, 3$) represents the tool state of center j, t_{jk} ($j = 1, 2, 3; k = 1, 2, \ldots$) is the kth operation stage performed by work center j, t_4 is used to denote the event of wafer entering system. Notice that the buffer-state representation in this model suffers an obvious drawback, i.e., the states of products stored in the same buffer are indistinguishable. For the sake of clarity, the corresponding places, i.e., $P_1 - P_3$, can be artificially classified according to the stages. The resulting Petri net is given in Fig. 4, in which P_{jk} ($j = 1,2,3; k = 1, 2, \ldots$) denotes the artificial buffer state before executing operation k in work center j. Note that, in the scheduling model described later, an upper capacity limit may have to be imposed on all artificial buffer states associated with the same work center.

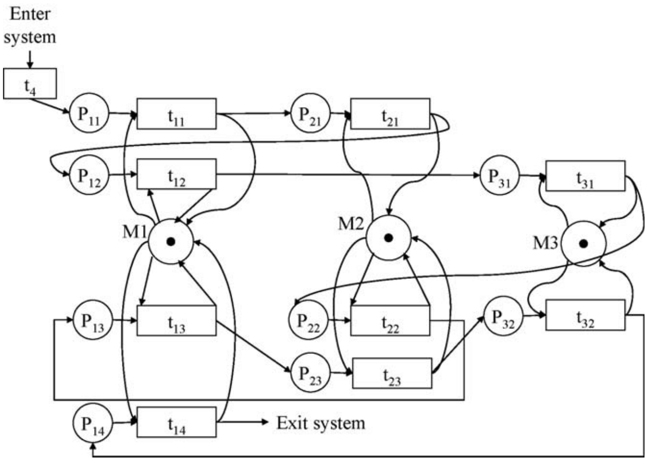

Fig. 4 – Petri-net model of a re-entrant flow process with artificially classified buffer states.

3. Optimal scheduling strategy

Given a feasible schedule, the Petri-net model can be used as the basis for simulating the production activities in a semiconductor manufacturing plant. Since the *best* production schedule cannot always be conveniently identified from such a graphic model, the token movements in a Petri net are alternatively described in this study with the constraints of mathematical programming model. An optimal schedule can then be generated by solving this model according to any given objective function.

Ierapetritou and Floudas (1998a,b) introduced a conceptual term, *event point*, to denote the instance when a task (and also the use of corresponding unit) begins or ends. On the basis of this idea, a binary variable $am(t, m, n) \in \{0, 1\}$ is adopted in this work to reflect the conditions of transition t and the place associated with tool m. Specifically, $am(t, m, n') = 1$ denotes that, starting from the event point n', transition t is *enabled* by the state of place m, i.e., one or more token is present in this place. These conditions are usually maintained for a finite period of time, which is denoted as $delay(t, m)$, until the enabled transition t can be fired. On the other hand, $am(t, m, n'') = 0$ simply means that transition t cannot be enabled at event point n'' by the state of place m.

The indices, sets, parameters and variables used in the proposed model are defined below:

- Indices
  - m: the place label associated with a tool;
  - n: the label of an event point;
  - p: the place label representing an actual buffer;
  - s: the place label denoting an artificial buffer;
  - t: the label of a transition.
- Sets
  - M: the set of all places representing the states of available tools;
  - M_t: a subset of M in which the places are all connected to transition t;
  - N: the set of all event points, i.e., $\{1, 2, \ldots, N\}$;
  - P: the set of all places representing the states of actual buffers;
  - S: the set of all places representing the states of artificial buffers;
  - S_p: a subset of S in which all places are associated with actual buffer $p \in P$;
  - T: the set of all transitions;
  - T_m: a subset of T in which all transitions are connected to place $m \in M$;
  - T_s^{in}: a subset of T which contains all input transitions of place $s \in S$;
  - T_s^{out}: a subset of T which contains all output transitions of place $s \in S$.
- Parameters
  - $delay(t, m)$: the delay time needed to fired transition $t \in T_m$ after it is enabled;
  - $st^I(s)$: the initial token number in place $s \in S$;
  - $st^U(p)$: the maximum token number allowed in place $p \in P$.
- Variables
  - $am(t, m, n)$: the binary variable used to reflect if transition $t \in T$ is enabled by the state of place $m \in M_t$ at event point $n \in N$;
  - H: the time horizon of the production campaign;
  - $st(s, n)$: the token number in place $s \in S$ at event point $n \in N$;
  - $T^E(t, m, n)$: the time at event point $n \in N$ when transition $t \in T$ can be enabled by the state of place $m \in M_t$;
  - $T^F(t, m, n)$: the time at event point $n \in N$ when transition $t \in T$ can be fired by the state of place $m \in M_t$.

3.1. Token movements

The token movements in a Petri net are created by *firing* transitions. Since an event point is regarded in this work as the instance when a transition starts to be enabled, the difference in token numbers at every place $s \in S$ between two consecutive event points can therefore be expressed as

$$st(s, n) - st(s, n-1) = \sum_{t \in T_s^{in}} \sum_{m \in M_t} am(t, m, n-1) - \sum_{t' \in T_s^{out}} \sum_{m' \in M_{t'}} am(t', m', n) \qquad \forall s \in S \quad \forall n \in N \tag{1}$$

where $st(s, 0) = st^I(s)$. To improve solution efficiency of the proposed optimization problem, it is necessary to set all unused binary variables to be zeros, i.e.:

$$am(t, m, n) = 0 \qquad \forall t \in T \quad \forall m \in M_t/M \quad \forall n \in N \tag{2}$$

Since a transition can only be fired after it has been enabled for a specified period of delay time, the firing time $T^F(t, m, n)$ should be calculated according to the enabling time $T^E(t, m, n)$ on the same event point, i.e.:

$$T^F(t, m, n) = T^E(t, m, n) + delay(t, m)am(t, m, n) \qquad \forall t \in T \quad \forall m \in M_t \quad \forall n \in N \tag{3}$$

For the same transition t, the enabling time and firing time at the current event point should be earlier than the corresponding times at the subsequent event point, i.e.:

$$T^E(t, m, n+1) \geq T^E(t, m, n) \tag{4}$$

$$T^F(t, m, n+1) \geq T^F(t, m, n) \qquad \forall t \in T \quad \forall m \in M_t \quad \forall n \in N \quad n \neq N \tag{5}$$

At a particular time instance, if a transition has been enabled but not fired for an event point, the same transition is not allowed to be enabled again for the subsequent event point. In other words, the following constraint should be imposed:

$$T^E(t, m, n+1) \geq T^F(t, m, n) - H[1 - am(t, m, n)] \qquad \forall t \in T \quad \forall m \in M_t \quad \forall n \in N \quad n \neq N \tag{6}$$

If two or more transitions can be enabled by the same place $m \in M$, then one of them (i.e., $t \in T_m$) can be enabled at an event point only after another one is enabled and then fired at the prior event point, i.e.:

$$T^E(t, m, n+1) \geq T^F(t', m, n) - H[1 - am(t', m, n)] \qquad \forall m \in M \quad \forall t, t' \in T_m \quad t \neq t'_t \quad \forall n \in N \quad n \neq N \tag{7}$$

Since the precedence order of enabling and then firing any pair of neighboring transitions (say $t \in Tm$, $t' \in Tm'$ and $t \neq t'$) is unambiguously specified in the given Petri net, this inherent feature must be stipulated in the model as

$$T^E(t, m, n+1) \geq T^F(t', m', n) - H[1 - am(t', m', n)] \quad \forall m \in M_t \quad \forall m' \in M_{t'} \quad m \neq m' \quad \forall n \in N \quad n \neq N \tag{8}$$

At event point $n+1$, the enabling action of transition $t \in T_m$ should occur after every transition in the same set, i.e., $\forall t' \in T_m$, is fired at all prior event points, i.e.:

$$T^E(t, m, n+1) \geq \sum_{\substack{n' \in N \\ n' \leq n}} \sum_{t' \in T_m} [T^F(t', m, n') - T^E(t', m, n')] \quad \forall m \in M \quad \forall t \in T_m \quad n \in N \quad n \neq N \tag{9}$$

Obviously, enabling and firing of all transitions must be completed within the production horizon, i.e.:

$$T^E(t, m, n) \leq H \tag{10}$$

$$T^F(t, m, n) \leq H \quad \forall t \in T \quad \forall m \in M_t \quad \forall n \in N \tag{11}$$

3.2. *Shared resources*

Since more than one transition may be present in set T_m, it is necessary to make sure that at most one transition can be enabled by place m at every event point:

$$\sum_{t \in T_m} am(t, m, n) \leq 1 \quad \forall m \in M \quad \forall n \in N \tag{12}$$

Since more than one place may be present in set M_t, it is also necessary to make sure that only the token in one of them is removed after firing transition t at any event point:

$$\sum_{m \in M_t} am(t, m, n) \leq 1 \quad \forall t \in T \quad \forall n \in N \tag{13}$$

Finally, the capacity limit of an actual buffer can also be imposed

$$\sum_{s \in S_p} st(s, n) \leq st\max(p) \quad p \in P \quad n \in N \tag{14}$$

3.3. *Linearization of minimum-horizon model*

Several different objective functions can be adopted to produce the optimal production schedule. One of them is for achieving the minimum horizon, i.e., min H. Due to Eqs. (6)–(8), the resulting optimization problem becomes a mixed integer nonlinear program (MINLP). These nonlinear constraints can be converted into linear forms by introducing extra real variables and additional logic constraints. For illustration convenience, let us consider Eq. (6) as an example. In this case, the new variable can be defined as

$$h_{am}(t, m, n) = Ham(t, m, n) \tag{15}$$

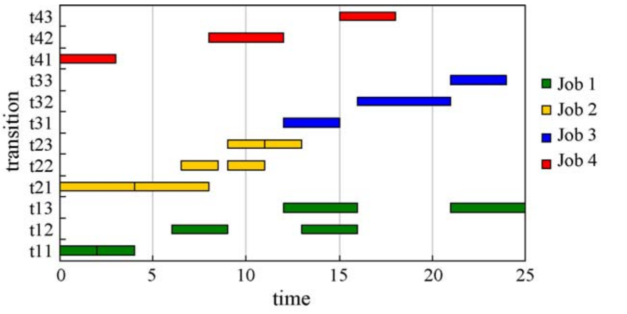

Fig. 5 – Gantt chart for transitions in Example 1–Scenario 1.

Eq. (6) can then be transformed to

$$T^s(t, m, n+1) \geq T^f(t, m, n) - H + h_{am}(t, m, n) \tag{16}$$

The value of the new variable $h_{am}(t, m, n)$ can be set with the following two logic constraints:

$$0 \leq h_{am}(t, m, n) \leq M_1 am(t, m, n) \tag{17}$$

$$0 \leq H - h_{am}(t, m, n) \leq M_2[1 - am(t, m, n)] \tag{18}$$

where M_1 and M_2 are large enough positive numbers.

4. Case studies

Three case studies are presented in the sequel to demonstrate the effectiveness of the proposed modeling approach. Production schedules of the semiconductor manufacturing processes described previously in Examples 1 and 2 are discussed in Case 1 and Case 2, respectively. The third is concerned with a simplified photolithography-etching process. Let us consider these three cases one-by-one in sequence.

4.1. *Case 1*

Let us first consider the FT process described in Example 1 and its Petri-net model in Fig. 1. A minimum-horizon schedule can be generated for specific workloads. The optimization results obtained for a simple scenario, i.e., testing 2, 2, 1 and 1 lots of products in job 1–4, respectively, are presented in Figs. 5 and 6. The Gantt chart for the enabled transitions in Petri net (or the operation stages in different jobs of the FT process) is given in Fig. 5. It is assumed in this example that the time unit is 10 min and thus the minimum horizon is 250 min. Notice that, due to the need to share machines, the operation stages in the same job may not be executed consecutively. The work schedule of machines can be identified according to Figs. 1 and 5 (see Fig. 6). It can be observed that, since the processing time of M3 is the longest among the three machines, the main focus of scheduling arrangements should therefore be placed upon minimization of its total operation time. The same optimization run has also been repeated for a larger scheduling task. In particular, the workloads of the four test jobs are increased

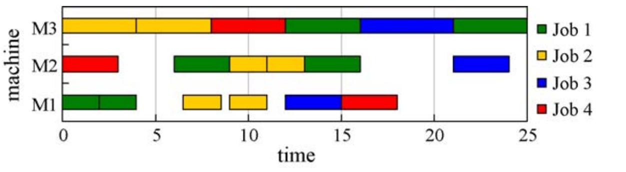

Fig. 6 – Gantt chart for machines in Example 1–Scenario 1.

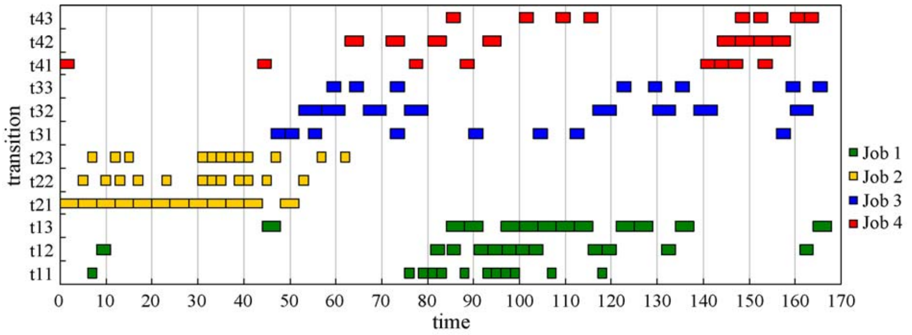

Fig. 7 – Gantt chart for transitions in Example 1–Scenario 2.

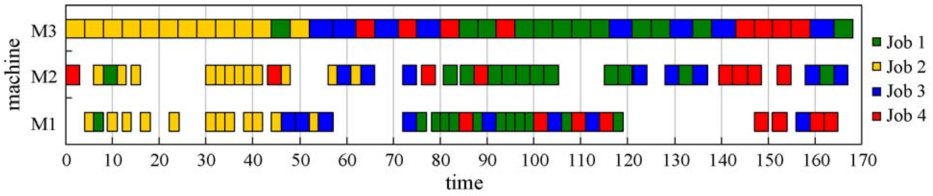

Fig. 8 – Gantt chart for machines in Example 1–Scenario 2.

to 12, 12, 8 and 8 lots, respectively in the second scenario. The minimum horizon in this case is 168 time units (1680 min), and the corresponding Gantt charts for the transitions and machines can be found in Figs. 7 and 8, respectively.

4.2. Case 2

As mentioned previously, the re-entrant flow pattern is one of the unique features that can be observed in semiconductor manufacturing facilities. A typical two-job four-machine process is considered in this case study (see Fig. 9). The corresponding Petri net (Fig. 10) can be obtained by following the proposed model-building method. Notice that the delay time of each transition (i.e., the processing time of corresponding operation stage) is also specified in this model. Let us further assume that the two resulting products are of equivalent qualities and, thus, can be used interchangeably. By fixing the total inventory of both types of unprocessed wafers to be 13 lots and solving the scheduling model, the minimum horizon can be determined to be 1830 min and the corresponding optimal initial states of buffers s_{11} and s_{21} are 5 lots and 8 lots, respectively. The Gantt chart of the enabled transitions can be found in Fig. 11, while that of the working machines is depicted in Fig. 12.

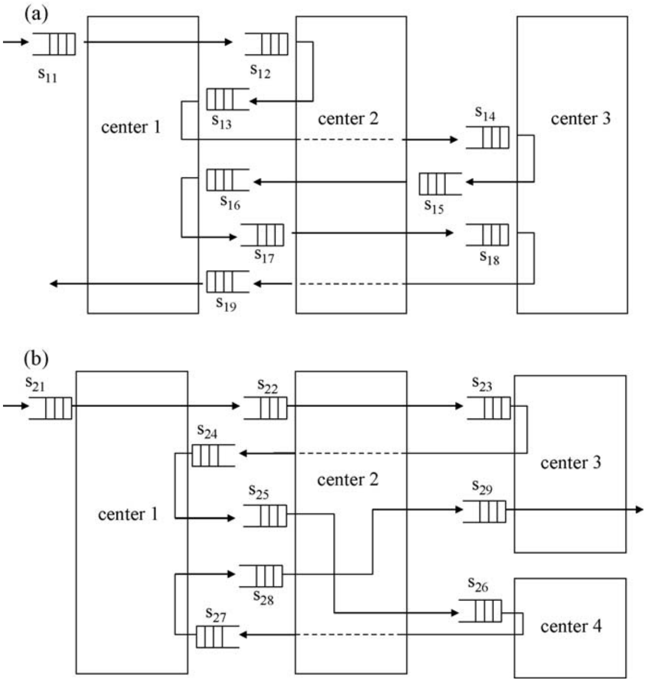

Fig. 9 – A typical re-entrant process with three machines: (a) job 1; (b) job 2.

The total operation period can obviously be reduced by adding more machines on-line. For example, let us consider the scenario when an additional *Center 1* is included in the plant. Notice that an extra place must be inserted in the Petri net in Fig. 10 to reflect the state of this new tool and, also, the same set of transitions should be connected to/from both M1 and this added place. The resulting Gantt charts can also be generated by solving the corresponding scheduling model (see Figs. 13 and 14). As expected, the minimum horizon in this scenario can be shortened to 1780 min.

4.3. Case 3

The typical production environment in a semiconductor plant resembles an automated assembly line in which many similar types of products with different specifications are manufactured by a step-by-step overall process. Each step is a complicated physiochemical batch process that can be carried out by a number of tools working in parallel. In the previous two examples only the efficiency related measures, e.g., throughput or horizon, are considered for stipulating the scheduling policy. However, it should be noted that the overall quality of every type of products is in general considered to be very sensitive to the combination of tools used in production. To address the efficiency and quality issues simultaneously, the constrained optimization problem of how to minimize campaign horizon but maintain quality standards is studied in this example.

Let us consider the Petri net given in Fig. 15, which is the model of a two-step two-job system. Since the system configuration here is quite simple, a description of the model-building procedure is omitted for the sake of brevity. Notice that the

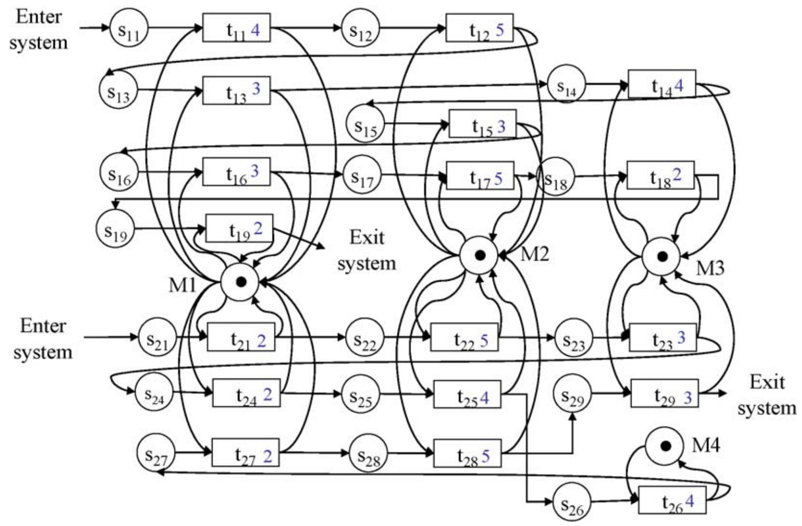

Fig. 10 – The Petri-net model of a three-machine two-job re-entrant process.

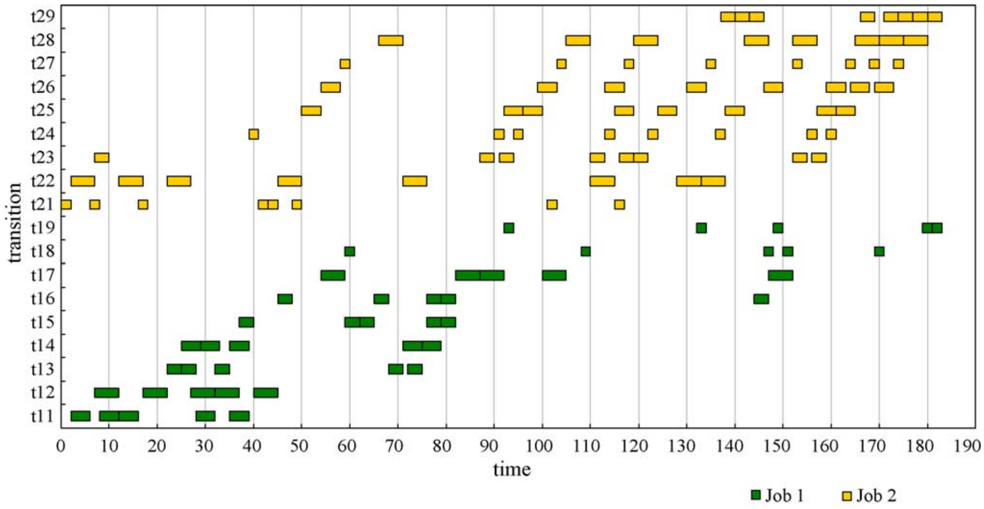

Fig. 11 – Gantt chart for transitions in Example 2–Scenario 1.

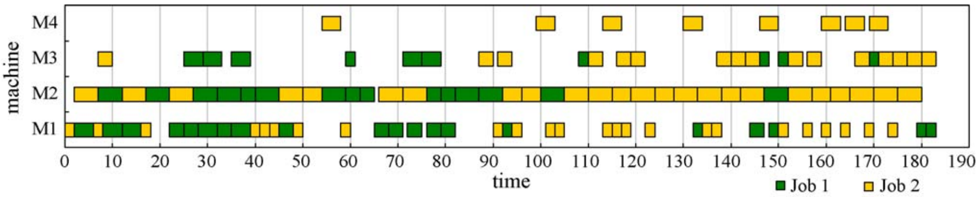

Fig. 12 – Gantt chart for machines in Example 2–Scenario 1.

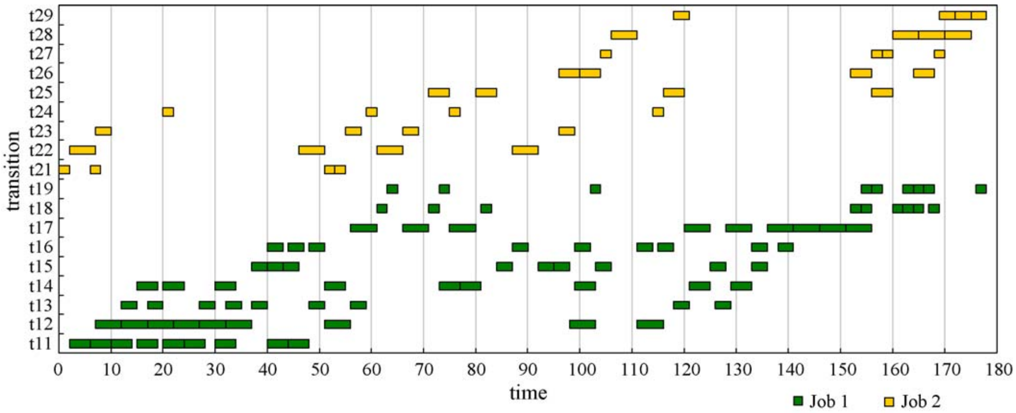

Fig. 13 – Gantt chart for transitions in Example 2–Scenario 2.

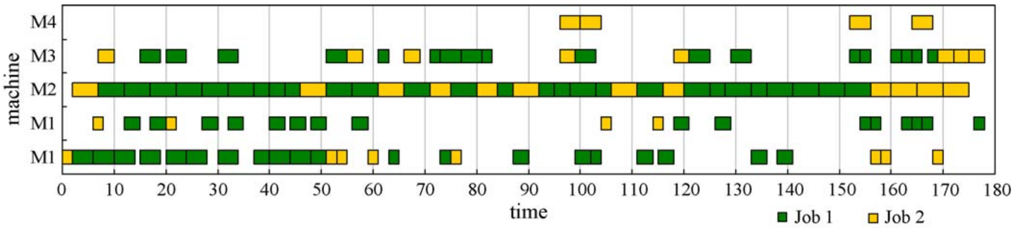

Fig. 14 – Gantt chart for machines in Example 2–Scenario 2.

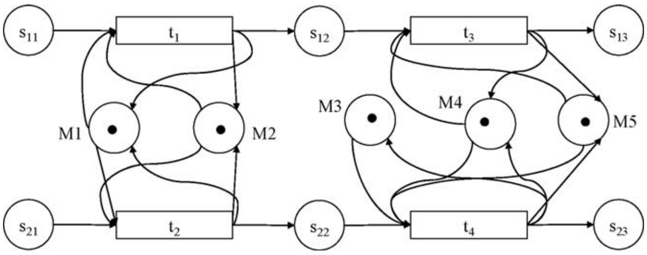

Fig. 15 – Petri-net model of a simplified photolithography-etching process.

first operation step (say photolithography) can be carried out in one of the two alternative tools, i.e., M1 and M2. The tool M3 is dedicated to the task of processing the second product in the subsequent step (say etching), while either M4 or M5 can be used to perform the same operation for both products. Thus, four possible combinations, i.e., (M1, M4), (M1, M5), (M2, M4) and (M2, M5), can be utilized to carry out either job, but two additional selections, i.e., (M1, M3) and (M2, M3), are available for implementing the second job exclusively. Finally, the delay times of the first-step transitions are both assumed to be 0.5 h and those in the second steps are 0.3 h.

Let us next use f_u^p to represent the fraction of products p which is processed in the uth combination. Thus, the following constraints should be imposed in the scheduling model:

$$\sum_{u=1}^{U_p} f_u^p = 1, \quad 0 \le f_u^p \le 1 \tag{19}$$

where $p = 1, 2, \ldots, P$. Let us further assume that the mean and variance of characteristic quality variable of *finished* products p produced in combination u (denoted as y_u^p) can be *estimated* in advance to be $\tilde{\mu}[y_u^p]$ and $\tilde{\sigma}^2[y_u^p]$, respectively. The resulting mean and variance for final products p generated from all combinations, i.e., $\tilde{\mu}[y^p]$ and $\tilde{\sigma}^2[y^p]$, can be determined according to the following equations:

$$\tilde{\mu}[y^p] = \sum_{u=1}^{U_p} f_u^p \tilde{\mu}[y_u^p] \tag{20}$$

$$\tilde{\sigma}^2[y^p] = \sum_{u=1}^{U_p} f_u^p \{(\tilde{\mu}[y_u^p])^2 + \tilde{\sigma}^2[y_u^p]\} - \left\{\sum_{u=1}^{U_p} f_u^p \tilde{\mu}[y_u^p]\right\}^2 \tag{21}$$

Table 2 – Estimated means and variances of finished products in Example 3.

Product	u	Combination	$\tilde{\mu}[y_u^p]$ (nm)	$\tilde{\sigma}^2[y_u^p]$ (nm)
1	1	M1, M4	−0.12	1.0
	2	M1, M5	−0.57	0.9
	3	M2, M4	−0.15	0.7
	4	M2, M5	−0.08	0.7
2	1	M1, M3	−0.19	1.3
	2	M1, M4	0.74	0.8
	3	M1, M5	0.71	0.6
	4	M2, M3	−0.40	0.6
	5	M2, M4	−0.58	0.6
	6	M2, M5	−0.21	1.4

A so-called *process capability index* of products p can be defined as (Del Castillo, 2002):

$$C_{pk}[y^p] = \left(\frac{USL^p - \tilde{\mu}[y^p]}{3\tilde{\sigma}[y^p]}, \frac{\tilde{\mu}[y^p] - LSL^p}{3\tilde{\sigma}[y^p]}\right)_{\min} \tag{22}$$

This index can be regarded as a measure of the capability of the process to achieve in-control values that lie in the range between USL^p and LSL^p. In order to incorporate the implied logic into the mathematical programming model, the following constraints must be introduced

$$MIx \ge [(\tilde{\mu}[y^p] - LSL^p) - (USL^p - \tilde{\mu}[y^p])]Ix \ge 0 \tag{23}$$

$$MIy \ge [(USL^p - \tilde{\mu}[y^p]) - (\tilde{\mu}[y^p] - LSL^p)]Iy \ge 0 \tag{24}$$

$$Ix + Iy = 1 \tag{25}$$

$$C_{pk}[y^p] = Ix\frac{(USL^p - \tilde{\mu}[y^p])}{3\tilde{\sigma}[y^p]} + Iy\frac{(\tilde{\mu}[y^p] - LSL^p)}{3\tilde{\sigma}[y^p]} \tag{26}$$

where $Ix, Iy \in \{0, 1\}$ and M is a large positive real number. Based on past experience, the processing ability can be considered as good enough to be on target if

$$C_{pk}[y^p] > 1.33 \tag{27}$$

In this example, all upper and lower control limits are chosen to be 3.2 and −3.2, respectively. As mentioned before, the values of $\tilde{\mu}[y_u^p]$ and $\tilde{\sigma}^2[y_u^p]$ are treated as given parameters, and these values are listed in Table 2.

By imposing the aforementioned additional constraints, i.e., Eqs. (19)–(27), in the scheduling model, the minimum

Table 3 – Optimal solution obtained in Example 3 – product quality.

Product 1 ($p = 1$)			Product 2 ($p = 2$)		
$\tilde{\mu}[y^p]$ (nm)	$\tilde{\sigma}[y^p]$ (nm)	$C_{pk}[y^p]$	$\tilde{\mu}[y^p]$ (nm)	$\tilde{\sigma}[y^p]$ (nm)	$C_{pk}[y^p]$
−0.1313	0.7007	1.4598	0.7100	0.6000	1.3833

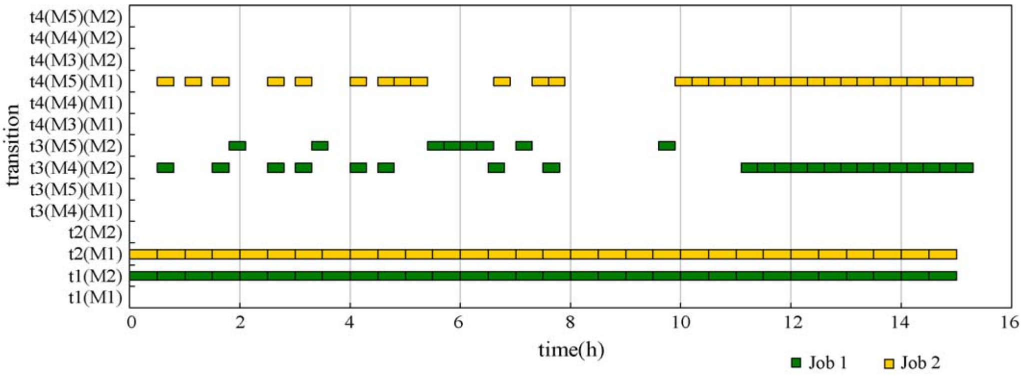

Fig. 16 – Gantt chart for transitions in Example 3.

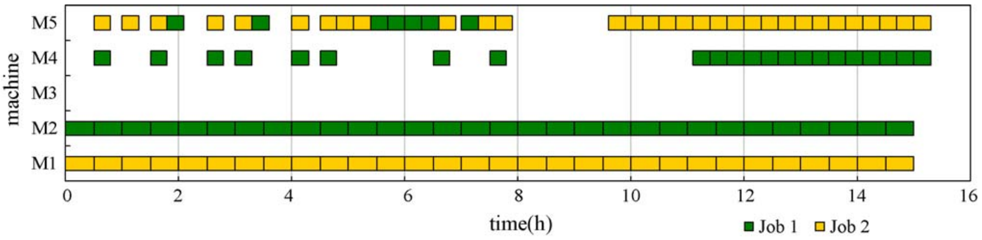

Fig. 17 – Gantt chart for machines in Example 3.

Table 4 – Optimal solution obtained in Example 3 – production sequence.

Product	1				2					
Step 1	M1	M1	M2	M2	M1	M1	M1	M2	M2	M2
Step 2	M4	M5	M4	M5	M3	M4	M5	M3	M4	M5
Quantity (lots)			22	8			30			
Fraction			0.7333	0.2667			1.0000			

horizon can be found to be 15.3 h. The Gantt charts for the transitions and also the machines (tools) are presented in Figs. 16 and 17, respectively. Finally, critical features of the corresponding optimal solution can be found in Tables 3 and 4.

5. Conclusions

A systematic approach is described in this paper to model typical semiconductor manufacturing activities with Petri nets. A mathematical program can then be constructed accordingly to generate the most efficient work schedules. Extra constraints can also be easily incorporated in this scheduling model to address other important issues (e.g., quality) in wafer production. From the optimization results presented in the case studies, it can be clearly observed that the proposed modeling strategy is feasible and effective for various moderately sized problems.

Nomenclature

$am(t,m,n)$	the binary variable used to reflect if transition $t \in T$ is enabled by the state of place $m \in M_t$ at event point $n \in N$;
$delay(t,m)$	the delay time needed to fired transition $t \in T_m$ after it is enabled;
H	the time horizon of the production campaign;
m	the place label associated with a tool;
M	the set of all places representing the states of available tools;
M_t	a subset of M in which the places are all connected to transition t;
n	the label of an event point;
N	the set of all event points, i.e., $\{1, 2, \ldots, N\}$;
p	the place label representing an actual buffer;
P	the set of all places representing the states of actual buffers;
s	the place label denoting an artificial buffer;
S	the set of all places representing the states of artificial buffers;
S_p	a subset of S in which all places are associated with actual buffer $p \in P$;
$st(s,n)$	the token number in place $s \in S$ at event point $n \in N$;
$st^I(s)$	the initial token number in place $s \in S$;
$st^U(p)$	the maximum token number allowed in place $p \in P$;
t	the label of a transition;
T	the set of all transitions;
T_m	a subset of T in which all transitions are connected to place $m \in M$;
T_s^{in}	a subset of T which contains all input transitions of place $s \in S$;
T_s^{out}	a subset of T which contains all output transitions of place $s \in S$;
$T^E(t,m,n)$	the time at event point $n \in N$ when transition $t \in T$ can be enabled by the state of place $m \in M_t$;
$T^F(t,m,n)$	the time at event point $n \in N$ when transition $t \in T$ can be fired by the state of place $m \in M_t$.

References

Allam, M., Alla, H., 1998. Modeling and simulation of an electronic component manufacturing system using hybrid petri nets. IEEE Transactions on Semiconductor Manufacturing 11 (3), 374–383.

Cavalieri, S., Mirabella, O., Marroccia, S., 1997. Improving flexible semiconductor manufacturing system performance by a colored Petri net-based scheduling algorithm. In: IEEE Sixth International Conference on Emerging Technologies and Factory Automation proceedings, Los Angeles, CA, pp. 369–374.

Chen, T., Wang, Y.C., 2009. A nonlinear scheduling rule incorporating fuzzy-neural remaining cycle time estimator for scheduling a semiconductor manufacturing factory. International Journal of Advanced Manufacturing Technology 45, 110–121.

Chen, T.R., Chang, T.S., Chen, C.W., 1995. Scheduling for IC sort and test with preemptiveness via Lagrangian-relaxation. IEEE Transactions on Systems Man and Cybernetics 25, 1249–1250.

Chiang, T.C., Huang, A.C., Fu, L.C., 2006. Modeling, scheduling, and performance evaluation for wafer fabrication: a queuing colored petri-net and GA-based approach. IEEE Transactions on Automation Science and Engineering 3 (3), 330–337.

Chien, C.F., Chen, C.H., 2007. Using genetic algorithms (GA) and a coloured timed Petri net (CTPN) for modeling the optimization-based schedule generator of a generic production scheduling system. International Journal of Production Research 45 (8), 1763–1789.

Chou, F.D., Chang, P.C., Wang, H.M., 2008. A simulated annealing approach with probability matrix for semiconductor dynamic scheduling problem. Expert Systems with Applications 35, 1889–1898.

David, R., Alla, H., 1994. Petri Nets for modeling of dynamic-systems—a survey. Automatica 30, 175–202.

Del Castillo, E., 2002. Statistical Process Adjustment for Quality Control. John Wiley & Sons, New York.

Duenyas, I., Fowler, J.W., Schruben, L.W., 1994. Planning and scheduling in Japanese semiconductor manufacturing. Journal of Manufacturing Systems 13, 323–332.

Floudas, C.A., Lin, X., 2004. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. Computers & Chemical Engineering 28, 2109.

Gupta, A.K., Sivakumar, A.I., 2006. Job shop scheduling techniques in semiconductor manufacturing. International Journal of Advanced Manufacturing Technology 27, 1163–1169.

Hsieh, B.W., Chen, C.H., Chang, S.C., 2001. Scheduling semiconductor wafer fabrication by using ordinal optimization-based simulation. IEEE Transactions on Robotics and Automation 17 (5), 599–608.

Hwang, T.K., Chang, S.C., 2003. Design of a Lagrangian relaxation-based hierarchical production scheduling environment for semiconductor wafer fabrication. IEEE Transactions on Robotics and Automation 19 (4), 566–578.

Ierapetritou, M.G., Floudas, C.A., 1998a. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. Industrial & Engineering Chemistry Research 37, 4341–4359.

Ierapetritou, M.G., Floudas, C.A., 1998b. Effective continuous-time formulation for short-term scheduling. 2. Multipurpose/multiproduct continuous processes. Industrial & Engineering Chemistry Research 37, 4360.

Jeng, M.D., Xie, X.L., Chou, S.W., 1998. Modeling, qualitative analysis, and performance evaluation of the etching area in an IC lot fabrication system using Petri nets. IEEE Transactions Semiconductor Manufacturing 11, 358–373.

Jeng, M.D., Xie, X.L., Hung, W.Y., 2000. Markovian timed Petri nets for performance analysis of semiconductor manufacturing systems. IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics 30 (5), 757–771.

Johri, P.K., 1993. Practical issues in scheduling and dispatching in semiconductor wafer fabrication. Journal of Manufacturing Systems 12, 474–485.

Kim, J., Moon, I., 2000. Synthesis of safe operating procedure for multi-purpose batch processes using SMV. Computers & Chemical Engineering 24, 385.

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations. I. MILP formulation. Computers & Chemical Engineering 17, 211.

Kumar, R., Tiwari, M.K., Allada, V., 2004. Modeling and rescheduling of a re-entrant wafer fabrication line involving machine unreliability. International Journal of Production Research 42 (21), 4431–4455, 1.

Kuo, C.H., Huang, H.P., 2003. Distributed performance evaluation of a controlled IC fab. IEEE Transactions on Robotics and Automation 19 (6), 1027–1033.

Lu, S.H., Kumar, P.R., 1991. Distributed scheduling based on due dates and buffer priorities. IEEE Transactions on Automatic Control 36, 1406–1416.

Lin, S.Y., Huang, H.P., 1998. Modeling and emulation of a furnace in IC fab based on colored-timed Petri net. IEEE Transactions on Semiconductor Manufacturing 11 (3), 410–420.

Maravelias, C.T., Grossman, I.E., 2003. A new general continuous-time state task network formulation for the short term scheduling of multi-purpose batch plants. Industrial & Engineering Chemistry Research 42, 3056.

Narahari, Y., Khan, L.M., 1996. Performance analysis of scheduling policies in re-entrant manufacturing systems. Computers and Operations Research 23 (1), 37–51.

Odrey, N.G., Green, J.D., Appello, A., 2001. A generalized Petri net modeling approach for the control of re-entrant flow semiconductor wafer fabrication. Robotics and Computer Integrated Manufacturing 17, 5–11.

Pantelides, C.C., 1994. Unified frameworks for the optimal process planning and scheduling. In: Rippin, D.W.T., Hale, J. (Eds.), Proceedings of the 2nd Conference on Foundations of Computer-Aided Process Operations, p. 253.

Pagageorgaki, S., Reklaitis, G.V., 1990. Optimal design of multipurpose batch plants. 1. Problem formulation. Industrial & Engineering Chemistry Research 29, 2054–2062.

Peterson, J.L., 1981. Petri Net Theory and Modeling of Systems. Prentice-Hall, Englewood Cliffs, New Jersey.

Pfund, M.E., Balasubramanian, H., Fowler, J.W., Mason, S.J., Rose, O., 2008. A multi-criteria approach for scheduling semiconductor wafer fabrication facilities. Journal of Scheduling 11, 29–47.

Shah, N., Pantelides, C.C., Sargent, R., 1993. A general algorithm for short-term scheduling of batch operations. II. Computational issues. Computers & Chemical Engineering 17 (2), 229–244.

Sourirajan, K., Uzsoy, R., 2007. Hybrid decomposition heuristics for solving large-scale scheduling problems in semiconductor wafer fabrication. Journal of Scheduling 10 (1), 41–65.

Uzsoy, R., Martinvega, L.A., Lee, C.Y., 1991. Production scheduling algorithms for a semiconductor test facility. IEEE Transactions on Semiconductor Manufacturing 4, 270–280.

Wu, D., Ierapetritou, M., 2004. Cyclic short-term scheduling of multiproduct batch plants using continuous-time representation. Computers & Chemical Engineering 28, 2271.

Xiong, H.H., Zhou, M.C., 1998. Scheduling of semiconductor test facility via petri nets and hybrid heuristic search. IEEE Transactions on Semiconductor Manufacturing 11 (3), 384–393.

Zhang, X., Sargent, R.W.H., 1996. The optimal operation of mixed production facilities—a general formulation and some approaches for the solution. Computers & Chemical Engineering 20, 897–904.

Zhang, X., Sargent, R.W.H., 1998. The optimal operation of mixed production facilities—extensions and improvements. Computers & Chemical Engineering 20, 1287–1295.