Fast Schedule Recovery Method for Semiconductor Packaging Lines with Machine Failures and Constrained Time Windows

Feifan Wang, *Member, IEEE,*, Yutong Su, Feng Ju, *Member, IEEE,* A. Bala Krishnan, Husam Dauod, and Nital S. Patel, *Senior Member, IEEE*

***Abstract*—In semiconductor packaging line, a master schedule is normally created every shift to optimize production for a three-week horizon. However, semiconductor packaging lines are susceptible to machine failures, causing the master schedule to be sub-optimal or even infeasible. Specifically, machine failures can result in due dates not to be met and time window constraints to be violated. In this study, we classify machine failures in semiconductor packaging lines into two categories: short and long machine failures, which can be identified when the failure happens. To handle short machine failures, extra time is added into the processing time of each lot to make the master schedule robust. When a long machine failure occurs, a mixed integer programming model is formulated to adjust the master schedule. The master schedule is taken as a warm start, and a short period schedule is obtained using CPLEX for the semiconductor packaging line to follow immediately. In this way, the semiconductor packaging line can quickly respond to long machine failure without replacing the whole master schedule or giving the master scheduler enough time to remake a new master schedule. Thus, the negative impact of machine failure is mitigated. Using the data from shop floor, a simulation model is developed with SimPy to simulate a real-world semiconductor packaging line and evaluate the proposed method. The experiment results show that the proposed method can achieve fast response to machine failures in semiconductor packaging lines.**

***Note to Practitioners*—The semiconductor packaging line operates as a highly intricate production system. Its inherent flexibility allows for the simultaneous processing of diverse products from various orders. However, this flexibility necessitates adherence to a master schedule, often created without accounting for potential disruptions, such as machine failures. When not promptly addressed with efficient production control measures, machine failures can lead to production losses and product quality concerns. Moisture absorption and die surface oxidation are common quality issues that arise in the semiconductor packaging line, primarily caused by extended wait times in buffers. To equip production engineers with an effective solution for real production settings, this study introduces a practical tool to swiftly respond to machine failures. By implementing this tool, production teams can mitigate the impact of disruptions and ensure smoother operations in the semiconductor packaging process.**

***Index Terms*—semiconductor packaging line, real-time, machine failure, residence time constraints.**

I. Introduction

Semiconductor manufacturing involves two key stages: wafer fabrication and packaging, commonly known as front-end and back-end processes, respectively [1], [2], [3]. The semiconductor packaging line operates as a flexible manufacturing system, capable of processing products of different types and orders concurrently. This adaptability, while advantageous, adds complexity to production system analysis and control [4], [5]. As a result, extensive research has focused on developing efficient scheduling techniques for the semiconductor packaging line, aiming to maximize throughput, meet order deadlines, and fulfill production requirements [6], [7], [8].

In semiconductor manufacturing companies, the core of the production efficiency lies in the master schedule, which is meticulously created during each shift, detailing the processing timeline and locations for each lot of products over the next three weeks. The master schedule plays a vital role in facilitating efficient production, optimizing resource allocation, enabling effective coordination, and enhancing adaptability to uncertainties [9]. However, crafting such a master schedule, while considering both feasibility and optimality, is a time-consuming process that typically demands several hours of run time.

In addition, the semiconductor packaging line may not always run as planned, and it can easily be disrupted by machine failures. Machine repair takes from several minutes to hours, causing the master schedule to be sub-optimal or even infeasible. Specifically, machine failures can result in delay. Since those lots directly impacted by machine failures may not arrive at downstream operations in time, it further leads to delay at downstream operations. In addition, products in a semiconductor packaging line are subject to residence time constraints. Due to oxidation and moisture absorption issues, they cannot stay in buffer too long [10]. Residence time constraints are considered in a master schedule to guarantee product quality, but the delay caused by machine failures may lead to violation of residence time constraints. When a long machine failure occurs, sticking to the original master schedule or immediately remaking a new master schedule may not be

Feifan Wang is currently with the Department of Industrial Engineering, Tsinghua University, Beijing China. He was previousely with the School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281 USA (e-mail: wangfeifan@tsinghua.edu.cn).

Yutong Su and Feng Ju are with the School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ 85281 USA (e-mail: yutongs1@asu.edu, fengju@asu.edu).

A. Bala Krishnan is with Intel Technology India Pvt. Ltd., Bengaluru, Karnataka 560037 India (e-mail: a.bala.krishnan@intel.com).

Husam Dauod, and Nital S. Patel are with the Intel Corporation, Chandler, AZ 85226 USA (e-mail: husam.dauod@intel.com, nital.s.patel@intel.com).

This work is supported in part by Intel Grant 579395195 and National Science Foundation Grant CMMI-1922739.

effective strategies to deal with it. These approaches can be time-consuming and may lead to violations of residence time constraints.

This study focuses on the timely response to machine failures in semiconductor packaging and testing lines, considering two practical requirements. First, there is a strong preference for retaining the master schedule due to the challenges of remaking it. Second, the computation time needs to be sufficiently small. In this study, we classify machine failures in semiconductor packaging lines into two categories: short and long machine failures, which can be identified by the semiconductor packaging line system when the failures happen. For short machine failures lasting less than one hour, we incorporate additional time to each operation of each lot in the master schedule to enhance its robustness. This allows the semiconductor packaging line to recover autonomously without any intervention. To address long machine failures, which usually last several hours, a mixed integer programming model is developed to adjust the master schedule. To enhance the computational efficiency, two strategies are applied. First, the mixed-integer programming model is built for one operation and subsequently applied on operations according to their packing line. Second, the master schedule is utilized as a warm start, generating with CPLEX Optimizer for the semiconductor packaging line to follow. In this way, the semiconductor packaging line can quickly respond to long machine failures without replacing the whole master schedule or giving the master scheduler enough time to remake a new master schedule. Thus, the negative impact of machine failure is minimized. Data from the shop floor are collected. Using those data, a simulation model is developed with Python and Simpy package to simulate a real-world semiconductor packaging line and evaluate the proposed method. The experiment results show that the proposed method can achieve a fast response to machine failures in semiconductor packaging lines.

The remainder of the paper is organized as follows. Section II reviews the literature. The semiconductor packaging line under study is described in Section III. The method to respond to machine failures is proposed in Section IV. Simulation experiment is used to evaluate the proposed method in Section V. Relevant issues are discussed in Section VI. Finally, Section VII concludes the study.

II. Literature review

Production systems are susceptible to uncertain disruptions, such as machine failures. For production systems that carry out production without using an explicit schedule, such as assemble-to-order systems [11], [12], serial lines [13], [14], [15] and deteriorating manufacturing systems [16], [17], [18], real-time production control, reactive scheduling and online scheduling are common ways to address uncertainty [19], [20]. The control policy can be created beforehand, and in the run time, one may control production by following the predetermined rule. When a production system is complex and flexible, a schedule is often required. One may make a deterministic but robust schedule to deal with minor uncertainty independently without intervention[21]; however, if the robust schedule can not reduce the negative impact of the uncertainty, the intervention is needed. One may use rescheduling methods to partially adjust the original schedule with small scale, including right shift rescheduling, single machine oriented match-up rescheduling, machine group oriented match-up rescheduling, affected operation rescheduling, and fix-sequence rescheduling[22], [19], [23], [24].. If a scheduling algorithm has computation time short enough, the same algorithm can also act as a rescheduling method and quickly create a new schedule in response to disruptions[6], [25]. The semiconductor packaging line in this study is complicate, and a schedule is required. We cannot merely rely on a robust schedule when long machine failure occurs. Considering the long computation time to make a master schedule, total rescheduling is not applicable to this problem. A proper way to control production quickly to machine failures in such a complex semiconductor packaging line has not been fully studied.

Another motivation for fast response to machine failures in semiconductor packaging lines is residence time constraints. Production systems with residence time constraints are evaluated and controlled, where products are perishable and become defective after staying in a buffer for too long [26], [27], [28]. Real-time control on machines' working mode can maintain high production and small scrap rates [29], [30], [31], [32], [33], [34]. In semiconductor manufacturing, both fabrication and packaging are restricted by residence time constraints. In wafer fabrication, a wafer is processed at a process module and required to be unloaded within a time limit [35], [36]. Different types of time limits are also observed in other process steps of wafer fabrication [37], [38]. Time windows restrict products in a semiconductor packaging line due to the concerns of oxidation and moisture absorption [10]. Failures caused by moisture include popcorn cracking [39], deformation [40] and adhesion degradation [41]. However, not much attention has been paid to production control of semiconductor packaging lines considering residence time constraints.

A semiconductor packaging line is often formulated and solved as a flexible job shop problem, which is a job shop problem with parallel machines and thus an NP-hard problem [42]. Practically, a semiconductor packaging line has more requirements, making the scheduling more complex not to mention production control in real time. Process steps of semiconductor packaging commonly include wafer mount, wafer sawing, die attach, wire bond, and inspection [3]. Reentrance is involved, and set up on a machine is required depending on product type and may take hours [6], [43], [25]. The number of setup operators could also be limited [6]. In addition, internal and external variability, such as unscheduled machine downtime and rush orders, is commonly seen [6]. Meta-heuristic methods are often used to create production schedule [7], [8]. However, given a schedule, how to respond to disruption is worth more research.

III. System description

A. Layout

In this paper, we study a segment of a semiconductor packaging line, presented in Fig. 1. There are four operations,

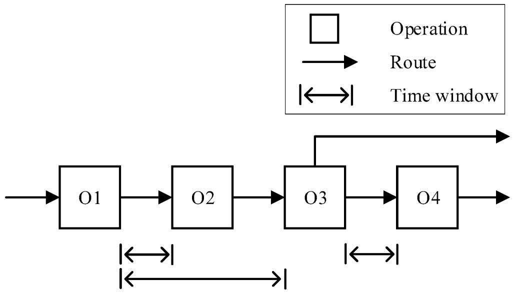


Fig. 1. The layout of the segment of a semiconductor packaging line under study

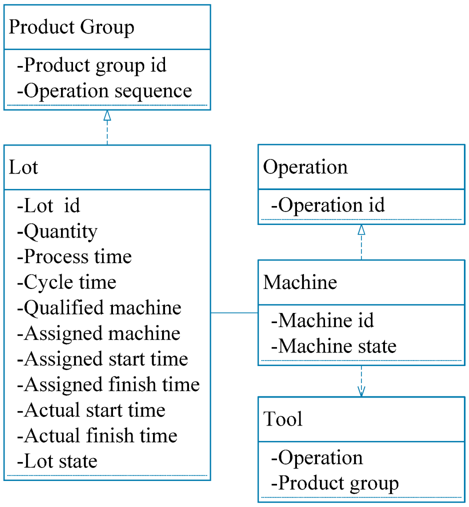


Fig. 2. The properties and relationship of product group, lot, operation, machine and tool

denoted by O1, O2, O3, and O4. Products of different types may have different requirements in terms of operations. Some products go over all four operations, and the others only need to visit the first three operations. Products are subject to residence time constraints, defined by time windows. A time window specifies the maximum time a product can stay in a certain area containing one or more than one buffer.

B. Product, product group and lot

Each product belongs to a product group. Products of the same product group share great similarities and require the same operations. The properties of the product group include product group id and operation sequence, shown in Fig. 2. In the semiconductor packaging line under study, there are 20 different product groups.

A lot consists of hundreds of products belonging to the same product group. Each lot is identified by a unique ID, such as L1, L2, etc. Lots serve as the minimum unit flowing within the production system. Each lot carries properties presented in Fig. 2. Different lots may carry various quantities of products, leading to different process time. A set of qualified machines is assigned to process each lot. Even the lots with the same product group may have different sets of qualified machines. A lot is assigned to one machine for each operation, chosen from the set of qualified machines. When a lot finishes the process at an operation, there are some supplementary works to do on the lot. It also takes time to move the lot to the next operation. The inter-operation time refers to the length of the lot finishing an operation to the lot becoming available in the buffer for the next operation. Thus, at any time, a lot can be waiting in a buffer, getting processed on a machine, or spending inter-operation time. A predetermined schedule specifies the assigned start time and assigned finish time of each operation for each lot. If the actual finish time of a lot is later than the assigned finish time, then the lot is delayed.

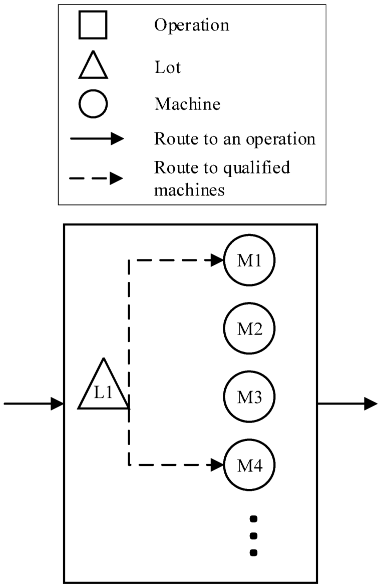
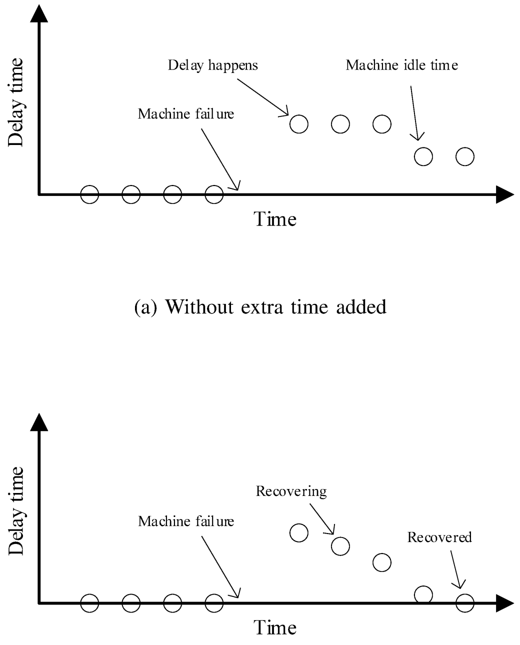


Fig. 3. The layout of an operation

C. Operation, machine and tool

An operation consists of a group of machines, each identified by a unique identifier such as M1, M2, etc. Each machine can process lots one at a time. When a machine becomes available, it selects a lot from its buffer for processing according to the predetermined master schedule if there is no machine failure occurs. To process a lot, a machine needs to have a tool installed. The tool type should match the product group of the lot. If the tool and product group are compatible, the machine can proceed with the processing without any additional setup. However, if the incoming lot belongs to a different product group that requires a different tool, a tool conversion time is needed.

Fig. 3 shows the layout of an operation that consists of several machines. For a lot at this operation, there is a set of qualified machines, which is a subset of all machines at the operation. The lot chooses one qualified machine according to the schedule to have the operation done.



(a) Without extra time added

(b) With extra time added

Fig. 4. Illustration of short machine failure

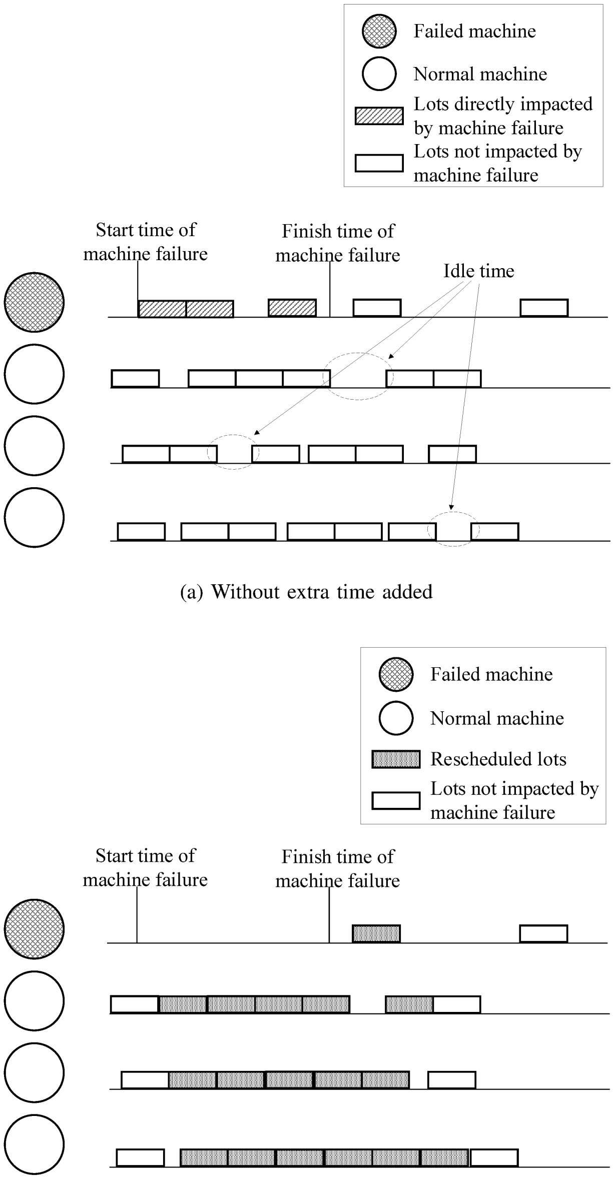
D. Schedule and disruption

A schedule is created every shift, specifying the releasing time, processing time, and assigned machine of a lot. Making a feasible and satisfactory schedule is complex, and it usually takes a long time. The production of the following three weeks is then carried out according to the schedule updated every shift. However, the production can be disrupted by machine failures. When a machine experiences a failure and is unable to operate, the lots assigned to that particular machine are forced to wait until the machine is repaired. All the lots assigned to the machine within the failure time will experience delays in their processing. Additionally, the occurrence of machine failures may lead to violations of residence time constraints.

IV. Method

A. Short machine failure

We classify machine failures into two categories, short machine failure and, long machine failure, and deal with them separately. To address short machine failure, which usually lasts less than one hour, a small amount of extra time is allocated to the processing time for each operation within every lot in the master schedule according to the historical short failure time recorded for the machine. This allocation results in machines having intermittent idle time between two consecutive lots. Thus, the semiconductor packaging line can recover soon on its own after the machine is repaired. Fig. 4 presents a situation where a machine fails to work. Each circle represents a lot assigned to the machine experiencing short machine failure. The horizontal axis is when a lot finishes the operation, and the vertical axis stands for the delay time. Machine failure occurs after the fourth lot is processed. Without extra time added, the delay time of lots lasts long and reduces only when there exists machine idle time, shown in Fig. 4a. In contrast, Fig. 4b shows the case with extra time added, and it starts recovering when the machine is repaired. The delay time of lots decreases until it finally reaches zero.


Failed machine
Normal machine
Lots directly impacted by machine failure
Lots not impacted by machine failure
Start time of machine failure
Finish time of machine failure
Idle time
Failed machine
Normal machine
Rescheduled lots
Lots not impacted by machine failure
Start time of machine failure
Finish time of machine failure

(a) Without extra time added

(b) With extra time added

Fig. 5. Illustration of adjustment for an operation

B. Long machine failure

Even with extra time added to each operation for each lot, a schedule cannot be robust enough to handle long machine failures, and a proper adjustment is required. The computation time should be short enough so that the factory floor can soon switch to the adjusted schedule without too much delay. The adjustment can allow the semiconductor packaging line to recover faster. Fig. 5 illustrates the adjustment for an operation.

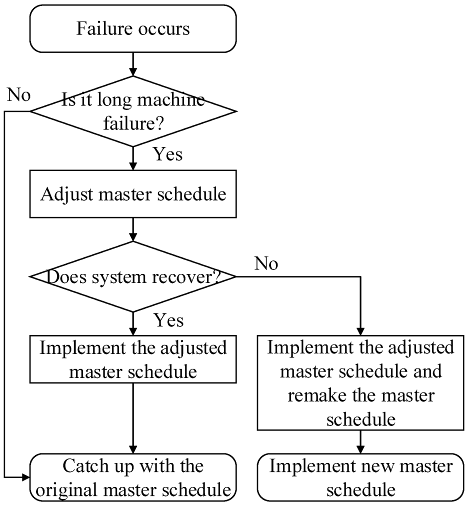

Failure occurs
Is it long machine failure?
No
Yes
Adjust master schedule
Does system recover?
No
Yes
Implement the adjusted master schedule
Implement the adjusted master schedule and remake the master schedule
Catch up with the original master schedule
Implement new master schedule

Fig. 6. The flowchart to handle machine failure

When a machine fails, some lots are directly impacted, shown in Fig. 5a. Without proper control, those lots and the following lots assigned to the same machine can have a long delay. The control takes advantage of machine idle time in the master schedule so that the total delay time is minimized and the residence time constraints are satisfied. After a short period, the production can catch up with the master schedule, shown in Fig. 5b. If the semiconductor packaging line cannot recover via the control, the quickly adjusted schedule gives the master scheduler time to remake a new master schedule.

Fig. 5 only shows the control for the operation with failed machine. The adjusted schedule also impacts downstream operations. Some lots may delay at one operation and not be able to start the next operation in time. Therefore, similar adjustment is carried out for the downstream operations.

Fig. 6 shows how a machine failure is handled. If it is a short machine failure, the system can recover on its own. If it is a long machine failure, the master schedule is adjusted so the system can catch up with the original master schedule or has enough time to remake a new master schedule.

1) Model of an operation: Assume a machine fails, and the repair is expected to be long. The decision horizon is determined and should be longer than the repair time. Let I be the total number of lots that are or will be operating within the decision horizon. Let K be the total number of machines at the operation. We first introduce the parameters of the model. $o_{i,k} \in \mathbb{R}_{\geq 0}$, for $i = 1, 2, \cdots, I$ and $k = 1, 2, \cdots, K$, specifies if the kth machine is a qualified machine for the ith lot and how long it takes to process. If $o_{i,k} = 0$, the kth machine is not qualified for the ith lot. If $o_{i,k} > 0$, the kth machine is qualified and the process time is $o_{i,k}$. Let e_i, for $i = 1, 2, \cdots, I$, be the time when the ith lot enters the buffer and is ready for the operation. Denote by α_i, for $i = 1, 2, \cdots, I$, the product group of the ith lot. $s_{k,\alpha_i,\alpha_{i'}}$ denotes the conversion time of the kth machine from product group α_i to product group $\alpha_{i'}$. Let t_k^{r}, for $k = 1, 2, \cdots, K$, be the time when the kth machine becomes available for the first time in the decision horizon. If the kth machine is the failed machine, then t_k^{r} is the time when the machine is repaired. Let p_i, for $i = 1, 2, \cdots, I$, be the time when the ith lot is scheduled to finish the process according to the original master schedule.

$$\min \quad f\left(x_{i,k}, z_{i,i'}, a_{i,k}, d_i\right) = \sum_{i=1}^{I} d_i, \tag{1}$$

s.t.

$$\sum_{k=1}^{K} a_{i,k} = 1, \qquad \text{for } i = 1, 2, \cdots, I, \tag{2}$$

$$x_{i,k} \leq M a_{i,k}, \qquad \text{for } i = 1, 2, \cdots, I, \text{ and } k = 1, 2, \cdots, K, \tag{3}$$

$$a_{i,k} \leq o_{i,k}, \qquad \text{for } i = 1, 2, \cdots, I, \text{ and } k = 1, 2, \cdots, K, \tag{4}$$

$$e_i \leq \sum_{k=1}^{K} x_{i,k}, \qquad \text{for } i = 1, 2, \cdots, I, \tag{5}$$

$$x_{i',k} - x_{i,k} \leq M\left(1 - z_{i,i'}\right) - 1 + 2M(1 - a_{i',k}) + 2M(1 - a_{i,k}),$$
$$\text{for } i = 1, 2, \cdots, I, \; i' = 1, 2, \cdots, I, \; i \neq i', \text{ and } k = 1, 2, \cdots, K, \tag{6}$$

$$x_{i',k} - \left(x_{i,k} + o_{i,k} + s_{k,\alpha_i,\alpha_{i'}}\right) \geq -M z_{i,i'} - 2M(1 - a_{i',k}) - 2M(1 - a_{i,k}),$$
$$\text{for } i = 1, 2, \cdots, I, \; i' = 1, 2, \cdots, I, \; i \neq i', \text{ and } k = 1, 2, \cdots, K, \tag{7}$$

$$x_{i,k} \geq \left(t_k^{\mathrm{r}} + s_{k,\beta_k,\alpha_i}\right) a_{i,k}, \qquad \text{for } i = 1, 2, \cdots, I, \text{ and } k = 1, 2, \cdots, K, \tag{8}$$

$$\sum_{k=1}^{K} x_{i,k} \leq \tau_i, \qquad \text{for } i = 1, 2, \cdots, I, \tag{9}$$

$$\sum_{k=1}^{K} \left(x_{i,k} + o_{i,k} a_{i,k}\right) - p_i \leq d_i, \qquad \text{for } i = 1, 2, \cdots, I, \tag{10}$$

$$x_{i,k} \in \mathbb{R}_{\geq 0}, \; a_{i,k} \in \{0, 1\} \qquad \text{for } i = 1, 2, \cdots, I, \text{ and } k = 1, 2, \cdots, K,$$

$$z_{i,i'} \in \{0, 1\} \qquad \text{for } i = 1, 2, \cdots, I, \; i' = 1, 2, \cdots, I, \text{ and } i \neq i',$$

$$d_i \in \mathbb{R}_{\geq 0} \qquad \text{for } i = 1, 2, \cdots, I.$$

Let τ_i be the latest start time of the ith lot due to residence time constraints. M is a large number. We define decision variables $x_{i,k} \in \mathbb{R}_{\geq 0}$, for $i = 1, 2, \cdots, I$ and $k = 1, 2, \cdots, K$, $z_{i,i'} \in \{0, 1\}$, for $i = 1, 2, \cdots, I$, $i' = 1, 2, \cdots, I$ and $i \neq i'$, $a_{i,k} \in \{0, 1\}$, for $i = 1, 2, \cdots, I$ and $k = 1, 2, \cdots, K$, and $d_i \in \mathbb{R}_{\geq 0}$, for $i = 1, 2, \cdots, I$. $x_{i,k} = 0$ means we do not use the kth machine to perform the operation for the ith lot. If $x_{i,k} > 0$, the ith lot is assigned to the kth machine and the process starts at time $x_{i,k}$. $z_{i,i'} \in \{0, 1\}$ is a binary decision variable. If the i'th lot starts earlier, we have $z_{i,i'} = 1$, otherwise $z_{i,i'} = 0$. $a_{i,k} = 1$ if the ith lot is assigned to the kth machine. Otherwise, $a_{i,k} = 0$. d_i is the delay time of the ith lot. Thus, the mixed integer programming model is developed as shown in the previous page.

Since exactly one machine is required for each lot, and thus the constraint given by (2) holds. The process time is always greater than one minute. Therefore, if $o_{i,k} > 0$, $a_{i,k}$ can be either 1 or 0. Otherwise, $a_{i,k} = 0$. M is always greater than $x_{i,k}$. The ith lot can be assigned to the kth machine, only when the kth machine is qualified for the ith lot. It means that $x_{i,k}$ and $a_{i,k}$ can be nonzero only when $o_{i,k}$ is nonzero. Thus, we have constraints (3) and (4). e_i is the time when the ith lot enters the buffer, and the start time of a lot should be later than its arrival time. Thus, we have (5).

Constraints (6) and (7) restrict that a machine processes one lot at a time. $x_{i,k}$ and $x_{i',k}$ should be mutually compared only when both the ith lot and the i'th lot are assigned to the kth machine, which means that both $x_{i,k}$ and $x_{i',k}$ are greater than zero. Therefore, (6) and (7) always hold, if either $x_{i,k}$ or $x_{i',k}$ is equal to zero. If both $x_{i,k}$ and $x_{i',k}$ are positive, (6) and (7) become (11) and (12), respectively, which are presented as follows.

$$x_{i',k} - x_{i,k} \leq M\left(1 - z_{i,i'}\right) - 1, \tag{11}$$

$$x_{i',k} - \left(x_{i,k} + o_{i,k} + s_{k,\alpha_i,\alpha_{i'}}\right) \geq -M z_{i,i'}. \tag{12}$$

Since both i and i' traverse set $\{1, 2, \cdots, I\}$, we only compare $x_{i,k}$ and $x_{i',k}$ when the ith lot starts process earlier than the i'th lot. If the i'th lot starts earlier, we have $z_{i,i'} = 1$ and (11) and (12) become (13) and (14), respectively, presented as follows.

$$x_{i',k} - x_{i,k} \leq -1, \tag{13}$$

$$x_{i',k} - \left(x_{i,k} + o_{i,k} + s_{k,\alpha_i,\alpha_{i'}}\right) \geq -M, \tag{14}$$

which always hold. If the ith lot starts earlier, then we require that the i'th lot should start at least after the ith lot finishes the process. In this case, we have $z_{i,i'} = 0$, and (11) and (12) become (15) and (16), respectively.

$$x_{i',k} - x_{i,k} \leq M - 1, \tag{15}$$

$$x_{i',k} - \left(x_{i,k} + o_{i,k} + s_{k,\alpha_i,\alpha_{i'}}\right) \geq 0. \tag{16}$$

(15) always holds, and (16) suggests that if $x_{i',k}$ is greater than $x_{i,k}$ then it should be at least greater than $(x_{i,k} + o_{i,k} + s_{k,\alpha_i,\alpha_{i'}})$.

Any lot assigned to the kth machine should start later than t_k^{r} plus conversion time, which is presented in (8). If the ith lot is not assigned to the kth machine, then both $x_{i,k}$ and $a_{i,k}$ are equal to zero and (8) still holds. Otherwise, $x_{i,k}$ should be greater than or equal to $(t_k^{\mathrm{r}} + s_{k,\beta_k,\alpha_i})$. (9) has residence time constraints to be satisfied. (10) gives the delay time, which is to be minimized in objective function (1).

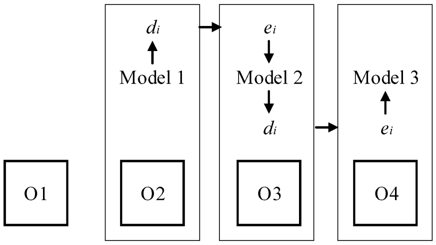


Fig. 7. All operations are coordinated

Thus, a mixed integer programming model is developed. The adjustment of the master schedule for the operation with long machine failure can be obtained by solving the model.

2) Warm start: When a machine has a long failure at one operation, the lots assigned to it remain in the buffer until it is repaired. Lots assigned to other machines of the same operation still follow the original master schedule. This approach is how the production is carried out without real-time intervention, which, though not ideal, can serve as a feasible solution within the model. This solution is taken as a warm start for the solver to speed up computation.

To address such a real-time decision making problem, we prefer to reach a good solution quickly rather than search for the optimal solution with long computation time. Warm start acts as a benchmark for an algorithm to solve the model, and thus the output is always better than the benchmark. Therefore, even with a very short computation time provided, an adjustment can be obtained at least better than no control.

3) Adjustment of downstream operations: In most cases of long machine failures, despite minimizing the objective function (1), some lots may still have positive delay times d_i if the model is only applied to one operation. These lots, failing to complete the operation on time, subsequently lead to delays in arriving downstream operations. Therefore, the downstream operations should also be adjusted according to the adjustment of the upstream operations. The process of this adjustment for downstream operations and the information-sharing mechanism are illustrate in Fig. 7. Assume long machine failure occurs in the second operation. The first model is developed for the second operation. If the delay time d_i is not zero, it will be used to update the arrival time e_i of the third operation. The same model provided in Section IV-B1 is then developed with the updated e_i and solved for the third operation. If the output d_i is of third operation is not zero, then imported to the model for the fourth operation. Thus, a model is developed for a single operation each time, and all operations are coordinated in a decentralized way.

TABLE I
SYSTEM CONFIGURATION OF THE SEMICONDUCTOR PACKAGING LINE

Setting	Value
Number of operations	4
Number of machines at operation 1	11
Number of machines at operation 2	3
Number of machines at operation 3	8
Number of machines at operation 4	15
Number of product groups	20
Span of schedule	3 weeks
Number of lots	528

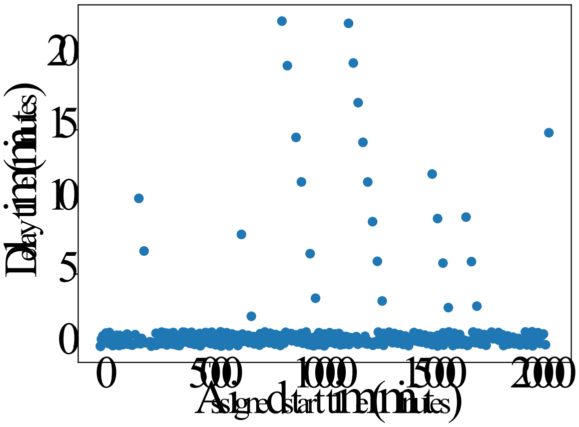


Fig. 8. Lot delay time over 3 weeks

V. EXPERIMENT

A. System configuration

The system configuration of the semiconductor packaging line in this experiment is presented in TABLE I. There are four operations, and each operation consists of several machines.

A simulation model is developed with Python and SimPy package. The parameters for building the simulation model are estimated from a real-world semiconductor packaging line. A 3-week master schedule for more than 500 lots of 20 different product groups is available. Both short and long machine failures are randomly generated in simulation run. Fig. 8 presents the delay time of all lots at the first operation in a simulation run. The horizontal axis gives the assigned start time of a lot, and the vertical axis shows the delay time. Most lots are processed in time, and their delay time is close to zero. When a machine fails for around 20 minutes, it could cause a lot to be delayed for about 20 minutes. Around 4 minutes is assigned as extra time to each lot. Thus, the system is recovering until the delay time finally reaches zero.

The first and second operations are subject to long machine failure. The mixed integer programming model, introduced in Section IV-B1, is developed with Python and solved by CPLEX Optimizer. An initial solution is given to the CPLEX Optimizer as a warm start. The maximum computation time is set to be 2 minutes. The best solution is exported if the optimal solution is not obtained within the maximum computation time. The experiment is conducted on a desktop computer with Intel(R) Core(TM) i7-10700 CPU, 16 GB RAM, and 64-bit Windows 10 Home operating system. At most 6 threads are assigned to the CPLEX Optimizer.

TABLE II
ORIGINAL MASTER SCHEDULE (MINUTE)

Machine	Lot	Ready time	Actual start time	Actual finish time	Assigned finish time	Delay time
M59	L406	3119	3173	3215.5	3216	0
M59	L372	3230	3230	3273.5	3274	0
M59	L82	3358	3358	3402.5	3403	0
M59	L94	3411	3411	3455.5	3456	0
M59	L381	3450	3456	3499.5	3500	0
M59	L136	3624	3624	3668.5	3669	0
M59	L84	3605	3669	3713.5	3714	0
M59	L65	3852	3852	3896.5	3897	0
M60	L378	2629	3208	3241.6	3242	0
M60	L396	3163	3242	3248.9	3249	0
M60	L322	1919	3249	3297.5	3298	0
M60	L463	2445	3298	3345.5	3346	0
M60	L166	3377	3377	3421.5	3422	0
M60	L389	3425	3425	3473.5	3474	0
M60	L90	3658	3658	3702.5	3703	0
M62	L108	2636	3167	3211.5	3212	0
M62	L285	3130	3212	3256.5	3257	0
M62	L210	2328	3257	3301.5	3302	0
M62	L168	2575	3302	3346.5	3347	0
M62	L12	1991	3347	3394.5	3395	0
M62	L371	2790	3395	3438.5	3439	0
M62	L373	3010	3439	3482.5	3483	0
M62	L222	2715	3483	3531.5	3532	0
M62	L277	3702	3702	3746.5	3747	0

In the experiment, the proposed method is compared with three other control methods, denoted by CM1, CM2, and CM3, respectively. The first benchmark, CM1, is to use no control. When a machine experiences a long failure, lots just wait until the machine is fixed. In the second method, CM2, lots are immediately moved to qualified machines without considering optimization, when a long machine failure happens. CM3 is similar to CM2 but has 90 minutes delay before responding to machine failure. Since it takes time from realizing the failure to rolling out the adjustment when manually done, this method reflects common practice in the shop floor of many semiconductor packaging lines.

B. An illustrative example

We let one machine at the second operation fail for 4 hours at 3167 minutes. TABLE II presents the original master schedule of the second operation for 12 hours after 3167 minutes. Three machines at the second operation are M59, M60, and M62. The ready time of a lot in the table is when the lot arrives in buffer and is ready for the operation. If there is no disruption, the actual start time of the operation goes according to the schedule, and the actual finish time is a little earlier than the assigned finish time. The delay time is zero for each lot.

TABLE III shows what will happen without control if machine M62 fails for 4 hours, namely the performance of CM1. The machine failure causes around 4 hours delay to 8 lots. The total delay time is 32.79 hours. By applying our proposed method, production control for 12 hours decision horizon is carried out, and the result is presented in TABLE IV. The total delay time is 9.55 hours, achieving 70.9% reduction.

TABLE III
IMPACT OF LONG MACHINE FAILURE ON MASTER SCHEDULE (MINUTE)

Machine	Lot	Ready time	Actual start time	Actual finish time	Assigned finish time	Delay time
M59	L406	3119	3173	3215.5	3216	0
M59	L372	3230	3230	3273.5	3274	0
M59	L82	3358	3358	3402.5	3403	0
M59	L94	3411	3411	3455.5	3456	0
M59	L381	3450	3456	3499.5	3500	0
M59	L136	3624	3624	3668.5	3669	0
M59	L84	3605	3669	3713.5	3714	0
M59	L65	3852	3852	3896.5	3897	0
M60	L378	2629	3208	3241.6	3242	0
M60	L396	3163	3242	3248.9	3249	0
M60	L322	1919	3249	3297.5	3298	0
M60	L463	2445	3298	3345.5	3346	0
M60	L166	3377	3377	3421.5	3422	0
M60	L389	3425	3425	3473.5	3474	0
M60	L90	3658	3658	3702.5	3703	0
M62	L108	2636	3407	3451.5	3212	239.5
M62	L285	3130	3451.5	3496.0	3257	239.0
M62	L210	2328	3496	3540.5	3302	238.5
M62	L168	2575	3540.5	3585.0	3347	238.0
M62	L12	1991	3585	3632.5	3395	237.5
M62	L371	2790	3632.5	3676.0	3439	237.0
M62	L373	3010	3676.0	3719.5	3483	236.5
M62	L222	2715	3719.5	3767.9	3532	235.9
M62	L277	3702	3767.9	3812.4	3747	65.4

TABLE IV
ADJUSTED MASTER SCHEDULE THROUGH THE PROPOSED METHOD (MINUTE)

Machine	Lot	Ready time	Actual start time	Actual finish time	Assigned finish time	Delay time
M59	L463	2445	3437	3484.5	3346	138.5
M59	L168	2575	3348	3392.5	3347	45.5
M59	L108	2636	3215.5	3260	3212	48
M59	L406	3119	3173	3215.5	3216	0
M59	L285	3130	3303.5	3348	3257	91
M59	L372	3230	3260	3303.5	3274	29.5
M59	L166	3377	3392.5	3437	3422	15
M59	L381	3450	3484.5	3528	3500	28
M60	L322	1919	3248.5	3297	3298	0
M60	L12	1991	3341.5	3389	3395	0
M60	L210	2328	3297	3341.5	3302	39.5
M60	L378	2629	3208	3241.6	3242	0
M60	L396	3163	3241.6	3248.5	3249	0
M60	L82	3358	3389	3433.5	3403	30.5
M60	L94	3411	3433.5	3478	3456	22
M60	L389	3425	3478	3526.4	3474	52.4
M60	L136	3624	3624	3668.5	3669	0
M60	L277	3702	3702	3746.5	3747	0
M60	L65	3852	3852	3896.5	3897	0
M62	L222	2715	3494	3542.5	3532	10.5
M62	L371	2790	3407	3450.5	3439	11.5
M62	L373	3010	3450.5	3494	3483	11
M62	L84	3605	3605	3649.5	3714	0
M62	L90	3658	3658	3702.5	3703	0

TABLE V
ADJUSTED MASTER SCHEDULE THROUGH CM2 (MINUTE)

Machine	Lot	Ready time	Actual start time	Actual finish time	Assigned finish time	Delay time
M59	L406	3119	3173	3215.5	3216	0
M59	L285	3130	3215.5	3260	3257	3
M59	L372	3230	3260	3303.6	3274	29.6
M59	L82	3358	3358	3402.5	3403	0
M59	L371	2790	3402.5	3446	3439	7
M59	L94	3411	3446	3490.5	3456	34.5
M59	L381	3450	3490.5	3534	3500	34
M59	L136	3624	3624	3668.5	3669	0
M59	L84	3605	3669	3713.5	3714	0
M59	L65	3852	3852	3896.5	3897	0
M60	L108	2636	3208	3252.5	3212	40.5
M60	L378	2629	3252.5	3286.2	3242	44.2
M60	L396	3163	3286.2	3293.1	3249	44.1
M60	L322	1919	3293.1	3341.5	3298	43.5
M60	L210	2328	3341.5	3386	3302	84
M60	L463	2445	3386	3433.5	3346	87.5
M60	L168	2575	3433.5	3478	3347	131
M60	L12	1991	3478	3525.5	3395	130.5
M60	L166	3377	3525.5	3570	3422	148
M60	L389	3425	3570	3618.4	3474	144.4
M60	L90	3658	3658	3700.5	3703	0
M62	L373	3010	3439	3482.5	3483	0
M62	L222	2715	3483	3531.4	3532	0
M62	L277	3702	3702	3746.5	3747	0

TABLE VI
ADJUSTED MASTER SCHEDULE THROUGH CM3 (MINUTE)

Machine	Lot	Ready time	Actual start time	Actual finish time	Assigned finish time	Delay time
M59	L406	3119	3173	3215.5	3216	0
M59	L372	3230	3230	3273.5	3274	0
M59	L82	3358	3358	3402.5	3403	0
M59	L285	3130	3407	3451.5	3257	194.5
M59	L371	2790	3451.5	3495	3439	56
M59	L94	3411	3495	3539.5	3456	83.5
M59	L381	3450	3539.5	3583	3500	83
M59	L136	3624	3624	3668.5	3669	0
M59	L84	3605	3669	3713.5	3714	0
M59	L65	3852	3852	3896.5	3897	0
M60	L378	2629	3208	3241.6	3242	0
M60	L396	3163	3242	3248.9	3249	0
M60	L322	1919	3249	3297.5	3298	0
M60	L463	2445	3298	3345.5	3346	0
M60	L166	3377	3377	3421.5	3422	0
M60	L108	2636	3421.5	3466	3212	254
M60	L210	2328	3466	3510.5	3302	208.5
M60	L168	2575	3510.5	3555	3347	208
M60	L12	1991	3555	3602.5	3395	207.5
M60	L389	3425	3602.5	3650.9	3474	176.9
M60	L90	3658	3658	3700.5	3703	0
M62	L373	3010	3439	3482.5	3483	0
M62	L222	2715	3483	3531.4	3532	0
M62	L277	3702	3702	3746.5	3747	0

As is shown in TABLEs V and VI, CM2 and CM3 can mitigate the impact of machine failure but perform not as well as our proposed method. The comparison of total delay time is presented in Fig. 9.

The delay time of the second operation after adjustment is taken as input for the adjustment of the third operation, and it results in 8 minutes total delay time at the third operation. Then, the 8 minutes delay time does not further cause delay at the fourth operation.

C. Simulation experiment with randomly generated long machine failure

The simulation model is implemented in Python using the SimPy package to emulate a real-world semiconductor packaging line. Two distinct branches of experiments have been designed to assess the performance of the proposed method in reducing delays and mitigating time window violations.

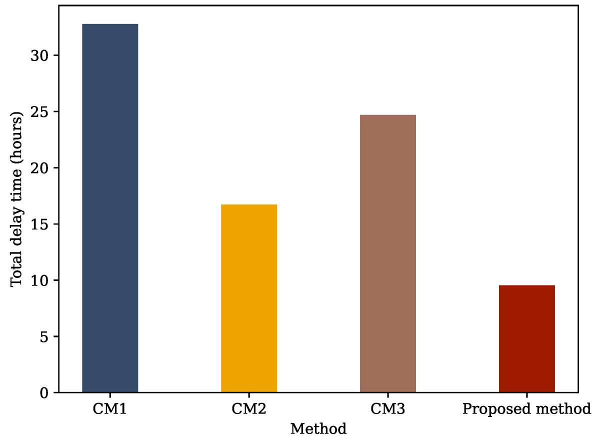
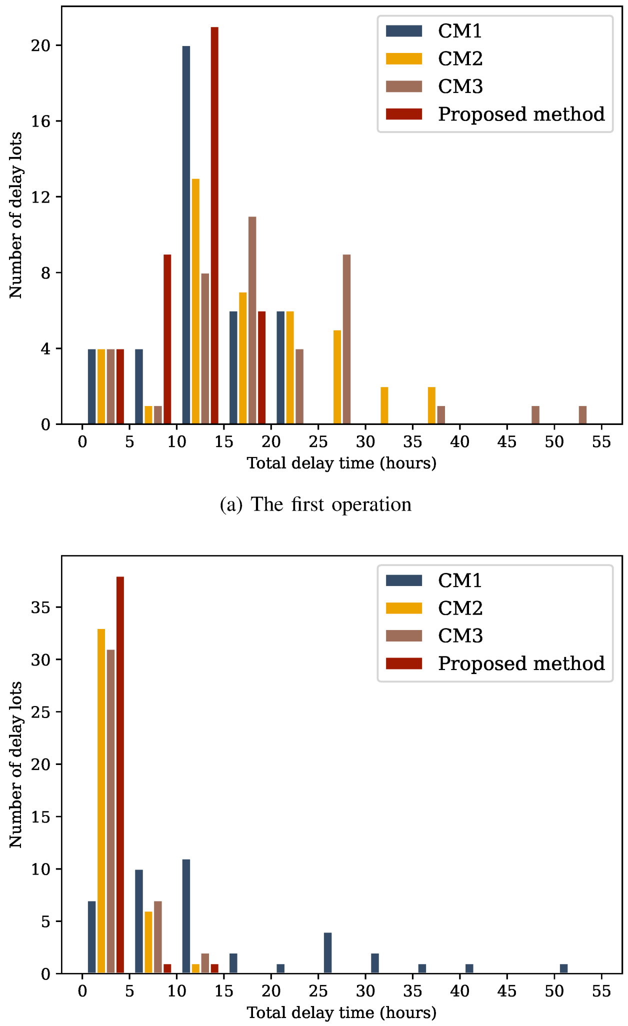


Fig. 9. Total delay time of different methods



(a) The first operation

(b) The second operation

Fig. 10. Total delay time comparison across various methods under randomly generated long machine failures

The first branch of experiments aims to measure the reduction of delays. We randomly generate long machine failures ranging from 200 minutes to 400 minutes. We test these failures in two separate sets of 40 cases each: one set for the first operation and another set for the second operation. Importantly, within each case, only one failure occurs. Figure 10 illustrates the total delay time using various methods, segmented into intervals: $[0, 5), [5, 10), ..., [45, 50), [50, \infty)$. Each interval reflects the count of delay lots falling within that range, with a higher concentration in lower intervals indicating superior results. The proposed method exhibits minimal improvement compared to CM1 during the first operation. The adjustments from CM2 and CM3 yield poorer results than CM1, possibly attributed to the limited idle time in the first operation. However, the scenario changes significantly during the second operation, which has much more idle time in the original master schedule. In this context, all production control methods, especially the proposed method, demonstrate considerable improvement compared to CM1. The boxplot of two operations is presented in Fig. 11 and also suggest the best performance of the proposed method.

The second branch focuses on mitigating time window violations. As shown in Fig. 1, there are three time windows. In the experiment, we focus on the time window between the first and second operations. The long machine failure duration is set from 100 minutes to 600 minutes. To thoroughly evaluate the performance of the proposed method, we randomly generate 1,000 cases for each failure duration at the second operation. Fig. 12 shows how frequently residence time constraint violation is in relation to the duration of machine failures. The horizontal axis represents the number of lots violating the residence time constraints, while the vertical axis represents the corresponding frequency. As the duration of machine failures increases, the number of cases with zero residence time constraint violations decreases. The results comparing three benchmark methods and the proposed method are presented in Fig. 13. In the CM1 method, the percentage of residence constraint violations increases from 10% to 30% as the failure time duration ranges from 100 minutes to 500 minutes. At the failure duration of 600 minutes, the violation rate remains around 30% of cases. Both CM2 and CM3 show an approximate 20% violation rate across all 6000 cases, with the violate rate of CM3 slightly higher compared to CM2. The proposed method demonstrates remarkable performance with zero violations among all 6,000 cases. Therefore, the proposed method has significant superiority over the three benchmark methods.

D. Control in run time

We integrate the mixed integer programming model into the simulation model to simulate the scenario in real-world factory, where a long machine failure occurs and the production control is carried out. The factory carries out production according to a schedule, but the system dynamics can deviate from the schedule due to uncertainty. Thus, before any adjustment, one needs to obtain parameters of the mixed integer programming model from real-time system state.

- *Lot set within decision horizon.* Machines may have short delay, and thus the lot set to be considered cannot be directly obtained from the master schedule. The lot set

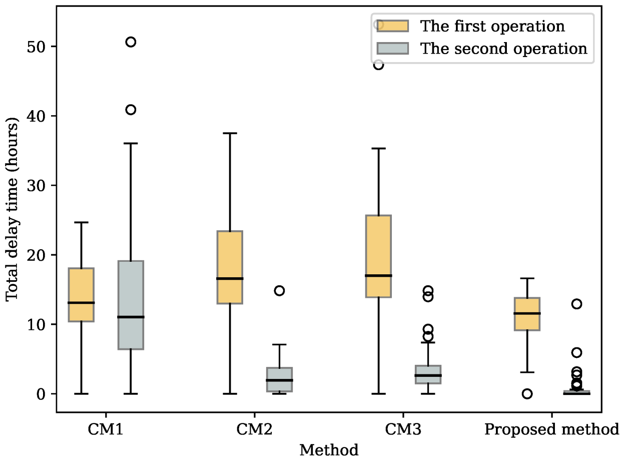


Fig. 11. Boxplot of total delay time with different method of two operations

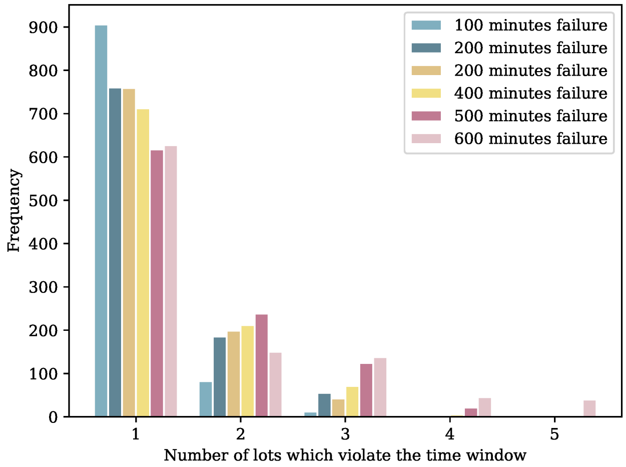


Fig. 12. Histogram of lots having time window violation

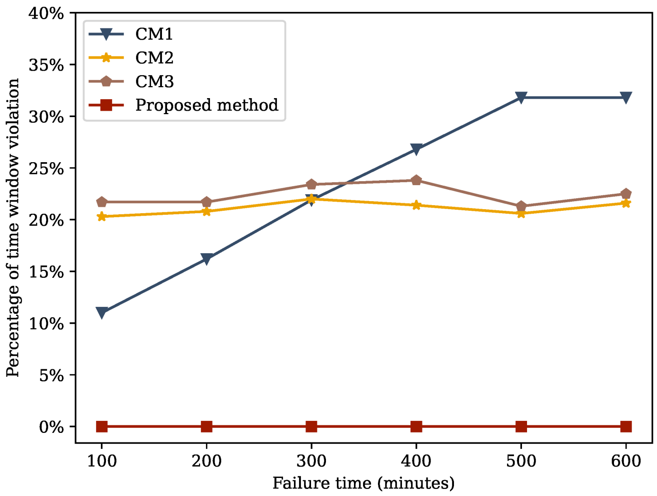


Fig. 13. Percentage of cases that have time window violation

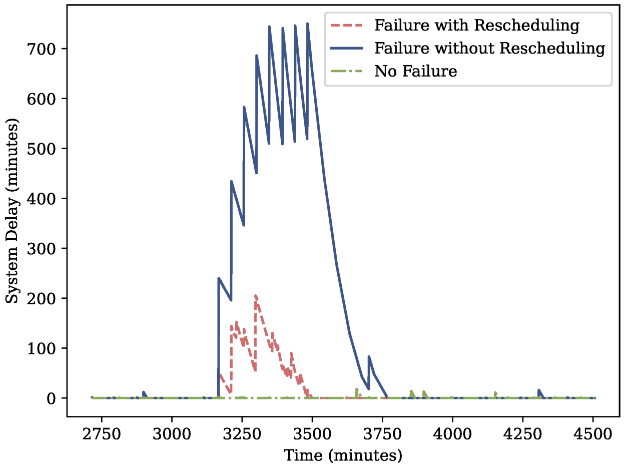


Fig. 14. System delay at the second operation: A comparative study of failure without rescheduling (2019.52 minutes total delay time) and failure with rescheduling (552.53 minutes total delay time) against no failure scenarios (82.82 minutes total delay time).

can be determined by checking the lot being processed on each machine.

- *The time when a machine becomes available* t_k^r. Due to short machine failure, the actual available time could be later than what suggests in the master schedule. By checking lots being processed and their start time, t_k^r for each machine can be determined.
- *The time that lot arrives in buffer* e_i. Lots may not arrive in buffer in time. The actual e_i can be later than what suggests in the master schedule. By checking the upstream machines and their delay time, e_i of each lot can be estimated.
- *The residence time constraint* τ_i. If the upstream operation is delayed, the actual τ_i could be greater than what suggests in the master schedule. The upstream operations should be considered to get the value of τ_i.

All other parameters are independent of real-time system state and can be easily obtained from master schedule.

After integrating the mixed integer programming model into the simulation, we create a long machine failure at the second operation. To assess the effectiveness of the proposed method, we compare the system delay under different scenarios: no failure, failure with rescheduling, and failure without rescheduling. The system delay is defined as the cumulative delay time of all lots in progress at any given time. The simulated machine failure spans 4 hours, starting at 3167 minutes into the simulation. To focus on the critical period around the failure, we present results specifically for this time frame. Figure 14 illustrates the outcomes of three different settings:

- The green dash-dot line represents the system delay when no failure occurs.
- The red dashed line depicts the system delay when the proposed rescheduling method is employed.
- The blue line represents the system delay when no rescheduling method is applied in the event of a failure.

Upon analyzing Figure 14, it is clear that the peak system delay without rescheduling exceeds 700 minutes, while with rescheduling, the peak delay is reduced to around 200

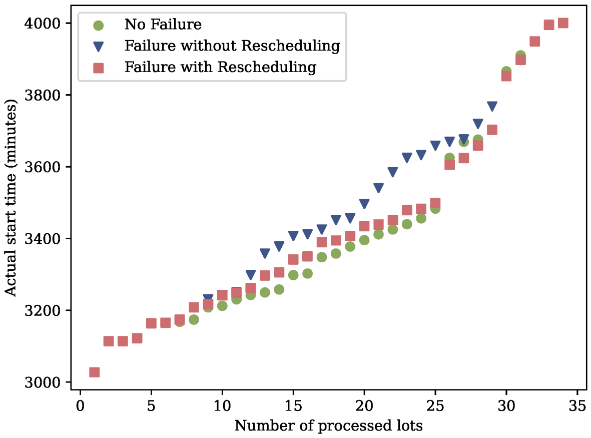


Fig. 15. The number of processed lots at given time around the failure start time

minutes. Furthermore, without rescheduling, the system takes until around 3750 minutes to recover, whereas the proposed rescheduling method facilitates a quicker recovery, with the system recovering around 3500 minutes into the simulation. Additionally, Fig. 15 presents a lot-level comparison among the three settings, specifically focusing on lots with assigned start times around the failure start time. The horizontal axis illustrates the number of lots processed over time, as indicated on the vertical axis. At any given time, the setting without long machine failure always processes more lots, representing the normal throughput for the system. It's worth highlighting that the setting with no rescheduling necessitates more recovery time than the proposed rescheduling method to catch up with the normal throughput.

VI. Discussion

The idle time shown in Fig. 5 is essential. The essence of the adjustment is to reduce delay time by making use of the idle time. Idle time exists for two reasons. First, the operation may not be the bottleneck of the production system. Machines may often be idle, waiting for lots to come. Second, the idle time could also be created in the master schedule purposely to trade efficiency for flexibility. If there is not much idle time, an operation may not be able to recover. In this situation, the proposed method can still minimize total delay time and the residence time constraint violation. In addition, a fast response to machine failures provides the master scheduler with enough time to remake a new master schedule.

The decision horizon depends on the length of machine failure and idle time. The long machine failure could lasts several hours or a whole day. The mixed integer programming needs to consider all lots directly impacted. A short decision horizon means there may not be enough idle time to make use of, while a long decision horizon results in a high problem complexity.

This study assumes that we know how long the machine will fail. When a machine fails to work, its repair time can be estimated according to the failure type. This estimation may not always be accurate. If the machine is capable of getting back to work early, it does not impact the adjusted master schedule. If the repair is delayed, based on the information about how long it could be delayed, production control can be performed again with similar process.

VII. Conclusion

Production systems are susceptible to uncertain disruption. This study focuses on a semiconductor packaging line. We classify machine failures into short machine failures and long machine failures and deal with them separately. The simulation experiment suggests that the proposed method can respond quickly to machine failures. This paper provides production engineers in semiconductor packaging lines with a practical tool operation management.

There are two directions for future research. First, model properties can be further explored and algorithms to solve the model can be studied. Second, more practical requirements observed in semiconductor packaging lines can be considered in modeling. Re-entrance is commonly seen. In some operations, lots with small size can merge to form a single large lot. Sometimes, a large lot splits into multiple small lots. Semiconductor packaging lines also face different disruptions, other than machine failure. It is worth studying fast response with those requirements considered.

References

[1] F. Zhang, J. Song, Y. Dai, and J. Xu, "Semiconductor wafer fabrication production planning using multi-fidelity simulation optimisation," *International Journal of Production Research*, vol. 58, no. 21, pp. 6585–6600, 2020.

[2] L. Li, Z. Sun, M. Zhou, and F. Qiao, "Adaptive dispatching rule for semiconductor wafer fabrication facility," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 354–364, 2012.

[3] A. Chen and R. H.-Y. Lo, *Semiconductor packaging: materials interaction and reliability*. Taylor & Francis, 2012.

[4] Y. Lu and F. Ju, "Smart manufacturing systems based on cyber-physical manufacturing services (cpms)," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 15 883–15 889, 2017.

[5] F. Wang, Y. Lu, and F. Ju, "Condition-based real-time production control for smart manufacturing systems," in *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2018, pp. 1052–1057.

[6] B.-s. Chung, J. Lim, I.-B. Park, J. Park, M. Seo, and J. Seo, "Setup change scheduling for semiconductor packaging facilities using a genetic algorithm with an operator recommender," *IEEE Transactions on Semiconductor Manufacturing*, vol. 27, no. 3, pp. 377–387, 2014.

[7] J. T. Lin and C.-M. Chen, "Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing," *Simulation Modelling Practice and Theory*, vol. 51, pp. 100–114, 2015.

[8] L. Y. Hsieh and C.-B. Cheng, "Efficient due-date quoting and production scheduling for integrated circuit packaging with reentrant processes," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 8, no. 8, pp. 1487–1495, 2018.

[9] H. Aytug, M. A. Lawley, K. McKay, S. Mohan, and R. Uzsoy, "Executing production schedules in the face of uncertainties: A review and some future directions," *European Journal of Operational Research*, vol. 161, no. 1, pp. 86–110, 2005.

[10] B. Han and D.-S. Kim, "Moisture ingress, behavior, and prediction inside semiconductor packaging: A review," *Journal of Electronic Packaging*, vol. 139, no. 1, 2017.

[11] M. I. Reiman and Q. Wang, "Asymptotically optimal inventory control for assemble-to-order systems with identical lead times," *Operations Research*, vol. 63, no. 3, pp. 716–732, 2015.

[12] Z. Atan, T. Ahmadi, C. Stegehuis, T. de Kok, and I. Adan, "Assemble-to-order systems: A review," *European Journal of Operational Research*, vol. 261, no. 3, pp. 866–879, 2017.

[13] F. Ju and J. Li, “A bernoulli model of selective assembly systems,” *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 1692–1697, 2014.

[14] F. Ju, J. Li, and W. Deng, “Selective assembly system with unreliable bernoulli machines and finite buffers,” *IEEE transactions on automation science and engineering*, vol. 14, no. 1, pp. 171–184, 2016.

[15] C.-B. Yan and Z. Zheng, “Problem formulation and solution methodology for energy consumption optimization in bernoulli serial lines,” *IEEE Transactions on Automation Science and Engineering*, 2020.

[16] K. Kang and V. Subramaniam, “Integrated control policy of production and preventive maintenance for a deteriorating manufacturing system,” *Computers & Industrial Engineering*, vol. 118, pp. 266–277, 2018.

[17] Y. Kang and F. Ju, “Flexible preventative maintenance for serial production lines with multi-stage degrading machines and finite buffers,” *IISE Transactions*, vol. 51, no. 7, pp. 777–791, 2019.

[18] Y. Kang, H. Yan, and F. Ju, “Performance evaluation of production systems using real-time machine degradation signals,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 1, pp. 273–283, 2019.

[19] F. Qiao, Y. Ma, M. Zhou, and Q. Wu, “A novel rescheduling method for dynamic semiconductor manufacturing systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 5, pp. 1679–1689, 2018.

[20] D. Gupta, C. T. Maravelias, and J. M. Wassick, “From rescheduling to online scheduling,” *Chemical Engineering Research and Design*, vol. 116, pp. 83–97, 2016.

[21] L. Liu, H.-y. Gu, and Y.-g. Xi, “Robust and stable scheduling of a single machine with random machine breakdowns,” *The International Journal of Advanced Manufacturing Technology*, vol. 31, no. 7, pp. 645–654, 2007.

[22] F. Qiao, L. Li, Y. Ma, and B. Shi, “Single machine oriented match-up rescheduling method for semiconductor manufacturing system,” in *International Conference on Intelligent Robotics and Applications*. Springer, 2012, pp. 217–226.

[23] R. J. Abumaizar and J. A. Svestka, “Rescheduling job shops under random disruptions,” *International journal of production research*, vol. 35, no. 7, pp. 2065–2082, 1997.

[24] S. Mason, S. Jin, and C. Wessels, “Rescheduling strategies for minimizing total weighted tardiness in complex job shops,” *International Journal of Production Research*, vol. 42, no. 3, pp. 613–628, 2004.

[25] I.-B. Park, J. Huh, J. Kim, and J. Park, “A reinforcement learning approach to robust scheduling of semiconductor manufacturing facilities,” *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1420–1431, 2019.

[26] R. Naebulharam and L. Zhang, “Bernoulli serial lines with deteriorating product quality: performance evaluation and system-theoretic properties,” *International Journal of Production Research*, vol. 52, no. 5, pp. 1479–1494, 2014.

[27] J.-H. Lee, J. Li, and J. A. Horst, “Serial production lines with waiting time limits: Bernoulli reliability model,” *IEEE Transactions on Engineering Management*, vol. 65, no. 2, pp. 316–329, 2017.

[28] J.-H. Lee, C. Zhao, J. Li, and C. T. Papadopoulos, “Analysis, design, and control of bernoulli production lines with waiting time constraints,” *Journal of Manufacturing Systems*, vol. 46, pp. 208–220, 2018.

[29] F. Ju, J. Li *et al.*, “Transient analysis of bernoulli serial line with perishable products,” *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 1670–1675, 2015.

[30] F. Ju, J. Li, and J. A. Horst, “Transient analysis of serial production lines with perishable products: Bernoulli reliability model,” *IEEE Transactions on automatic control*, vol. 62, no. 2, pp. 694–707, 2016.

[31] N. Kang, F. Ju, and L. Zheng, “Transient analysis of geometric serial lines with perishable intermediate products,” *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 149–156, 2016.

[32] F. Wang and F. Ju, “Decomposition-based real-time control of multi-stage transfer lines with residence time constraints,” *IISE Transactions*, pp. 1–17, 2020.

[33] F. Wang, F. Ju, and N. Kang, “Transient analysis and real-time control of geometric serial lines with residence time constraints,” *IISE Transactions*, vol. 51, no. 7, pp. 709–728, 2019.

[34] F. Wang and F. Ju, “Simulation-based real-time production control with different classes of residence time constraints,” in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2022, pp. 1860–1865.

[35] Q. Zhu, N. Wu, Y. Qiao, and M. Zhou, “Scheduling of single-arm multi-cluster tools with wafer residency time constraints in semiconductor manufacturing,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 1, pp. 117–125, 2014.

[36] C. Pan, M. Zhou, Y. Qiao, and N. Wu, “Scheduling cluster tools in semiconductor manufacturing: Recent advances and challenges,” *IEEE transactions on automation science and engineering*, vol. 15, no. 2, pp. 586–601, 2017.

[37] A. Klemmt and L. Mönch, “Scheduling jobs with time constraints between consecutive process steps in semiconductor manufacturing,” in *Proceedings of the 2012 winter simulation conference (WSC)*. IEEE, 2012, pp. 1–10.

[38] H.-J. Kim and J.-H. Lee, “Three-machine flow shop scheduling with overlapping waiting time constraints,” *Computers & Operations Research*, vol. 101, pp. 93–102, 2019.

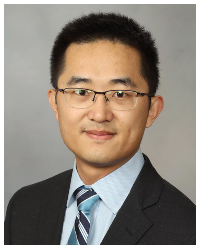
[39] J. E. Galloway and B. M. Miles, “Moisture absorption and desorption predictions for plastic ball grid array packages,” in *InterSociety Conference on Thermal Phenomena in Electronic Systems, I-THERM V*. IEEE, 1996, pp. 180–186.

[40] S. Yoon, C. Jang, and B. Han, “Nonlinear stress modeling scheme to analyze semiconductor packages subjected to combined thermal and hygroscopic loading,” *Journal of Electronic Packaging*, vol. 130, no. 2, pp. 024 502–1–024 502–5, 2008.

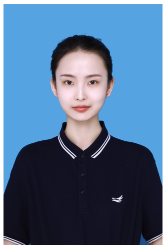
[41] T. Y. Tee and Z. Zhong, “Integrated vapor pressure, hygroswelling, and thermo-mechanical stress modeling of qfn package during reflow with interfacial fracture mechanics analysis,” *Microelectronics reliability*, vol. 44, no. 1, pp. 105–114, 2004.

[42] M. Mastrolilli and L. M. Gambardella, “Effective neighbourhood functions for the flexible job shop problem,” *Journal of scheduling*, vol. 3, no. 1, pp. 3–20, 2000.

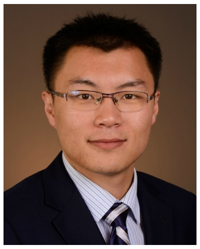
[43] J. Huh, I. Park, S. Lim, B. Paeng, J. Park, and K. Kim, “Learning to dispatch operations with intentional delay for re-entrant multiple-chip product assembly lines,” *Sustainability*, vol. 10, no. 11, p. 4123, 2018.



Feifan Wang received the bachelor’s degree from the Department of Industrial Engineering, Zhejiang University of Technology, Hangzhou, China, in 2013, the master’s degree from the Department of Industrial and Systems Engineering, Zhejiang University, Hangzhou, China, in 2016, and the Ph.D. degree from the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ, USA, in 2021. He is currently an Assistant Professor with the Department of Industrial Engineering at Tsinghua University, Beijing, China. His research focuses on modeling, analysis, optimization, and control of complex systems, with applications in healthcare delivery systems and production systems. He is also a member of the Institute for Operations Research and the Management Sciences. He was a recipient of multiple awards, including the Design and Manufacturing Best Paper Award from the IISE Transactions, the Best Student Paper Award from IEEE CASE, and the Dean’s Dissertation Award from ASU. He has been a finalist for the Best Paper Award on Healthcare Automation from IEEE CASE twice.



Yutong Su earned her Bachelor’s degree in Statistics from Xi’an Jiao Tong University, Xi’an, China, in 2019. Continuing her academic journey, she pursued a Master’s degree in Statistics from Rice University, Houston, USA, graduating in 2020. Currently, she is a PhD candidate in industrial engineering at Arizona State University, Tempe, USA. Her research interests include machine learning, optimization and scheduling in manufacturing systems.



Feng Ju is an associate professor with the School of Computing and Augmented Intelligence, at Arizona State University, in Tempe, AZ. He received a B.S. degree from Shanghai Jiao Tong University, Shanghai, China, in 2010, and an M.S. degree in electrical and computer engineering and Ph.D. degree in industrial and systems engineering from the University of Wisconsin, Madison, WI, USA, in 2011 and 2015, respectively. His current research interests include modeling, analysis, continuous improvement, and optimization of manufacturing systems and additive manufacturing. He is also a member of the Institute for Operations Research and the Management Sciences, Institute of Industrial and Systems Engineers, and Institute of Electrical and Electronics Engineers. He has been a recipient of multiple awards, including the best paper awards in IISE Transactions and IFAC MIM and best student paper awards in IEEE CASE and IFAC INCOM.

Balakrishnan (Bala) Ananthanarayanan works at Intel Corporation on Optimization applications using Mathematical Integer Programming (MIP) and constraint programming (CP) for scheduling and Inventory Optimization. Bala graduated with a Master of Science in Software Engineering from India. Bala's interests include exploring open machine/deep learning data sets and novel techniques to glean meaningful insights from them.

Husam Dauod received his M.S. (2016) and Ph.D. (2019) in Industrial and Systems Engineering from Binghamton University, SUNY. He is currently a Research Scientist at Intel Corp. His research interests include mathematical modelling, optimization, and simulation and their application in smart manufacturing, warehousing, and supply chain.

Nital S Patel (SM'01) received the B.Tech degree from the Indian Institute of Technology in 1991, and the M.S. and PhD. Degrees from the University of Maryland, College Park, in 1993 and 1995, respectively, all in electrical engineering.

The is currently a Senior Principal Engineer with Intel Corporation, Chandler, AZ, where his focus in on research and development of smart manufacturing capabilities. He holds 11 patents and has numerous publications in the areas of semiconductor process control and machine learning and AI applications for machine vision and time series analysis. He is the recipient of the University of Maryland, Electrical and Computer Engineering Distinguished Alumni award and the Mahboob Khan award from the Semiconductor Research Corporation. Nital has also served as a past editor of the IEEE Transactions on Semiconductor Manufacturing.