Proceeding of the 2006 IEEE
Conference on Automation Science and Engineering
Shanghai, China, October 7-10, 2006

Token Delays and Generalized Workload Balancing for Timed Event Graphs with Application to Cluster Tool Operation

Tae-Eog Lee[1], Hwan-Yong Lee[2], and Ramavarapu S. Sreenivas[3]
1: (corresponding author) Korea Advanced Institute of Science and Technology (KAIST),
Department of Industrial Engineering
373-1 Gusung-dong, Yusung-gu, Daejon 305-701, Korea
telee@kaist.ac.kr
2: Samsung Electronics, LCD Business, Suwon, Korea
hwanyong.lee@samsung.com
3: University of Illinois at Urbana-Champaign,
Department of Industrial & Enterprise Systems Engineering
rsree@uiuc.edu

Abstract— **There have been numerous works on controlling task delays in discrete event systems such as automated manufacturing systems. Since the systems are often modelled by timed event graphs, the task delays correspond to token delays in the models. We characterize the token delays based on imbalance between the circuit ratios. We prove that the token delays can be eliminated by balancing the circuit ratios. We propose strategies for balancing circuit ratios or workloads for timed event graphs and apply them to eliminating wafer delays in a cluster tool. We demonstrate that token delay analysis based on circuit ratio imbalance and circuit ratio balancing are effective for identifying and eliminating wafer delays in a cluster tool. We discuss how circuit ratio and circuit ratio balancing can be considered as generalizations of conventional workload and workload balancing for flow lines.**

Index Terms— **timed event graph, token delay, workload balancing, cluster tool, wafer delay.**

I. Introduction

Automated or robotized manufacturing systems often have limited buffer spaces and material handling capacity. Therefore, the job processing modules and material handling systems such as robots are tightly coupled with each other in the sense that they restrict the tasks of each other and hence task delays or blockings frequently occur. Such task delays often cause cycle time increase or quality troubles. For instance, in a cluster tool for semiconductor manufacturing, a wafer should stay within a chamber after processing until it is unloaded by the robot. Excessive or non-uniform wafer delays cause critical quality problems due to residual gases and heat within the chambers [6]. Wet stations for cleaning wafers also have strict time constraints that the wafers cleaned at a chemical bath should be immediately rinsed at a water rinse bath [12]. There have been works on scheduling automated manufacturing systems with strict time constraints on task delays, including cluster tools and hoist-based production systems for printed circuit board fabrication, or timed Petri nets or event graphs with time window constraints on token sojourn times at places [2], [4], [5], [6], [7], [8], [11], [12], [14], [15], [16], [18], [21], [22], [24]. Kim et al. [6] identify conditions for which wafer delays in a dual-armed cluster tool do not exceed a specified upper limit. It has been known that task delays like wafer delays are caused by token delays at the corresponding places in the timed event graph model, which are the differences between the token sojourn times and the token holding times [4], [6], [7]. For timed event graphs that have time window constraints between the firing epochs of pairs of transitions, Lee and Park [7] develop a necessary and sufficient condition, based on circuits, for which there exists a firing schedule that satisfies the time constraints. These works suggest that token delays or time constraints are closely associated with token holding times and tokens of circuits. Although it is now possible to verify whether a given decision-free discrete event system satisfies time window constraints between pairs of events, we further wish to identify and reduce or eliminate any token delays or task delays. In semiconductor manufacturing, it is most desirable to reduce or eliminate unnecessary wafer delays completely, if possible, in order to achieve extreme quality targets. Other discrete event systems such as real-time embedded systems and asynchronous circuits also have similar time constraints.

In this paper, we characterize token delays of timed event graphs based on imbalance between the circuit ratios, the ratios of the sums of the token holding times to the numbers of tokens in the circuits, in a TEG. We prove that the token delays can be eliminated when the circuit ratios are all balanced. We apply these results to identifying and eliminating wafer delays in a cluster tool for semiconductor manufacturing. We discuss how circuit ratio and circuit ratio balancing can be considered as generalizations of conventional workload and workload balancing for flow lines. Finally, we propose strategies for balancing circuit ratios or workloads for timed event graphs and apply them to eliminating wafer delays in a cluster tool.

II. Token Delays in Timed Event Graphs

Petri nets have been used for modelling and analyzing operational behavior of discrete event dynamic systems. A Petri net consists of bars, circles, arrows, and dots that indicate transitions, places, arcs, and tokens, respectively. They usually model events or activities, activities or conditions, precedence relation between transitions and paces, and entities or parts, respectively [17]. An event graph or marked graph is a Petri net, of which each place has exactly one input and one output transition. For expositional convenience, we assume that only places have durations and all transitions are instantaneous transitions. It is known that a transition with a positive firing delay is equivalently transformed into a place with the token holding time the same as the firing delay and instantaneous transitions [1]. An event graph with token holding times at the places is called a *timed event graph* (TEG). A TEG is called *strongly connected* if for any two different transitions T_i and T_j there exists a path from T_i to T_j. A strongly connected TEG is live, that is, each transition keeps firing indefinitely, if and only if every circuit has a token. A firing schedule of a TEG is called *K-periodic* if $x_i^{r+K} = x_i^r + K\lambda \ \forall i \in \mathcal{T}, r = 1, 2, \ldots$, where x_i^r and λ indicate the r-th firing epoch of transition T_i and the average cycle time, respectively. For a circuit, the *circuit ratio* is the ratio of the sum of the token holding times to the number of tokens in the circuit. The maximum circuit ratio among all circuits is called the *critical circuit ratio* λ and a circuit with the ratio is called *critical*. It is also known that the minimum average cycle time among all feasible firing schedules of a TEG is the same as the critical circuit ratio of the TEG. Most TEGs assume the *earliest* firing policy that a transition fires as soon as it can fire. When the TEG is strongly connected, the average cycle time of an earliest starting schedule is the same as the critical circuit ratio λ. The earliest firing schedule is known to converge in a finite time to a K-periodic schedule with the average cycle time λ [1]. The periodicity K can be computed from the critical circuits and their numbers of tokens in the TEG [1]. Any TEG has also a 1-periodic earliest schedule with the cycle time λ by delaying the initial firings of the transitions appropriately, where each transition fires for each constant interval λ [1]. There are also non-earliest firing schedules with the minimum average cycle time λ, K-periodic or 1-periodic, that can have delays in transition firings. However, earliest starting schedules have advantages, including easy implementation by token play of the TEG without *a priori* computation and storage of the timing schedule and enforcement of logical correctness by keeping the precedence relations between the events through event-based timing control [10], [11], [12], [13], [25]. We therefore examine token delays in K-periodic and 1-periodic earliest starting schedules of a strongly connected live TEG. To do this, we first define token delays in a circuit and a path in a TEG.

Definition 1: (Token Delays)

1) The *token delay at a place* is the total sojourn time of a token at the place minus the token holding time at the place.
2) The *token delay on a path* from transition T_i to transition T_j is the sum of the token delays at the places in the path.
3) The *token delay on a circuit* is the sum of the token delays at its places during a cycle of the circuit.

Token Circulation Interpretation in TEGs: In a Petri net, the number of tokens in a circuit is not conserved. For a transition, the number of tokens that are generated in the output transitions is not necessarily the same as the number of tokens that are removed from the input transitions. Therefore, the movement of tokens in a Petri net cannot be interpreted as a network flow. However, in an event graph, the number of tokens in a circuit is constant for any marking [17]. A transition in a circuit removes one token from the input place in the circuit and adds one token into the output place in the circuit. Therefore, we can have an interpretation that the same tokens circulate in a circuit. Then, the token delay on a circuit or a path, which is defined as the sum of token delays at the places, not necessarily defined for a particular token, can be regarded as the total delay time at the places that each token undergoes during its traversal of the circuit or the path, that is, *traversal delay* of a token.

We characterize the value of the token delays. We let h_{ij} and τ_{ij} be the token holding time and the number of tokens of place P_{ij}, respectively. For a path ψ, h_ψ and τ_ψ represent the the sum of the token holding times and the the sum of the numbers of tokens of the places in the pat.

Theorem 1: Suppose that the TEG is strongly connected and live, and has the critical circuit ratio λ. Then, for any firing schedule with cycle time λ, the average token delay on a circuit π is

$$d_\pi = \tau_\pi \lambda - h_\pi. \tag{1}$$

Further, when the circuit π is critical, the token delay on the circuit is always 0.

Theorems 1 holds for any schedule with cycle time λ regardless of the periodicity and the firing or timing policy. For a 1-periodic schedule, the token delays do not vary and hence the token delays, not on average, can be identified.

Theorem 2: Suppose that a TEG is strongly connected and live. For an 1-periodic schedule $\{x_i^r | x_i^r = x_i + r\lambda \ \forall i \in \mathcal{T}, \forall r = 1, 2, \ldots\}$ with cycle time λ, the followings hold.

1) The token delay on any path $\psi(i,j)$ is always $d_{ij} = x_j - x_i + \tau_{\psi(i,j)}\lambda - h_{\psi(i,j)}$.
2) The token delay for a cycle along a circuit π is always the same as $d_\pi = \tau_\pi \lambda - h_\pi$.

We note that the above theorem does not require the 1-periodic schedule to be earliest. We observe that the token delay depends on the firing schedule.

Once the cycle time λ is known and the initial lags are given, an 1-periodic earliest schedule with cycle time λ can be computed from an associated graph. We define a directed graph $G(N,A)(\delta)$ such that node set is $N \equiv \{T_i | i \in \mathcal{N}\} \cup \{T_0\}$ and arc set is $A \equiv \{(T_i, T_j) | (i,j) \in \mathcal{P}\} \cup D$. Each arc (i,j) has length $h_{ij} - \tau_{ij}\lambda$. T_0 is a dummy start node. D is the set of dummy arcs from T_0 to a subset of nodes. For each critical circuit π in the TEG, select an arbitrary transition in the circuit T_i. Let C be the set of such selected nodes. Add each dummy arc (T_0, T_i) with length $\delta_i \ \forall i \in C$. δ_i is an initial lag of transition T_i. In other words, $\delta \equiv \{\delta_i\}$ defines the relative timing difference of the critical circuits, which determines a 1-periodic schedule. Since there is no token delay in a critical circuit, different choices of τ_π transitions for which initial lags are defined are all equivalent. We let $\gamma_{0i}(\delta)$ denote the longest path from T_0 to T_i in graph $G(N,A)(\delta)$.

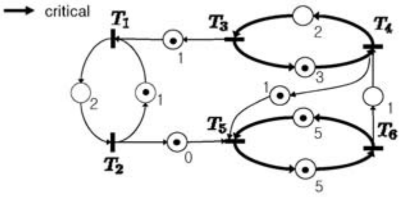

Fig. 1. Another TEG.

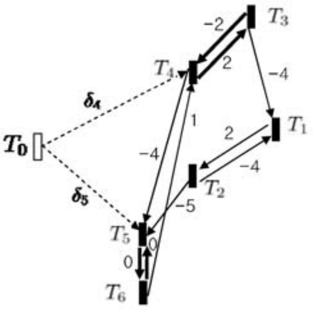

Fig. 2. Graph $G(N,A)(\delta)$.

Theorem 3: Suppose that a TEG is strongly connected and live. For an 1-periodic earliest schedule with cycle time λ with initial lag δ, $\{x_i^r | x_i^r = x_i + r\lambda \ \forall i \in \mathcal{T}, \forall r = 1, 2, \ldots\}$, the token delay on any path $\psi(i,j)$ is always

$$d_{\psi(i,j)}(\delta) = \tau_{\psi(i,j)}\lambda - h_{\psi(i,j)} + \gamma_{0j}(\delta) - \gamma_{0i}(\delta). \tag{2}$$

Example 1: **(Token delays in an 1-periodic earliest schedule)** We consider a more complicated TEG in Fig. 1. The number near a place indicates the token holding time at the place. The critical circuit ratio is 5. There are two critical circuits. Fig. 2 illustrates graph $G(N,A)(\delta)$ for the TEG in Fig. 1. We note that the topology of the graph is equivalent to the TEG. Suppose that the initial lags are given as $\delta_4 = 1$ and $\delta_0 = 0$. Then, by computing $\gamma_{0j}(\delta)$'s, we have an 1-periodic earliest schedule in Fig. 3. A number on a dotted arrow indicates the token delay between the two transitions. The solid tail of a dotted arrow corresponds to the token holding time. Consider two paths in the TEG, $\psi_1(3,5) = (T_3 \to T_1 \to T_2 \to T_5)$ and $\psi_2(3,5) = (T_3 \to T_4 \to T_5)$. Since $\gamma_{05}(\delta) = 0$ and $\gamma_{03}(\delta) = 3$, from Theorem 3, we have $d_{\psi_1(3,5)}(\delta) = 2 \times 5 - 3 + 0 - 3 = 4$ and $d_{\psi_2(3,5)}(\delta) = 2 \times 5 - 4 + 0 - 3 = 3$. We note that $h_{45} = 1$ and $h_{25} = 0$.

The token delays in a K-periodic schedule cyclically change while the average value is maintained. Therefore, even if the average token delay on a path is less than a specified limit, for instance, the maximum allowable wafer delay in a cluster tool, then the maximum token delay may exceed the limit. Further, even if the maximum token delay is within the limit, fluctuation of the

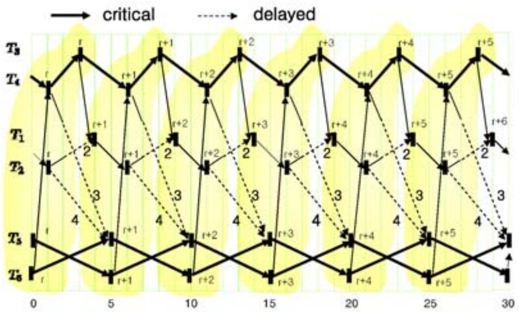

Fig. 3. An 1-periodic earliest schedule.

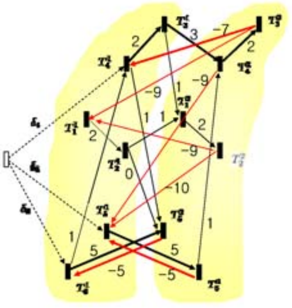

Fig. 4. Graph $G^K(N, A)(\delta)$.

token delays are often undesirable. For instance, non-uniformity in wafer delays degrades the quality or makes quality control or process engineering difficult. We now characterize the token delays in K-periodic schedules, not on average.

Theorem 4: Suppose that a TEG is strongly connected and live. For a K-periodic schedule with cycle time λ, $\{x_i^r | x_i^{r+K} = x_i^r + K\lambda \ \forall i \in \mathcal{T}, \forall r = 1, 2, \ldots\}$, the followings hold.

1) The token delays on a path $\psi(i,j)$ between a pair of transitions T_i and T_j repeat K values of

$$d^k_{\psi(i,j)} = \ x_j^{k+\tau_{\psi(i,j)} - \lfloor \frac{k+\tau_{\psi(i,j)}-1}{K} \rfloor K} - x_i^k + \lfloor \frac{k+\tau_{\psi(i,j)}-1}{K} \rfloor K\lambda - h_{\psi(i,j)} \ \forall k = 1, 2, \ldots, K. \quad (3)$$

2) The average token delay on a path $\psi(i,j)$ between a pair of transitions T_i and T_j is

$$\bar{d}_{\psi(i,j)} = \tau_{\psi(i,j)}\lambda - h_{\psi(i,j)} + \bar{l}_{ij}, \quad (4)$$

where $\bar{l}_{ij} = \frac{\sum_{k=1}^{K}(x_j^k - x_i^k)}{K}$.

We see that Theorems 2 and 4 reduce to Theorem 1 by taking $j = i$, where $l_{ij} = 0$ and $\bar{l}_{ij} = 0$. We note that Theorem 4 does not require the K-periodic schedule to be earliest starting either. From Theorems 2 and 4, the token delays on a path depend on the firing schedule.

We now show that a K-periodic earliest schedule and the token delays in the schedule can be computed from a directed graph, similarly as for a 1-periodic earliest schedule. Suppose that for the TEG, the critical circuit ratio λ, the cyclicity K, and the initial lags are given. From the proof of Theorem 4, we have the following precedence relations:

$$x_j^{k+\tau_{\psi(i,j)} - \lfloor \frac{k+\tau_{\psi(i,j)}-1}{K} \rfloor K} - x_i^k \geq h_{ij} - \lfloor \frac{k+\tau_{\psi(i,j)}-1}{K} \rfloor K\lambda \quad \forall (i,j) \in \mathcal{P}, \forall k = 1, 2, \ldots, K. \quad (5)$$

Similarly as for $G(N, A)(\delta)$, we define a directed graph $G^K(N, A)(\delta)$ from the precedence relations such that node set is $N \equiv \{T_i^k | i \in \mathcal{N}, k = 1, 2, \ldots, K\} \cup \{T_0\}$ and arc set is $A \equiv \{(i^k, j^{k+\tau_{ij} - \lfloor \frac{k+\tau_{\psi(i,j)}-1}{K} \rfloor K}) | (i,j) \in \mathcal{P}, k = 1, 2, \ldots, K\} \cup D$. Each arc $(i^k, j^{k+\tau_{ij} - \lfloor \frac{k+\tau_{\psi(i,j)}-1}{K} \rfloor K})$ has length $(h_{\psi(i,j)} - \lfloor \frac{k+\tau_{ij}-1}{K} \rfloor K\lambda)$. T_0 is a dummy start node. D is the set of dummy arcs from T_0 to a subset of nodes. It is selected as follows. For each critical circuit π in the TEG, select arbitrary τ_π transitions in the circuit, $T_i, i = 1, 2, \ldots, \tau_\pi$. C is the set of such nodes for all critical circuits. Then, D is the set of arcs from T_0 to each of such node with length δ_i. As seen in Fig. 5, when a critical circuit π has τ_π tokens, τ_π work cycles of the circuit are concurrently performed. Therefore, each of them has a separate initial lag. Since there is no token delay in a critical circuit, any different choices of τ_π transitions for which initial lags are defined are all equivalent. We assume that the initial lag value $\delta \equiv \{\delta_i\}$ is given. Let $\gamma_{0j^k}(\delta)$ denote the longest path length from T_0 to a node $T_j^k \in N$. Fig. 4 illustrates the graph for the TEG in Fig. 1.

Theorem 5: Suppose that a TEG is strongly connected and live. For a K-periodic earliest schedule with initial lag δ, the token delays on a path $\psi(i,j)$ have K different values:

$$d^k_{\psi(i,j)}(\delta) = \gamma_{0j^{k+\tau_{ij} - \lfloor \frac{k+\tau_{\psi(i,j)}-1}{K} \rfloor K}}(\delta) - \gamma_{0i^k}(\delta) + \lfloor \frac{k+\tau_{\psi(i,j)}-1}{K} \rfloor K\lambda - h_{\psi(i,j)} \ \forall k = 1, 2, \ldots, K. \quad (6)$$

Example 2: Consider Example 1 again. Suppose that $\delta_4 = 3, \delta_4 = 2$, and $\delta_6 = 0$. From graph $G^K(N, A)(\delta)$ in Fig. 4, we compute $\gamma_{0j^k}(\delta)$'s and determines a 1-periodic earliest schedule, as shown in Fig. 5. For the path $\psi_1(3,5)$ in Example 1, we have $\gamma_{03^1} = 5$, $\gamma_{05^1}(\delta) = 2$, $\gamma_{05^2}(\delta) = 5$, and $\gamma_{03^2} = 10$. Since $\lambda = 5$, $K = 2$, $\tau_{\psi_1(3,5)} = 2$, and $h_{\psi_1(3,5)} = 3$, from equation (6), we have $d^1_{35}(\delta) = \gamma_{05^{(1+2) - \lfloor \frac{1+2-1}{2} \rfloor 2}}(\delta) - \gamma_{03^1}(\delta) + \lfloor \frac{1+2-1}{2} \rfloor 2 \times 5 - 3 = 2 - 5 + 10 - 3 = 4$. $d^2_{35}(\delta) = \gamma_{05^{(2+2) - \lfloor \frac{2+2-1}{2} \rfloor 2}}(\delta) - \gamma_{03^2}(\delta) + \lfloor \frac{2+2-1}{2} \rfloor 2 \times 5 - 3 = 5 - 10 + 10 - 3 = 2$. We observe that the two token delays 2 and 4 alternatively occur all on place P_{25} in the 1-periodic earliest schedule.

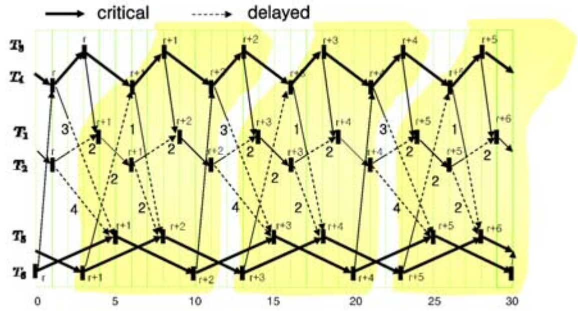

Fig. 5. K-periodic earliest schedule for another TEG.

Generalized Workload and Workload Balancing

Workload measure and workload balancing have been widely used for conventional flow lines, where job transfer between stations are assumed not to be restricted. The workload for a station is defined to be the sum of process times of the jobs that are processed at the station. The cycle time is determined by the workload at the bottleneck station, regardless of the job processing sequence. Imbalance between the workloads of the stations causes waiting jobs at the upstream stations prior to the bottleneck and starvation or idle times at the downstream stations, and hence reduces the overall resource utilization. Therefore, workload balancing, by reducing the bottleneck workload or increasing or compensating the workloads at the non-bottleneck stations, has been essential for reducing the cycle time and the work-in-progress or queues. Many job flow control methods for controlling work-in-progress inventory, including kanban systems, are basically intended for compensating the workload imbalance.

On the other hand, there have been attempts to define workload measures for manufacturing systems with automated material transfer such as cluster tools that is modelled by a TEG [6], [9], [11]. The proposed workload measures for a process step, of which work cycle consists of wafer processing and a set of robot tasks, include the times for the required robot tasks as well as the process time. We observe that the workload measures coincide with the circuit ratio and essential for characterizing the wafer delays. We therefore propose that the workload for a work cycle of a resource in an automated system is defined to be the circuit ratio of the corresponding circuit in the TEG model. Circuit ratio and circuit ratio balancing are then called *generalized workload* and *generalized workload balancing*, respectively. In the next sections, we present application of those concepts for characterizing and eliminating wafer delays of a cluster tool.

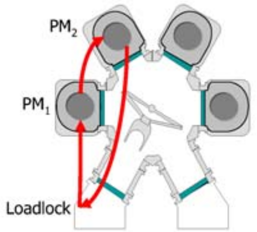

Fig. 6. Single-armed cluster tool with four processing chambers.

III. Wafer Delays in a Single-Armed Cluster Tool

We now analyze wafer delays for a cluster tool based on token delays in the corresponding TEG model. Wafers undergo a series of process steps by visiting the processing chambers, also called processing modules (PMs), in series. A process step may have parallel identical PMs. Therefore, wafer flow patterns are series-parallel in general. Fig. 6 illustrates a single-armed cluster tool [9]. Each wafer is unloaded from a loadlock, undergoes the process steps at the PMs, and returns into the loadklock. Fig. 7 shows a TEG model for a single-armed cluster tool. Details can be found in [9].

Each process step is associated two circuits. For instance, process step 1 are performed by circuit $\pi_1^1 \equiv (T_3 \to T_4 \to T_5 \to T_6 \to T_{13} \to T_3)$ and circuit $\pi_2^1 \equiv (T_1 \to T_2 \to T_3 \to T_4 \to T_5 \to T_6 \to T_7 \to T_8 \to T_1)$. Circuit π_1^1 has always $m_1 = 2$ tokens as many as the number of parallel PMs for the process step. It can be easily shown that circuit π_2^i also has the same number of tokens. The circuit ratios of π_1^i and π_i^2 are $\frac{p_1+l+u}{m_1}$ and $\frac{p_1+2u+2l+3v}{m_1}$, respectively, where intentional delay at place $D_{6,13}$ is excluded. Since the second ratio is larger, the work cycles of the process step is dominated by the second circuit π_2^1. The circuit is called a *processing circuit* for the process step. Generally, for process step i, the circuit ratio of the processing circuit is $\frac{p_i+2u+2l+3v}{m_i}$ [9]. The robot repeats a work cycle that passes places for robot tasks and their associated transitions, which also defines a circuit called a *robot circuit*. Since the robot circuit has a single token, the circuit ratio is $(n+1)(u+l+2v)$ [9]. The workload for a process step or a robot in a cluster tool has been defined as the work that is essentially required for the corresponding resource to complete a work cycle [6], [9], [11], [25]. These definitions also extend the conventional workload for flow lines by counting material handling tasks into the workload. The workload definitions are exactly the same as the circuit ratios of the corresponding circuits in the TEG model. The circuit ratio of each circuit would be the cycle time of the circuit or the corresponding work cycle if the circuit or work cycle

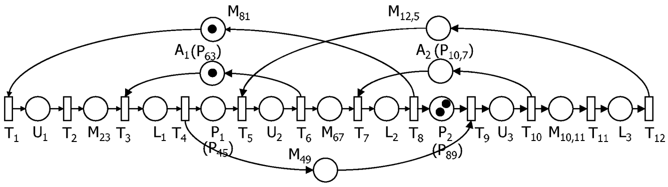

Fig. 7. TEG model for a single-armed cluster tool.

is not interfered by other circuit or work cycles.

Theorem 6: For a single-armed cluster tool with the backward sequence, the followings hold.

1) The average wafer delay at process step i of an earliest starting K-periodic schedule is the same as $m_i \max\{\max_{k=1,2,\ldots,n} \frac{p_k+2l+2u+3v}{m_k}, (n+1)(u+l+2v)\} - (p_i+2l+2u+3v)$ regardless of the periodicity.
2) Furthermore, the wafer delays at each process step are constant for an 1-periodic schedule.
3) Finally, they are all zero regardless of the periodicity when all work cycles in the tool are completely balanced, that is, $\frac{p_k+2l+2u+3v}{m_k} = (n+1)(u+l+2v)\ \forall k = 1,2,\ldots,n$.

We now show that the token delays can be completely prevented when the TEG satisfies some condition.

Theorem 7: If all circuits of a strongly connected live TEG have the same circuit ratio value, an earliest firing schedule of a TEG has no token delay at any place regardless of the periodicity.

Proof: An earliest starting schedule is K-periodic or 1-periodic [1]. However, by Theorem 1, the average token delay in a critical circuit is zero. Since all circuits are critical, the token delays in every circuit are zero. Any place belongs to a critical circuit because the TEG is strongly connected. Therefore, the average token delays at any place, path, and circuit are all zero. Since the token delay cannot be negative, all token delays in even in a K-periodic schedule should be zero. ■

Strategies for Generalized Workload Balancing

We now discuss how workloads or circuit ratios should be balanced. A circuit ratio can be changed by adjusting the token holding times and the number of tokens in the circuit. Modifications of the net topology, including adding or deleting a path and adding a place change, the circuits and circuit ratios. The feasible adjustment or modifications in the TEG model depend on the application domain. For instance, in a cluster tool model, the token holding times correspond to the process times or the robot task times. Although the process times are determined by process engineering, they often can be adjusted within some technical ranges [6]. We may add parallel PMs to a bottleneck process step, which increases the number of tokens in the processing circuits. We may add a place between two transitions with a token holding time as long as the desired delay in order to intentionally delay the firing of a transition. We may add a circuit that regulates the time interval, called *wafer delivery interval (WDI)*, for which a new wafer is unloaded into the tool [5]. We note that popular kanban systems for regulating the work-in-progress inventory are also essentially to add intentional delays to the upstream operations through added signal feedback circuits. The robot task sequence also can be changed. This causes the robot circuit to be changed [11], [12]. In general TEGs, many circuits are interwoven. It is often computationally prohibitive to identify all circuits and their relationships. Therefore, we can develop a mixed integer program for balancing the circuit ratios.

IV. Conclusion

We characterized the average token delays for a TEG based on the circuit ratio imbalance. We showed that the token delays are constant for an 1-periodic schedule, and cyclically changes for a K-periodic schedule. We also proved that the token delays disappear when the circuit ratios are all identical. We discussed that the circuit ratio measure and circuit ratio balancing are generalizations of the conventional workload measure and workload balancing. We applied the token delay analysis to estimate wafer delays and the generalized workload balancing to eliminate the wafer delays in a single-armed cluster tool. Our token delay analysis and generalized workload balancing are simple but effective for scheduling time-constrained decision-free discrete event systems, which have been addressed by many studies.

We may extend token analysis and generalized workload balancing approach to more general Petri nets or TEGs with random variations or disruptions in the holding times.

References

[1] F. L. Baccelli, G. Cohen, G. J. Olsder, and J. P. Quadrat, *Synchronization and Linearity - An Algebra for Discrete Event Systems*. New Tork, NY:John Wiley, 1992.

[2] H. Chen, C. Chu, and J.-M. Proth, "Cyclic scheduling of a hoist with time window constraints," *IEEE Transacion on Robotics and Automation*, vol. 14, no. 1, pp. 144–152, 1998.

[3] P. Chretienne, "The basic scheduling problem with deadlines," *Discrete Applied Mathematics*, vol. 30, no. 2/3, pp. 109–123, 1991.

[4] W. Khansa, P. Aygaling, and J. P. Denat, "Structural analysis of p-time petri nets," in *Proceedings of the CESA 1996 Imacs Multiconference IEEE-SMC*, (Lille, France), pp. 127–136, July 1996.

[5] J. H. Kim and T. E. Lee, "Schedule stabilization and robust timing control for time-constrained cluster tools," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1039–1044, 2003.

[6] J. H. Kim, T. E. Lee, H. Y. Lee, and D. B. Park, "Scheduling analysis of time-constrained dual-armed cluster tools," *IEEE Transactions on Semiconductor Manufacturing*, vol. 16, no. 3, pp. 521–534, 2003.

[7] T. E. Lee and S.-H. Park, "An extended event graph with negative places and tokens for time window constraints," *IEEE Transactions on Automation Science and Engineering,* vol. 2, no. 4, pp. 319–332, 2005.

[8] T. E. Lee and S. H. Park, "Steady state analysis of a TEG with time window constraints," *Proceedings of IEEE International Conference on Automation Science and Engineering (CASE),* Edmonton, Canada, pp. 1–6, 2005.

[9] T. E. Lee, H. Y. Lee, and Y. H. Shin, "Workload balancing and scheduling of a single-amred cluster tool," in *Proceedings of the 5th APIEMS Conference, Gold coast, Australia*, pp. 1–15, 2004.

[10] T. E. Lee, "Stable earliest schedule for cyclic job shops: a linear system approach," *The international Journal of Flexible Manufacturing Systems*, vol. 12, no. 1, pp. 59–82, 2000.

[11] H. Y. Lee and T. E. Lee, "Scheduling single-armed cluster tools with reentrant wafer flows," *IEEE Transactions on Semiconductor Manufacturing,* 2006 (*to appear*).

[12] T. E. Lee, H. Y. Lee, and S. J. Lee, "Scheduling a wet station for wafer cleaning with multiple job flows and multiple wafer-handling robots," *International Journal of Production Research,* 2006 (*to appear*). (A restriced version appeared at Proceedings of Conference on Modeling and Analysis of Semiconductor Manufacturing, Singarpore, 2005 (MASM 2005).)

[13] T. E. Lee and M. E. Posner, "Performance measures and schedules in periodic job shops," *Operations Research,* vol. 45, no. 1, pp. 72–91, 1997.

[14] L. Lei and T. J. Wang, "Determining optimal cyclic hoist schedules in a single-hoist electorplating line," *IIE Transactions*, vol. 26, no. 2, pp. 25–33, 1994.

[15] J. Liu, Y. Jiang, and Z. Zhou, "Cyclic scheduling of a single hoist in extended electroplating lines: a comprehensive integer programming solution," *IIE Transactions*, vol. 34, pp. 905–914, 2002.

[16] M. A. Manier, C. Varnier, and P. Baptiste, "Constraint-based model for the cyclic multi-hoists scheduling problem," *Production Planning and Control*, vol. 11, no. 3, pp. 244–257, 2000.

[17] T. Murata, "Petri nets:properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[18] W. C. Ng, "A branch and bound algorithm for hoist scheduling of a circuit board production line," *The International journal of Flexible Manufacturing Systems*, vol. 8, pp. 45–65, 1996.

[19] K. Onaga, M. Silva, and T. Watanabe, "On periodic schedules for deterministically timed petri net systems," in *Proceedings of the Fourth International Workshop on Petri Nets and Performance Models*, pp. 210–215, 1991.

[20] J. H. Paek and T. E. Lee, "Scheduling a dual-armed cluster tool with intermediate buffers", Proceedings of International Conference on Industrial Engineering, Busan, Korea, 2002.

[21] L. W. Phillips and P. S. Unger, "Mathematical programming solution of a hoist scheduling program," *AIIE Transactions*, vol. 8, no. 2, pp. 219–225, 1976.

[22] S. Rostami, B. Hamidzadeh, and D. Camporese, "An optimal periodic scheduler for dual-arm robots in cluster tools with residency constraints," *IEEE Trans. Robot. Automat.*, vol. 17, pp. 609–618, 2001.

[23] J. W. Seo and T. E. Lee, "Steady state analysis of cyclic job shops with overtaking," *International Journal of Flexsible Manufacturing Systems*, vol. 14, no. 4, pp. 291–318, 2002.

[24] G. W. Shapiro and H. L. W. Nuttle, "Hoist scheduling for a pcb electroplating facility," *IIE Transactions*, vol. 20, no. 2, pp. 157–167, 1988.

[25] Y. H. Shin, T. E. Lee, J. H. Lee, and H. Y. Lee, "Modeling and implementing a real-time scheduler for dual-armed cluster tool", *Computers in Industry*, vol. 45, pp. 13–27, 2001.

[26] J. Wang, *Timed Petri Nets.* Kluwer Academic Publishers, Boston, 1998.