Preventive maintenance scheduling: a simulation-optimization approach

Agus Darmawan and D. Daniel Sheu

Department of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia;
Department of Industrial Engineering & Engineering Management, National Tsing Hua University, Hsinchu, Taiwan, R.O.C

ABSTRACT
This paper presents a framework for preventive maintenance (PM) scheduling in the semiconductor industry. We propose an approach for finding PM’s start time within a PM window to minimize production losses due to maintenance activities. In this study, we consider re-entrant process in which wafers will enter the same equipment location several times, but in different stages and sometimes different processes. Due to the optimization problem’s complexity, we develop meta-heuristics such as a genetic algorithm and particle swarm optimization to solve it and compare with the resource leveling as well as the baseline. In the algorithm, we embed discrete event simulation to mimic a wafer fab process and get its performance. The proposed approach able to identify the best arrangement of PM’s start time within a PM window and provides a way to optimize PM schedules for a complex system by simultaneously utilizing meta-heuristics and discrete event simulation.

1. Introduction
The semiconductor industry has grown rapidly to become a significant driver for economic growth. Many companies in this area are trying to compete in a competitive market by increasing their system performance in term of throughput rate, work-in-process inventory, cycle time, and overall equipment efficiency (Montoya- Torres, 2006). In practice, industries are often interested in maximizing throughput rate to increase profits by maximizing equipment availability. However, this problem may not be straightforward to solve. Since equipment should be maintained at periodic time intervals usually within a PM window, and equipment cost is a large proportion of manufacturing cost in the semiconductor industry, optimizing preventive maintenance schedules is an important issue, and various conditions must be considered in scheduling equipment effectively.

There are mainly two types of maintenance activities: preventive maintenance, whose activities can be planned and corrective maintenance, which is related to non-foreseeable breakdowns (De Jonge & Scarf, 2020; Marmier et al., 2009). Many companies’ resources
have been diverted towards maintenance management due to the importance and complexity involved in planning maintenance (Assid et al., 2015; Cheng et al., 2018; De Jonge & Scarf, 2020; Li et al., 2009). In term of the equipment quality, Panagiotidou and Tagaras (2008) divide maintenance into two types. First one is minimal maintenance that upgrades the equipment’s quality state without affecting its age. The second is perfect preventive maintenance that fully upgrades the equipment to the as-good-as-new condition.

Sheu and Wu (1999) report that two thirds to three-quarters of the manufacturing costs are equipment cost. Therefore, preventive maintenance scheduling is an essential topic in semiconductor manufacturing systems. PM is a reasonable solution to reduce the probability of machine failures. However, PM actions increase the production cost and consume the system’s time-capacity (Franciosi et al., 2020; Renna, 2012; Safaei et al., 2010). Previous studies have proposed possible methods for the preventive maintenance scheduling problem (PMSP). However, most of them used relatively simple cases in term of number of tools involved, number of PM tasks, assumption of PM duration, scheduling horizon, and the way of modelling a system (Mosley et al., 1998; Charles et al., 2003; Suryadi & Papageorgiou, 2004; Yao et al., 2007; Cassady & Kutanoğlu, 2005; Crespo Marquez et al., 2006; Moghaddam and Usher, 2011; Xiao et al., 2016; Chang, 2018). To deal with process planning, scheduling and PM decisions, minimizing the total completion time of a set of jobs becomes an important issue (Gurel & Akturk, 2008; Liao et al., 2017; Xiao et al., 2016).

There are many factors that should be considered simultaneously in PM scheduling. From the system point of view, system output is the function of several inputs, such as throughput, WIP, cycle time, equipment utilization, etc. Besides, realistic considerations for modelling the system should be incorporated, so that the results make sense and are applicable. Since it is difficult to use any analytical method, we combine heuristic optimization and discrete event simulation. Contributions of this study include:

1. Identifying the best arrangement for PM’s start time within a PM window.
2. Providing a framework for solving real complex systems by simultaneously utilizing meta-heuristics and discrete event simulation for optimization

2. Overview of the method

2.1. Problem statement

Semiconductor manufacturing systems are highly complex systems in which many conditions and inter-dependent activities should be considered (Ghahramani et al., 2020; Qiao et al., 2020; Wang et al., 2020). There are hundreds of step sequences during wafer fabrication, and each location in wafer fabs consists of several pieces of equipment. Since equipment degrades with production run times, preventive maintenance should be performed at a periodic time interval to prevent shut-downs.

Semiconductor manufacturing equipment generally has scheduled PM tasks within specific PM windows. As long as the PM time is within an allowable window, there is flexibility in the timing of the PMs. The best start time for each PM should be determined so that production losses due to maintenance downtime can be minimized, yielding an optimum throughput rate. Different PM tasks may use different measures to identify which chamber to be maintained, such as time units (e.g. hour, day, week, and month),
power consumption, wafer thickness and number of wafers. For example, chemical vapor deposition (CVD) machines use the number of wafers produced together with time units to quantify PM’s need (Sheu & Kuo, 2006).

In wafer fab, several products are produced by many machines and many stages in the system. This process is made more complicated by a re-entrant process, in which wafers will enter the same equipment location several times, but in different stages and sometimes different processes. This situation makes any analytical method extremely difficult. By applying meta-heuristics, such as genetic algorithms and particle swarm optimization, to generate random schemes for the start time of PM tasks, called input factors, discrete event simulation can identify the output of those input factors. The approach can identify a near-optimum scheme for the start time of PM tasks.

2.2. Scenario description

This study focuses on preventive maintenance scheduling at a Chemical Vapor Deposition (CVD). At this location, a chemical process is performed to produce high-purity, high-performance materials, and to deposit specific substances onto wafers, for desired wafer characteristics.

The materials/substrates are typically transferred by gas source or liquid source, see Figure 1. Due to the use of tools for CVD processes, parts of a tool such as chambers, furnace tubes, bell jars need to be maintained and cleaned periodically.

In this study, we consider 12 product types of wafer with varying product mix. We also take into account arrival schedules and their lot size. Each product moves from one facility (a group of the tool) to another with different sequences. There are several machines in a facility, where each machine consists of 1–3 chambers. A wafer may enter the same facility several times but in different stages (re-entrant process). We consider 15 different PM tasks with their PM window and duration for each machine, such as lamp exchange, heater exchange, valve exchange, etc. Since it is difficult to solve analytically, we use simulation-optimization to mimic the wafer fab process and solve PM scheduling problem.

2.3. Methodology

Figure 2 shows the flowchart of research methodology used in this study for preventive maintenance optimization.

Symbol definitions

- $i$: Index for tool, $i = 1, 2, \ldots, I$
- $j$: Index for chamber in a tool. The total number of chambers in tool $i$ is $J_i$, $j = 1, 2, \ldots, J_i$
- $s$: Index for the PM tasks. The maximum number of PM tasks at chamber $j$ of tool $i$ is $S_{ij}$, $s = 1, 2, \ldots, S_{ij}$.
- $t$: Index for time. $t = 1, 2, \ldots, T$ where $T$ is the planning horizon.
- $F_c$: Fitness value of chromosome $c$. ($F_c = O_c - O_{c, \text{lowest}}$) and $c = 1, 2, \ldots, C$
- $P_c$: Probability that chromosome $c$ would be selected.
- $O_c$: Simulation output of chromosome $c$.
- $O_{c, \text{lowest}}$: Lowest simulation output of chromosomes.

Decision variables
Figure 1. An example of CVD system.
Figure 2. The flowchart of research methodology.
\( ST_{ij} \): Start time for PM task \( s \) at chamber \( j \) of tool \( i \).

\( PM_{ij}(t) \): Binary decision variable regarding PM task \( s \) at chamber \( j \) of tool \( i \) at time \( t \).

\[
PM_{ij}(t) = \begin{cases} 
1 & t \in (ST_{ij}, ST_{ij} + d_{ij}) \\
0 & \text{otherwise}
\end{cases}
\]

\( PM_{ij}(t) = 1; \) The chamber is preventively maintained.

**Parameters**

\( L_{ij} \): Lower bound of PM window for PM task \( s \) at chamber \( j \) of tool \( i \).

\( U_{ij} \): Upper bound of PM window PM task \( s \) at chamber \( j \) of tool \( i \).

\( P \): Population size.

\( C \): Crossover rate.

\( M \): Mutation rate.

\( d_{ij} \): Duration for PM task \( s \) at chamber \( j \) of tool \( i \).

\( Ch_i \): The maximum number of chambers that can be maintained at the same time in tool \( i \).

**Objective Function:** In PM scheduling, two kinds of objective functions are tested for evaluating the proposed approaches, maximize throughput and minimize manpower, which are common in the semiconductor industry. Please note that we perform simulation-optimization separately for these two different objective functions and compare the results.

(a) Maximize throughput rate

The goal is to maximize throughput rate within the planning horizon in order to increase the number of wafers produced.

(a) Minimize manpower

The goal is to minimize the level of resources (manpower) within the planning horizon.

We use discrete-event simulation to get the throughput rate and the level of manpower. Although we set the objective function based on that indicator, we also collect information about cycle time, work in process (WIP) and equipment utilization at the end of simulation.

**Constraints**

Three constraints are identified as follows:

(1) The start time of a PM task must lie within the PM window for the corresponding chamber of the corresponding tool.

\[
L_{ij} \leq ST_{ij} \leq U_{ij}
\]

(2) There is no interruption during PM and there is only one PM at a time.

\[
(ST_{ij} + d_{ij}) \leq ST_{ijt}; \; s \neq t, s, t = 1, 2 \ldots S_{ij}
\]
Table 1. PM window.

| Tool (i) | Chamber (j) | PM task (s) | Spec. Unit | PM Window | Std. Time |
|----------|-------------|-------------|------------|-----------|-----------|
| 1        | 1           | PM 1        | Pcs        | \((L_{111}, U_{111})\) | \(d_{111}\) |
|          |             | PM 2        | H          | \((L_{112}, U_{112})\) | \(d_{112}\) |
| ...      |             | PM 5\_i    | Spec       | \((L_{155_i}, U_{155_i})\) | \(d_{155_i}\) |
| ...      | 1           | PM 1        | Pcs        | \((L_{111}, U_{111})\) | \(d_{111}\) |
|          |             | PM 2        | H          | \((L_{112}, U_{112})\) | \(d_{112}\) |
| ...      |             | PM 5\_i    | Spec       | \((L_{155_i}, U_{155_i})\) | \(d_{155_i}\) |
| ...      | 1           | PM 1        | Pcs        | \((L_{111}, U_{111})\) | \(d_{111}\) |
|          |             | PM 2        | H          | \((L_{112}, U_{112})\) | \(d_{112}\) |
| ...      |             | PM 5\_i    | Spec       | \((L_{155_i}, U_{155_i})\) | \(d_{155_i}\) |

Find the best level of start time within PM window

(3) Maximum number of chambers that can be maintained at the same time in a tool

\[
\sum_{j=1}^{J_i} \sum_{s=1}^{S_j} PM_{ijs}(t) \leq Ch_i(t)
\]  

Table 1 presents the problem addressed in this study. In semiconductor manufacturing, e.g. in Taiwan, there are different measures to identify the need for maintenance services such as number of wafers produced (pieces; PCS), units of time (hours; H), thickness, power consumption, or another specification unit (Spec). Since each PM task has a specific PM window, the start time for maintenance must be done within this window.

3. Design of genetic algorithm

This section explains the design of a genetic algorithm (GA) and the procedure for optimizing preventive maintenance schedules. GA is a well-known metaheuristic method for optimization problem with large spaces (Wang et al., 2019). In this study, Visual C++ and Flexsim 4.3 as discrete event simulators are simultaneously used in a genetic algorithm procedure, as shown in Figure 3.

Solutions to the PM problem are modeled as chromosomes, where each gene of a chromosome represents a start time of PM, see Figure 4. For the initial population of a genetic algorithm, we generate genes randomly. To ensure that solution is feasible, we consider constraints such as the start time of PMs must lie within the PM window \((L_{ijs} \leq ST_{ijs} \leq U_{ijs})\), only one chamber can be maintained at a time, and only one PM can be done at the same time. A redo action is performed whenever an infeasible solution exists during the initial population step.
Since genetic algorithms are sensitive to the parameters such as population size (P), crossover rate (C) and mutation rate (M), experiments are conducted to find the near-optimum values for these parameters, as in (Pongcharoen et al., 2002) and (Yang et al., 2007). We use a practical problem with a one-month planning horizon, as described in section 4, for the experimented design. Table 2 presents the results of regression for $3^K$ factorial design with three replications of a one-month planning horizon.

Table 2 shows that constant, P, C and $C^2$ are significant factors since the p-values for these predictors are less than 0.05. The coefficient of constants P, C and $C^2$ are 4177, 1.43, 112.3, 78.2 with standard errors of 15.09, 0.48, 31.82, and 21.74 respectively.

The output of throughput is modelled with the following formulae:

$$\text{Throughput rate} = 4177 + 1.43P + 112.3C - 78.2C^2 \quad (4)$$

According to equation 4, a high population would significantly increase the throughput rate. Since a high throughput rate is preferred and the highest level of experimental factors’ population size is 60, we use the population size of 60 for further study. By taking the first derivative of equation 4 with respect to crossover rate and constraining it with the level of crossover as in the experimental factors, the best crossover rate for this study is 0.718 ($\frac{dy}{dx} = 112.3 - 156.4C = 0$; $0.3 \leq C \leq 0.9$). Since the mutation rate is not a significant factor, and a low level is commonly adopted, the mutation rate of 0.02 is used. Hence, the values of population size, crossover rate and mutation rate are 60, 0.718 and 0.02, respectively. Parameter tuning for metaheuristics is an important step to make a
Figure 4. An example of chromosome.
good result. Taguchi’s experimental design, big bang-big crunch, grid search and other intelligent optimization methods can be used as alternatives for parameter tuning (Zhang et al., 2018; Gao, Zhou, et al. 2019; Wang & Kumbasar, 2019).

Combining genetic algorithm and discrete event simulation is time-consuming. We implement several methods to reduce computation time as follows:

1. Simulation is only performed for feasible solutions, and a penalty is directly added to infeasible chromosomes without running a simulation.
2. Converting all codes of Flexsim simulation model into C++ language. The default language in Flexsim 4.3 is Flexscript.
3. 3D graphical rendering of the model is closed to reduce computation time dramatically.
4. Three personal computers are used to run multiple models simultaneously.

A penalty is applied whenever an infeasible solution exists. The decision to determine whether chromosome generates feasible or infeasible solution is based on a violation of the existing constraints, as described in the previous section. For infeasible solution, we directly add a small number to that chromosome as the output. However, for feasible chromosomes, the discrete-event simulator evaluates the output. The details of the mechanism for integrating the genetic algorithm in Visual C++ and the discrete-event simulator (e.g. Flexsim) are described through the diagram and pseudo-code, as shown in Figure 5 and Table 3.

The method to integrate the genetic algorithm in Visual C++ and Flexsim is as follows. We send a feasible solution to the specific file in drive C and then the algorithm in Visual C++ call model of the system in Flexsim and wait until the Flexsim model closed by itself. When the Flexsim model is called, it will import data from drive C and run a simulation. At the end of simulation, Flexsim sends the result to drive C and then automatically closed by itself. The simulation result is read through an algorithm in Visual C++ as the output of the chromosome. By making this relationship, we get a near-optimal solution at the end.

Table 2. The result of regression analysis for DOE.

| Predictor | Coef   | SE Coef | T   | P-Value |
|-----------|--------|---------|-----|---------|
| Constant  | 4176.76| 15.09   | 276.86 | 0.000   |
| P         | 1.4334 | 0.4774  | 3.00 | 0.004   |
| C         | 112.33 | 31.82   | 3.53 | 0.001   |
| M         | −112.6 | 116.2   | −0.97 | 0.336   |
| P²        | −0.001759 | 0.004892 | −0.36 | 0.720   |
| C²        | −78.19 | 21.74   | −3.50 | 0.001   |
| M²        | 384.8  | 305.8   | 1.26 | 0.212   |
| P*C      | −0.0775 | 0.4217  | −0.18 | 0.855   |
| P*M       | −0.260 | 2.288   | −0.11 | 0.910   |
| C*M       | −57.9  | 152.5   | −0.38 | 0.706   |
| P*C*M     | 0.868  | 3.531   | 0.25 | 0.807   |

The experimental factors

| Factors | Level |
|---------|-------|
| P       | 1 2 3 |
| C       | 30 40 60 |
| M       | 0.3 0.6 0.9 |

The values in Table 2 are rounded to the nearest integer. The result of regression analysis for DOE.
In the fitness evaluation, the chromosome with the highest output (e.g. throughput) earns the highest fitness, and the chromosome with the lowest output earns the lowest fitness. This term can be obtained in the following form:

$$F_c = O_c - O_{c,\text{lowest}} \quad for \ c = 1, 2, \ldots, C$$  \hspace{1cm} (5)
To select pairs of parents for the next generation, the roulette-wheel principle is adopted by considering the probability that the $c$th chromosome would be chosen ($P_c$), as presented in equation 6 (Yang et al., 2007). Therefore, chromosomes with a high fitness value have a high chance of being selected as the next generation’s parents.

$$P_c = \frac{F_c}{\sum_{c=1}^{C} F_c} \quad (6)$$

We use crossover rate and mutation rate to determine which chromosomes need to undergo crossover and mutation. Table 4 shows a two-crossover point in this study. The stopping criterion in genetic algorithm is achieved if there is no significant improvement from the previous best so far (gap < 0.05) or when the results converge.

### 4. Results

This study compares three solution approaches, genetic algorithm, particle swarm optimization, and resource leveling. Similar to GA, PSO evaluates multiple solutions (particles) at a time. Particles in PSO keep track of their best performances so far as their personal best (Dong & Zhou, 2017; Gao, Cao, et al., 2019). The simulation approaches of GA and PSO had the simultaneous advantages of meta-heuristic and discrete event simulation. The results of resource leveling from a previous study (Hsu, 2006) are used for comparison. We use two different kinds of problem to determine the performance of approaches for solving the problem, as follows:

1. **Practical problem**

   This model’s problem is based on an existing system in the Chemical Vapor Deposition (CVD) location in a semiconductor manufacturing system. Table 5 presents the tool group data. For comparison, we use five performance indicators to identify the goodness of the approaches for a 195-day (6.5 months) planning horizon with 30 replications. MANOVA (Multivariate Analysis Of Variance) is used to verify the approaches’ significance because there are five performance indicators.

   Table 6 shows the example result of MANOVA testing in Minitab 14.0, where GA-DES performs better than PSO-DES since the p-value less than 0.05.

   Table 7 presents a summary of all MANOVA test. It shows that GA-DES performs better than other approaches in the case of minimizing manpower. Nevertheless, under the objective of maximizing throughput, GA-DES and PSO-DES are not statistically

| Crossover | $X_{111}$ | $X_{112}$ | $X_{113}$ | $X_{114}$ | ... | $X_{30,2,13}$ | $X_{30,2,14}$ | $X_{ij}$ |
|-----------|----------|----------|----------|----------|-----|----------------|----------------|--------|
| Parent 1  | $X_{111}$ | $X_{112}$ | $X_{113}$ | $X_{114}$ | ... | $X_{30,2,13}$ | $X_{30,2,14}$ | $X_{ij}$ |
| Parent 2  | $X_{111}$ | $X_{112}$ | $X_{113}$ | $X_{114}$ | ... | $X_{30,2,13}$ | $X_{30,2,14}$ | $X_{ij}$ |
| Random    | $X_{111}$ | $X_{112}$ | $X_{113}$ | $X_{114}$ | ... | $X_{30,2,13}$ | $X_{30,2,14}$ | $X_{ij}$ |
| Crossover point 1 | $X_{111}$ | $X_{112}$ | $X_{113}$ | $X_{114}$ | ... | $X_{30,2,13}$ | $X_{30,2,14}$ | $X_{ij}$ |
| Crossover point 2 | $X_{111}$ | $X_{112}$ | $X_{113}$ | $X_{114}$ | ... | $X_{30,2,13}$ | $X_{30,2,14}$ | $X_{ij}$ |

(Two-point crossover operation)
different at the $p = 0.05$ level. However, both are better than the existing reference and resource leveling (Hsu, 2006). In terms of hiring manpower, GA-DES and PSO-DES are $11.76 - 17.65\%$ less than the reference.

2. Additional problems or Scenarios

Table 8 shows five additional scenarios based on artificial perturbations of the real data in the previous section.

Based on PM’s best start time for each approach in solving different scenarios, 30 replications of the simulation are performed, as summarized in Table 9. The maximum number of manpower is associated with the hiring level. Most of the results show that the hiring level of GA-DES and PSO-DES are similar. Nevertheless, the hiring level of GA-DES and PSO-DES can be up to 15.38\% less than resource leveling, except for scenario 2.

Because there are multiple outputs, we then use MANOVA to evaluate the performance of three problem-solving approaches. GA-DES and PSO-DES perform better than resource leveling for all scenarios. However, there is not enough statistical evidence at the 0.05 level to conclude that GA-DES performs differently from PSO-DES. In general, the output model based on the PM’s start time of GA-DES are statistically equal to the output model based on the PM’s start time of PSO-DES.

Table 5. The tool group data.

| No | Tool Group Name | No of Machines | No of Chambers | P.1  | P.2  | P.3  |
|----|-----------------|----------------|----------------|------|------|------|
| 1  | Tool Group 1    | 3              | 2              | 77.2972 | 82.3647 | 75.5038 |
| 2  | Tool Group 2    | 6              | 1              | 41.0695 |        |      |
| 3  | Tool Group 3    | 3              | 1              | 13.9156 |        |      |
| 4  | Tool Group 4    | 3              | 3              | 284.4216 |        |      |
| 5  | Tool Group 5    | 1              | 1              | 27.3516  |        |      |
| 6  | Tool Group 6    | 7              | 1              | 45.1765  | 41.0695 |      |
| 7  | Tool Group 7    | 1              | 1              | 88.8000  |        |      |
| 8  | Tool Group 8    | 2              | 2              | 85.9174  |        |      |
| 9  | Tool Group 9    | 2              | 3              | 181.6186 |        |      |
| 10 | Tool Group 10   | 2              | 2              | 98.3518  |        |      |

Table 6. An example of MANOVA test.

| Objective: Minimize manpower |
|-------------------------------|
| MANOVA for Method (PSO-DES vs GA-DES) |
| $s = 1$ $m = 0.0$ $n = 27.5$ |

| Test      | Statistic | F    | Num | Denom | P      |
|-----------|-----------|------|-----|-------|--------|
| Wilks’    | 0.80225   | 7.025| 2   | 57    | 0.002  |
| Lawley-Hotelling | 0.24649 | 7.025 | 2 | 57 | 0.002 |
| Pillai’s  | 0.19775   | 7.025| 2   | 57    | 0.002  |
| Roy’s     | 0.24649   |      |     |       |        |
Table 7. Comparison of the approaches performances.

| Performances | PSO-DES | GA-DES | Resource Leveling |
|--------------|---------|--------|------------------|
|              | Max: Throughput | Min: Manpower | Max: Throughput | Min: Manpower | Min: Manpower | Average | Std | Average | Std | Average | Std |
| Objectives   | Average | Std  | Average | Std  | Average | Std | Average | Std | Average | Std |
| Throughput   | 44,872  | 76   | 44,214  | 43   | 43,768  | 53   | 43,608  | 113  |
| WIP          | 363     | 78   | 356     | 76   | 527     | 93   | 587     | 103  |
| Cycle Time (Days) | | | | | | | | |
| Product 1    | 3.305   | 0.159 | 3.358   | 0.173 | 3.339   | 0.167 | 3.760   | 0.105 |
| Product 2    | 1.329   | 0.062 | 1.358   | 0.063 | 2.189   | 0.066 | 2.470   | 0.063 |
| Product 3    | 0.921   | 0.093 | 0.927   | 0.069 | 0.894   | 0.087 | 0.883   | 0.045 |
| Product 4    | 1.870   | 0.176 | 1.891   | 0.168 | 1.741   | 0.175 | 1.779   | 0.135 |
| Product 5    | 2.544   | 0.210 | 2.566   | 0.279 | 2.397   | 0.179 | 2.449   | 0.235 |
| Product 6    | 2.248   | 0.217 | 2.241   | 0.196 | 2.111   | 0.195 | 2.138   | 0.138 |
| Product 7    | 1.296   | 0.130 | 1.305   | 0.153 | 1.220   | 0.110 | 1.208   | 0.139 |
| Product 8    | 1.356   | 0.119 | 1.379   | 0.085 | 1.341   | 0.120 | 1.343   | 0.108 |
| Product 9    | 2.164   | 0.193 | 2.254   | 0.223 | 2.080   | 0.240 | 2.184   | 0.188 |
| Product 10   | 1.994   | 0.180 | 1.981   | 0.192 | 1.945   | 0.158 | 1.732   | 0.708 |
| Product 11   | 1.825   | 0.158 | 1.858   | 0.191 | 1.715   | 0.161 | 1.523   | 0.629 |
| Product 12   | 0.880   | 0.273 | 0.965   | 0.210 | 1.030   | 0.298 | 0.829   | 0.406 |
| Equipment Utilization (Percentage) | | | | | |
| Tool Group 1 | 43.91   | 0.034 | 43.58   | 0.139 | 43.91   | 0.033 | 43.70   | 0.019 |
| Tool Group 2 | 14.78   | 0.005 | 14.68   | 0.047 | 14.79   | 0.005 | 14.71   | 0.006 |
| Tool Group 3 | 26.58   | 0.020 | 26.38   | 0.086 | 26.58   | 0.017 | 26.45   | 0.010 |
| Tool Group 4 | 31.52   | 0.000 | 31.29   | 0.101 | 31.52   | 0.000 | 31.37   | 0.000 |
| Tool Group 5 | 14.79   | 0.000 | 14.68   | 0.047 | 14.79   | 0.000 | 14.72   | 0.000 |
| Tool Group 6 | 13.22   | 0.016 | 13.12   | 0.043 | 13.22   | 0.014 | 13.16   | 0.008 |
| Tool Group 7 | 45.80   | 0.080 | 45.45   | 0.149 | 45.80   | 0.075 | 45.59   | 0.046 |
| Tool Group 8 | 35.33   | 0.060 | 35.06   | 0.116 | 35.33   | 0.057 | 35.17   | 0.026 |
| Tool Group 9 | 24.74   | 0.000 | 24.56   | 0.079 | 24.74   | 0.000 | 24.62   | 0.000 |
| Tool Group 10| 27.06   | 0.013 | 26.86   | 0.087 | 27.06   | 0.010 | 26.93   | 0.010 |
| Manpower     | 15      | 14    | 15      | 14    | 20      | 15    | 17     | 17   |

Statistical Analysis: MANOVA
(Multivariate analysis of variance)
Obj: Minimize Manpower    Obj: Maximize Throughput
GA-DES, PSO-DES, Resource GA-DES, PSO-DES > Reference
Leveling > Reference
GA-DES, PSO-DES > Resource Leveling, DES = PSO-DES
GA-DES > PSO-DES
*Note:
> > means better than
'=' means there is not enough evidence at the 0.05 level to conclude that methods are different each other
5. Conclusions and future research

5.1. Conclusions

Preventive maintenance scheduling in the semiconductor industry is a complex problem where many independent variables are simultaneously involved. Many studies in the literature address preventive maintenance scheduling problem (PMSP). However, most of the existing studies use simple cases of PMSP in terms of number of tools involved, number of PM tasks, assumption of PM duration, and scheduling horizon as well as in

Table 8. The scenario of problem scale.

| Scenario | Tool | Chamber | PM Task | Planning Horizon |
|----------|------|---------|---------|------------------|
| Scenario 1 | 10   | 4       | 10      | 1 month          |
| Scenario 2 | 15   | 3       | 15      | 2 month          |
| Scenario 3 | 20   | 4       | 10      | 3 month          |
| Scenario 4 | 25   | 3       | 10      | 2 month          |
| Scenario 5 | 30   | 1–3     | 15      | 1 month          |

Table 9. The results of scenarios.

| Objective       | Scenario | Method       | Average | Std | Average | Std | Max | MANOVA (Multivariate analysis of variance) |
|-----------------|----------|--------------|---------|-----|---------|-----|-----|------------------------------------------|
| Minimize Manpower | Scenario 1 | Resource leveling | 3,777.50 | 130.04 | 5.83 | 0.75 | 8 | GA-DES, PSO-DES > Resource leveling |
|                 |          | GA-DES       | 3,831.67 | 135.18 | 5.90 | 0.71 | 8 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 3,768.33 | 178.59 | 5.77 | 0.68 | 7 | GA-DES, PSO-DES > Resource leveling |
|                 | Scenario 2 | Resource leveling | 9,373.33 | 98.47 | 7.50 | 0.78 | 9 | GA-DES, PSO-DES > Resource leveling |
|                 |          | GA-DES       | 9,405.00 | 118.79 | 7.80 | 0.76 | 9 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 9,435.00 | 135.44 | 7.83 | 0.65 | 9 | GA-DES, PSO-DES > Resource leveling |
|                 | Scenario 3 | Resource leveling | 14,768.33 | 149.41 | 9.10 | 1.30 | 12 | GA-DES, PSO-DES > Resource leveling |
|                 |          | GA-DES       | 14,772.50 | 146.27 | 8.90 | 1.06 | 11 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 14,760.83 | 148.66 | 8.80 | 1.10 | 11 | GA-DES, PSO-DES > Resource leveling |
|                 | Scenario 4 | Resource leveling | 10,008.33 | 80.23 | 9.03 | 1.22 | 12 | GA-DES, PSO-DES > Resource leveling |
|                 |          | GA-DES       | 10,012.50 | 63.23 | 8.37 | 1.16 | 11 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 10,023.33 | 67.25 | 8.57 | 1.14 | 11 | GA-DES, PSO-DES > Resource leveling |
|                 | Scenario 5 | Resource leveling | 4,350.00 | 62.97 | 9.10 | 1.40 | 13 | GA-DES, PSO-DES > Resource leveling |
|                 |          | GA-DES       | 4,355.00 | 53.90 | 8.77 | 1.43 | 11 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 4,355.83 | 39.22 | 8.37 | 1.63 | 11 | GA-DES, PSO-DES > Resource leveling |

Maximize Throughput

| Objective       | Scenario 1 | Method       | Average | Std | Average | Std | Max | MANOVA (Multivariate analysis of variance) |
|-----------------|----------|--------------|---------|-----|---------|-----|-----|------------------------------------------|
|                 | Scenario 1 | GA-DES       | 3,833.33 | 137.61 | 5.97 | 0.76 | 8 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 3,825.00 | 134.71 | 6.00 | 0.79 | 8 | GA-DES, PSO-DES > Resource leveling |
|                 | Scenario 2 | GA-DES       | 9,410.00 | 108.40 | 7.73 | 1.08 | 10 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 9,439.17 | 114.97 | 7.90 | 0.88 | 10 | GA-DES, PSO-DES > Resource leveling |
|                 | Scenario 3 | GA-DES       | 14,830.00 | 117.70 | 9.17 | 1.02 | 11 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 14,803.33 | 150.96 | 9.53 | 1.01 | 11 | GA-DES, PSO-DES > Resource leveling |
|                 | Scenario 4 | GA-DES       | 10,025.83 | 72.96 | 7.97 | 1.30 | 11 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 10,024.17 | 68.70 | 8.23 | 1.10 | 11 | GA-DES, PSO-DES > Resource leveling |
|                 | Scenario 5 | GA-DES       | 4,364.17 | 69.07 | 8.67 | 1.35 | 11 | GA-DES ≡ PSO-DES |
|                 |          | PSO-DES      | 4,360.00 | 71.20 | 8.67 | 1.40 | 11 | GA-DES, PSO-DES > Resource leveling |

*Note:
* > * means better than

== means there is not enough evidence at the P = 0.05 level to conclude that methods are different each other

5. Conclusions and future research

5.1. Conclusions

Preventive maintenance scheduling in the semiconductor industry is a complex problem where many independent variables are simultaneously involved. Many studies in the literature address preventive maintenance scheduling problem (PMSP). However, most of the existing studies use simple cases of PMSP in terms of number of tools involved, number of PM tasks, assumption of PM duration, and scheduling horizon as well as in
modelling a real system. In this study, we use a real-world problem in the wafer fab. Re-entrant process, which is a common phenomenon in the wafer fab, makes problem becomes more challenging to solve.

We propose a framework for solving preventive maintenance scheduling, mainly to find the best start time of PM within a PM window in order to minimize production losses due to maintenance activities. We use discrete event simulation to mimic a real complex system in the wafer fab and embed it in metaheuristics (e.g. GA, PSO) algorithm to get the near-optimum results. Based on the PM’s start time of the proposed approach, we collect system output such as throughput, WIP, cycle time, equipment utilization and the level of manpower.

The results show that combining metaheuristics and discrete event simulation performs better than resource leveling and the baseline. Although computation time is still too long, this study provides a promising approach for solving real complex systems by utilizing metaheuristics and discrete event simulation for optimization.

### 5.2. Future research

We acknowledge some limitations of this paper and suggest a few points to extend this study. Embedding discrete event simulation in metaheuristics algorithm for solving a real complex system is time-consuming. Therefore, designing a more efficient approximation or analytical method would certainly represent an important research avenue. The use of technology for intelligent maintenance through data-driven based fault prognosis will become an important research direction (Zhong et al., 2020). In this study, we use First In First Out (FIFO) as the dispatching rule. Hence further study could make the dispatching rule as independent variables. Including maintenance cost to determine PM’s efficiency and considering safety stock policy due to maintenance activities are also the potential research extension.

### Disclosure statement

No potential conflict of interest was reported by the author(s).

### ORCID

Agus Darmawan [http://orcid.org/0000-0003-1291-1680](http://orcid.org/0000-0003-1291-1680)

### References

Assid, M., Gharbi, A., & Hajji, A. (2015). Joint production, setup and preventive maintenance policies of unreliable two-product manufacturing systems. *International Journal of Production Research*, 53(15), 4668–4683. [https://doi.org/10.1080/00207543.2015.1030468](https://doi.org/10.1080/00207543.2015.1030468)

Cassady, C. R., & Kutanoglu, E. (2005). Integrating preventive maintenance planning and production scheduling for a single machine. *IEEE Transaction on Semiconductor Manufacturing*, 54(2), 304–309. [https://doi.org/10.1109/TR.2005.845967](https://doi.org/10.1109/TR.2005.845967)

Chang, C. C. (2018). Optimal age replacement scheduling for a random work system with random lead time. *International Journal of Production Research*, 56(16), 5511–5521. [https://doi.org/10.1080/00207543.2018.1425017](https://doi.org/10.1080/00207543.2018.1425017)
Charles, A. S., Floru, I.-R., Azzaro-Pantel, C., Pibouleau, L., & Domenech, S. (2003). Optimization of preventive maintenance strategies in multipurpose batch plant: Application to semiconductor manufacturing. Computers & Chemical Engineering, 27(4), 449–467. https://doi.org/10.1016/S0098-1354(02)00216-8

Cheng, G. Q., Zhou, B. H., & Li, L. (2018). Integrated production, quality control and condition based maintenance for imperfect production systems. Reliability Engineering and System Safety, 175(7), 251–264. https://doi.org/10.1016/j.ress.2018.03.025

Crespo Marquez, A., Gupta, J. N., & Ignizio, J. P. (2006). Improving preventive maintenance scheduling in semiconductor fabrication facilities. Production Planning & Control, 17(7), 742–754. https://doi.org/10.1080/09537280600901525

De Jonge, B., & Scarf, P. A. (2020). A review on maintenance optimization. European Journal of Operational Research, 285(3), 805–824. https://doi.org/10.1016/j.ejor.2019.09.047

Dong, W., & Zhou, M. (2017). A supervised learning and control method to improve particle swarm optimization algorithms. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47(7), 1135–1148. https://doi.org/10.1109/TSMC.2016.2560128

Franciosi, C., Voisin, A., Miranda, S., Riemma, S., & Lung, B. (2020). Measuring maintenance impacts on sustainability of manufacturing industries: From a systematic literature review to a framework proposal. Journal of Cleaner Production, 260(19), 1–19. https://doi.org/10.1016/j.jclepro.2020.121065

Gao, K. Z., Cao, Z. G., Zhang, L., Chen, Z. H., Han, Y. Y., & Pan, Q. K. (2019). A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems. IEEE/CAA Journal of Automatica Sinica, 6(4), 904–916. https://doi.org/10.1109/JAS.2019.1911540

Gao, S., Zhou, M., Wang, Y., Cheng, J., Yachi, H., & Wang, J. (2019). Dendritic neuron model with effective learning algorithms for classification, approximation, and prediction. IEEE Transactions on Neural Networks and Learning Systems, 30(2), 601–614. https://doi.org/10.1109/TNNLS.2018.2846646

Ghahramani, M., Qiao, Y., Zhou, M., O’Hagan, A., & Sweeney, J. (2020). AI-based modeling and data-driven evaluation for smart manufacturing processes. IEEE/CAA Journal of Automatica Sinica, 7(4), 1026–1037. https://doi.org/10.1109/JAS.2020.1003114

Gurel, S., & Akturk, M. S. (2008). Scheduling preventive maintenance on a single CNC machine. International Journal of Production Research, 46(24), 6797–6821. https://doi.org/10.1080/00207540701487833

Hsu, F. 2006. Establishment of manpower standard for equipment preventive maintenance. Master Thesis, Dept. of Industrial Engineering & Engineering Management, National Tsing Hua University.

Li, L., Ambani, S., & Ni, J. (2009). Plant-level maintenance decision support system for throughput improvement. International Journal of Production Research, 47(24), 7047–7061. https://doi.org/10.1080/00207540802375705

Liao, W., Zhang, W., & Jiang, M. (2017). Multi-objective group scheduling optimization integrated with preventive maintenance. Engineering Optimization, 49(11), 1890–1904. https://doi.org/10.1080/0305215X.2017.1280258

Marmier, F., Varnier, C., & Zerhouni, N. (2009). Proactive, dynamic and multi-criteria scheduling of maintenance activities. International Journal of Production Research, 47(8), 2185–2201. https://doi.org/10.1080/00207540802311106

Moghaddam, K. S., & Usher, J. S. (2011). Sensitivity analysis and comparison of algorithms in preventive maintenance and replacement scheduling optimization models. Computers & Industrial Engineering, 61(1), 64–75. https://doi.org/10.1016/j.cie.2011.02.012

Montoya-Torres, J. R. (2006). Manufacturing performance evaluation in wafer semiconductor factories. International Journal of Productivity and Performance Management, 55(3/4), 300–310. https://doi.org/10.1108/17410400610653246

Mosley, S. A., Teyner, T., & Uzsoy, R. M. (1998). Maintenance scheduling and staffing policies in a wafer fabrication facility. IEEE Transactions on Semiconductor Manufacturing, 11(2), 316–323. https://doi.org/10.1109/66.670182
Panagiotidou, S., & Tagaras, G. (2008). Evaluation of maintenance policies for equipment subject to quality shifts and failures. *International Journal of Production Research, 46*(20), 5761–5779. https://doi.org/10.1080/00207540601182260

Pongcharoen, P., Hicks, C., Braiden, P. M., & Stewardson, D. J. (2002). Determining optimum genetic algorithm parameters for scheduling the manufacturing and assembly of complex products. *International Journal of Production Economics, 78*(3), 311–322. https://doi.org/10.1016/S0925-5273(02)00104-4

Qiao, F., Ma, Y., Zhou, M., & Wu, Q. (2020). A novel rescheduling method for dynamic semiconductor manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 50*(5), 1679–1689. https://doi.org/10.1109/TSMC.2017.2782009

Renna, P. (2012). Influence of maintenance policies on multi-stage manufacturing systems in dynamic conditions. *International Journal of Production Research, 50*(2), 345–357. https://doi.org/10.1080/00207543.2010.537703

Safaei, N., Banjevic, D., & Jardine, A. K. S. (2010). Impact of the use-based maintenance policy on the performance of cellular manufacturing systems. *International Journal of Production Research, 48*(8), 2233–2260. https://doi.org/10.1080/00207540802710273

Sheu, D. D., & Kuo, J. Y. (2006). A model for preventive maintenance operations and forecasting. *Journal of Intelligent Manufacturing, 17*(4), 441–451. https://doi.org/10.1007/s10845-005-0017-6

Sheu, D. D., & Wu, J. (1999). Benchmarking of Equipment Management for Taiwan’s Semiconductor Fabrications Plants. NSC88-2213-E-007-037.

Suryadi, H., & Papageorgiou, L. G. (2004). Optimal maintenance planning and crew allocation for multipurpose batch plant. *International Journal of Production Research, 42*(2), 355–377. https://doi.org/10.1080/00207540310001602955

Wang, J., Hu, H., Pan, C., Zhou, Y., & Li, L. (2020). Scheduling dual-arm cluster tools with multiple wafer types and residency time constraints. *IEEE/CAA Journal of Automatica Sinica, 7*(3), 776–789. https://doi.org/10.1109/JAS.2020.1003150

Wang, J., & Kumbasar, T. (2019). Parameter optimization of interval Type-2 fuzzy neural networks based on PSO and BBBC methods. *IEEE/CAA Journal of Automatica Sinica, 6*(1), 247–257. https://doi.org/10.1109/JAS.2019.1911348

Wang, J., Zhou, M., Guo, X., & Qi, L. (2019). Multi-period asset allocation considering dynamic loss aversion behavior of investors. *IEEE Transactions on Computational Social Systems, 6*(1), 73–81. https://doi.org/10.1109/TCSS.2018.2883764

Xiao, L., Song, S., Chen, X., & Coit, D. W. (2016). Joint optimization of production scheduling and machine group preventive maintenance. *Reliability Engineering and System Safety, 146*(2), 68–78. https://doi.org/10.1016/j.ress.2015.10.013

Yang, T., Kuo, Y., & Cho, C. (2007). A genetic algorithms simulation approach for multi-attribute combinatorial dispatching decision problem. *European Journal of Operational Research, 176*(3), 1859–1873. https://doi.org/10.1016/j.ejor.2005.10.048

Yao, X. D., Fernandez, G. E., Fu, M. C., & Marcus, S. I. (2007). Optimal preventive maintenance scheduling in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing, 17*(3), 345–356. https://doi.org/10.1109/TSM.2004.831948

Zhang, P., Shu, S., & Zhou, M. (2018). An online fault detection model and strategies based on SVM-grid in clouds. *IEEE/CAA Journal of Automatica Sinica, 5*(2), 445–456. https://doi.org/10.1109/JAS.2017.7510817

Zhong, K., Han, M., & Han, B. (2020). Data-driven based fault prognosis for industrial systems: A concise overview. *IEEE/CAA Journal of Automatica Sinica, 7*(2), 330–345. https://doi.org/10.1109/JAS.2019.1911804