*Proceedings of the 2016 Winter Simulation Conference*
*T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.*

# SCHEDULING PREVENTIVE MAINTENANCE WITHIN A QUEUE TIME FOR MAXIMUM THROUGHPUT IN SEMICONDUCTOR MANUFACTURING

Adar A. Kalir

Fab/Sort Manufacturing Division
Intel Corporation
2 HaZoran ST.
Qiriat Gat, ISRAEL

Israel Tirkel

Department of Industrial Engineering and Management
Ben-Gurion University
Beer-Sheva, ISRAEL

## ABSTRACT

We address the PM-QT problem, of scheduling preventive maintenance (PM) activities on tools within queue time (QT) restrictions, such that the overall throughput is maximized and the QT restrictions are not violated. Despite the increased occurrences of QT in semiconductor manufacturing, this problem has not been explicitly addressed. We formulate this problem as a mixed integer linear program (MILP) and propose a cross-entropy (CE) heuristic approach for its efficient solution. We show that the CE solutions are indeed efficient in runtime reductions with almost no compromise of the solution's quality (less than 1% difference between MILP and CE solutions for large scale problems).

## 1 INTRODUCTION

The Semiconductor manufacturing process is one of the most complex processes in industrial use today (Mosley *et al*. 1998). An important factor in streamlining the manufacturing process is the equipment downtime. Therefore, maintaining high equipment availability has been regarded as one of the major goals of the industry. Fabrication equipment is highly sophisticated and expensive, thus it is subject to unpredictable failures and requires periodical preventive maintenance (PM) activities and calibration. In order to maximize the profits of fab operation, the PM plan has to be scheduled carefully (Yao *et al*. 2004) and should take into consideration various factors, including queue time (QT) restrictions.

QT restrictions are time limitations placed between two or more operation steps in process flow of wafer fabrication, as illustrated in Figure 1. These restrictions are used in order to limit the degradation of wafers' quality while they are in sensitive processing states. Failure to process lots through the specified steps within the allotted time would result in rework or even in scrap of the damaged lots (Burda, 2008). Typically, the steps with the QT restriction contain some of the most expensive toolsets which are also, by design, the bottlenecks of the fab.

This research is concerned with the problem of scheduling PM activities, in such a manner that throughput is maximized and QT restrictions are not violated. Despite a large number of studies that deal with the problem of PM scheduling in the semiconductor industry (e.g. Dijkhuizen and Hareten 1997; Yao *et al*. 2001 and 2004; Crespo Marquez *et al*. 2006), there is little research addressing the problem of scheduling PM activities within a QT (Altman and Kalir 2009).

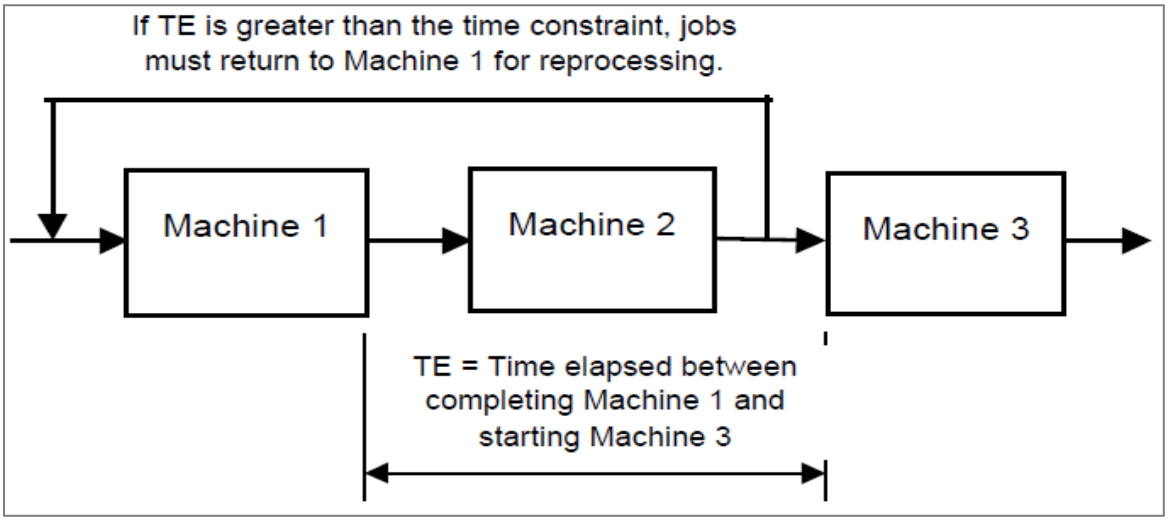



Figure 1: Sample diagram of a QT system (Robinson 1998).

This paper is organized as follows. We first define the PM-QT scheduling problem, including the model inputs and outputs. Then, the problem is formulated as a mixed integer linear program (MILP) model. Two solution approaches are proposed next. The goal of the first solution approach is to obtain an optimal solution for small scale problems, whereas the goal of the second solution approach is to obtain a near-optimal solution for larger scale problems. A cross entropy (CE) method is deployed for the heuristic solution. The CE algorithm is then tested using Visual Basic Application (VBA) and LINDO API software. A numerical study is presented which consists of two main phases. The first phase contains sensitivity analysis of the CE algorithm parameters, based on which the algorithm parameters are calibrated. The second phase tests and compares both solution approaches using three different scale problems. Finally, conclusions are offered based on the study's results that the optimization model can be utilized as an effective PM scheduling tool for problems within a planning horizon of up to a full week (168 hours) and up to 10 PMs, as opposed to the heuristic solution approach that can be utilized for larger scale problems.

## 2 LITERATURE REVIEW

The production planning and scheduling of a wafer fabrication process is a complex process. Uzsoy *et al.* (1992) list several factors that make it so complex, such as: long and reentrant product flows, uncertain yields, diverse equipment characteristics, critical queue time window restrictions, significant equipment downtime, production and development on shared toolsets.

The reliability and availability of equipment in semiconductor manufacturing fabs has become an important issue for yield improvement (Yao *et al*. 2004) as well as for factory capacity and performance in terms of outputs, inventory, cycle time and velocity. Dijkhuizen and Hareten (1997) and Yao *et al*. (2004) offer a hierarchical structure for planning and scheduling of PM activities, as illustrated in Figure 2. The first (long term) stage is concerned with optimal PM planning for long-term horizon. A finite interval $(t, t + \Delta t)$ is determined during which a PM activity must be carried out. Crespo-Marquez *et al*. (2006) suggest that the design of a PM plan needs to take into account factors, such as: production plan, tool's failure dynamics, operating conditions of the process, and consequences of various maintenance activities regarding investments in instrumentation, diagnostic, and repair tools. For solving this problem they propose a semi-Markovian decision process (SMDP) approach. Alardhi *et al*. (2007) suggest that the binary nature of maintenance scheduling problems makes them adequate for integer optimization. They present a binary integer program model designed to produce an optimal maintenance schedule for cogeneration plants in terms of maximizing the available number of units in each plant for a twelve-month demand cycle.

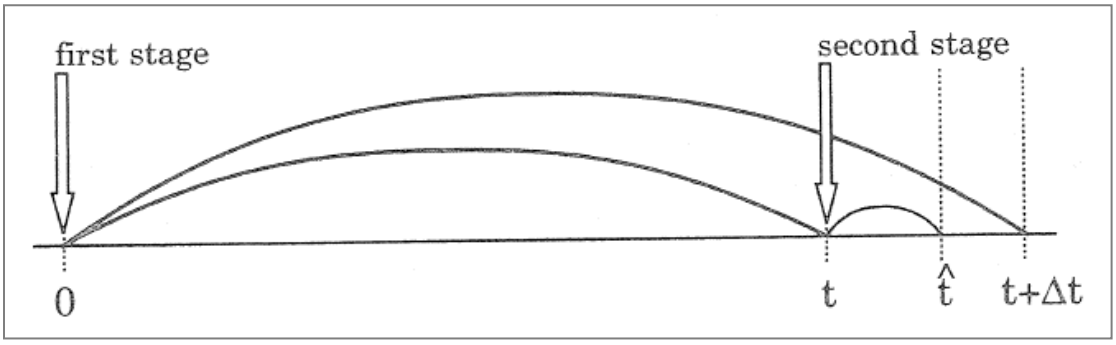



Figure 2: Hierarchical approach to PM planning and scheduling (Dijkhuizen and Hareten 1997).

The second (short term) stage is deployed once the optimal PM plan has been set. The purpose in this stage is to determine the optimal starting time t for a specific PM activity to commence within the interval that was determined in the first stage. For this problem, there is a need to consider the following: PM plan, status of production (i.e. work in process, WIP), tool operating condition, possible functional dependencies between tools and tool components; interdependence among PM tasks; and recourse constraints (e.g. headcount of maintenance technicians). In order to solve this, Yao *et al*. (2004) develop mixed integer program (MIP) models for scheduling all PM activities of a toolset over a planning horizon. Their models incorporate interdependence among different PM tasks, production planning data (e.g. projected WIP levels), manpower constraints, and associated PM time windows and costs. However, the models have a planning horizon shorter than the time between two successive PM activities.

Over the past decade, a variety of optimization and heuristic techniques were used for different types of PM planning and scheduling problems in different industrial settings. Raza and Al-Turki (2007) examine the effectiveness of simulated annealing (SA) and tabu search heuristics in solving the problem of scheduling maintenance activities on a single machine. They show that a hybrid tabu search and SA algorithm heuristic can be effective in reaching near optimal solutions with reduced computation time. Moghaddam and Usher (2009, 2011) also apply SA to solve a multi-objective optimization model for determining PM and replacement schedules for repairable and maintainable series of system components. In their optimization model, the planning horizon is segmented into discrete and equally sized periods. Three possible actions for each component are defined (maintenance, replacement, or do nothing) are considered within each period. Total costs and overall reliability of the system are considered as the objective functions. Suresh and Kumarappan (2006) develop an optimization model and use a combination of genetic algorithm with simulated annealing. They apply their method to determine the PM schedule in a power system.

One of the more recent heuristic techniques is known as the Cross Entropy (CE) method. The CE method, pioneered by Rubinstein (1997), was motivated by an adaptive algorithm for estimating probabilities of rare events in complex stochastic networks. He realized that with a simple modification, the method could be used not only for estimating probabilities of rare events but also for solving difficult combinatorial optimization problems. This is done by translating the original "deterministic" optimization problem into a related "stochastic" estimation problem and then applying the rare-event simulation mechanism. The CE method involves an iterative procedure where each iteration can be broken down into two phases:

- Phase I: Generate a random data sample (trajectories, vectors, etc.) according to a specified mechanism.
- Phase II: Update the parameters of the random mechanism based on the data, to produce a "better" sample in the next iteration.

The principal outcome of this approach is the construction of a random sequence of solutions which converges probabilistically to a near-optimal solution for the problem. Several recent applications of combinatorial optimization problems in general and of scheduling problems in particular, demonstrate the effectiveness of the CE method as a practical means for solving NP-hard problems. For example, Margolin (2002) examines the CE method on the single machine total weighted tardiness problem (SMTWP) and on single machine common due date problem (SMCDDP) to show that in both of these scheduling problems the random sequence of solutions truly converges to a near-optimal solution.

When QT restrictions are imposed on a sequence of operations, it requires careful and comprehensive coordination between PM tasks and incoming WIP, in order to minimize the damage caused to production flow and equipment productivity. Thus, in order to optimize the QT segment performance, PM tasks on each toolset are scheduled to optimize their availability while also considering the QT restrictions (Altman and Kalir 2009). Prior research has focused primarily on QT segment capacity, QT WIP scheduling and trade-offs between yield and cycle time and/or output. Burda (2008) provides a mathematical optimization with success to schedule WIP for some of the cases. Van Sickle and Hertzler (2006) utilize simulation to examine the trade-offs among time, capacity and quality impacts associated with QT limits. Other than these works and alike, there is hardly any work that addresses the explicit problem of scheduling maintenance activities over a planning horizon of toolsets within a QT. This problem is addressed in this paper and the CE method is deployed for devising a proposed solution algorithm.

## 3 PROBLEM FORMULATION

In this section, the problem addressed in this paper is defined and formulated. The following information is considered in our modeling framework:

- Operations: sequence of operation steps (route), process times, WIP at the beginning of the planning horizon for each operation step and WIP look ahead for first operation step in the sequence.
- Production tools: set of tools that may process each operation step and batch sizes.
- QT constraints: QT-sensitive sections in the sequence (start and end operations and QT limit).
- PM tasks: time windows for each tool, in which a PM activity must take place and PM durations.
- Technician (headcount) constraints: number of available technicians for performing PM's.

Only PM activities that their time windows fall within the planning horizon are considered. The length of each PM is static and independent on the schedule. There can be more than one PM for each tool within the planning horizon.

The problem is formulated as a MILP model. As mentioned earlier, a number of researchers developed MILP models for the purpose of solving PM scheduling problems (Yao *et al*. 2004; Alradni *et al*. 2007). As in Alradni *et al*. (2007), the binary nature of the maintenance scheduling problems makes it a reasonable choice. We shall use binary variables for the determination of the optimal time for each PM task to occur (within a static time window) and for the representation of the PM task duration (in which the production tool is down). The objective function represents maximum throughput and the constraints ae linear representing WIP flow, QT restrictions and PM task impacts.

### 3.1 Notation

Table 1 gives the indices, parameters, and constants used in the formulation while Table 2 lists the decision variables.

Table 1: Indices, parameters, and constants.

| Index/ Parameter | Description |
|---|---|
| $i = 1\ldots n$ | Operation index |
| $j = 1\ldots m$ | Tool index |
| $k = 1\ldots K$ | Tool group with the same maintenance scheme |
| $t = 1\ldots T$ | Time (discrete) index, where T is the planning horizon |
| $o_j = 1\ldots O_j$ | PM set of activities for each tool j |
| $E_{j,oj}$ | The earliest time window of PM $o_j$ on tool j |
| $L_{j,oj}$ | The latest time window of PM $o_j$ on tool j |
| $D_{j,oj}$ | The duration of PM $o_j$ on tool j |
| $TL_{i,r}$ | QT constraint in time units between operation *i* and operation *r* |
| $B_j$ | Batch size of tool j |
| $R_i$ | Process time of operation i |
| $S_t$ | Max planned PM in time t |
| $S_{t,k}$ | Max planned PM in time t for tool group k |
| $CW_{i,t}$ | The amount of the WIP that resides at operation *i* at time t |
| $CP_{i,t,j}$ | The output of operation *i* by the end of time $t-[Ri]^+$ (<0) from tool j |

Table 2: Decision variables.

| Decision variables | Description |
|---|---|
| $w_{i,t}$ | The amount of WIP that resides at operation i at the beginning of time t |
| $p_{i,t}$ | The output of operation i at the end of time t |
| $p_{i,t,j}$ | The output of operation i of tool j at the end of time t |
| $u_{j,t}$ | Binary variable: 1 if tool j is available at time t (zero otherwise) |
| $y_{j,oj,t}$ | Binary variable: 1 if PM $o_j$ starts at time t on tool j (zero otherwise) |

### 3.2 Model Formulation

The objective in Equation (1) is to maximize the total output of all operation steps over the planning horizon. Constraint-set (2) preserves WIP flow at any operation step and time period while constraint-set (3) limits the output from each operation step to not exceed the available WIP. Constraint-set (4) limits the total output of a toolset to the sum of its entities. Constraint-set (5) limits the output of tool j at the end of time t to its batch size (if it can be done within the time period). Note that $u_{j,t}$ is defined only for the time window of the PM, plus its duration and the operation's process time. Constraint-set (6) limits the output from tool j to the batch size during the process time from all the operation steps. Constraint-set (7) enforces the queue time restriction. Constraints (8) set PM type $o_j$ to start at time t on tool j, while Constraint-set (9) ensures that the tool would not be available during the PM. Constraints (10) and (11) limit the various PMs that are planned at the same time for all the tools and for each specific toolset, respectively. Constraints (12) set $u_{j,t}$ to 1 for each $u_{j,t}$ that does not belong to one of the PM time windows. Lastly, constraints (13) and (14) define the non-negativity and binary variables respectively.

$$Max \sum_i \sum_t p_{i,t} \tag{1}$$

$$w_{i,t} = \begin{Bmatrix} CW_{i,t} & t-1 \le 0 \\ CW_{i,t-1} - p_{i,t-1} + p_{i-1,t-1} & first\ operation \\ w_{i,t-1} - p_{i,t-1} + p_{i-1,t-1} & otherwise \end{Bmatrix} \quad \forall i,t \tag{2}$$

$$\sum_{\tau=t}^{t+R_i-1} p_{i,t} \le w_{i,t} \qquad \forall i,t \tag{3}$$

$$p_{i,t} \le \sum_j p_{i,t,j} \qquad \forall i,t \tag{4}$$

$$p_{i,t,j} \le \begin{Bmatrix} CP_{i,t,j} & t - R_i < 0 \\ B_j \cdot u_{\tau,j} \quad \forall j, o_j, t \in \left\{E_{j,o_j}, L_{j,o_j} + D_{j,o_j} + R_i\right\} & \forall \tau = t, t-1, \dots, r - R_i + 1 \end{Bmatrix} \tag{5}$$

$$\sum_{i=1}^{n} \sum_{\tau=t}^{t+R_i-1} p_{i,\tau,j} \le B_j \qquad \forall j \tag{6}$$

$$\sum_{y=i+1}^{r} w_{y,t} \le \sum_{\tau=t}^{t+TL_{i,r}+R_r-1} p_{r,\tau} \qquad \forall (i,r) \in nestedQT, t \tag{7}$$

$$\sum_{t=E_{j,o_j}}^{L_{j,o_j}} y_{j,o_j,t} = 1 \qquad \forall j, o_j \tag{8}$$

$$\sum_{\tau=t}^{t+D_{j,o_j}-1} u_{\tau,j} \le D_{j,o_j} - D_{j,o_j} \cdot y_{j,o_j,t} \quad \forall j, o_j, t \in \left\{E_{j,o_j}, L_{j,o_j}\right\} \tag{9}$$

$$\sum_{j=1}^{m} u_{t,j} \ge m - S_t \qquad \forall t \tag{10}$$

$$\sum_{j=1}^{m_k} u_{t,j} \ge m_k - S_{t,k} \qquad \forall t \tag{11}$$

$$u_{t,j} = 1 \qquad \forall t \ne \left\{E_{j,o_j}, L_{j,o_j} + D_{j,o_j} - 1\right\} \tag{12}$$

$$w_{i,t}, p_{i,t}, p_{i,t,j} \ge 0 \tag{13}$$

$$y_{j,o_j,t}, u_{t,j} \in \{0,1\} \tag{14}$$

## 4 SOLUTION APPROACH

### 4.1 The CE Algorithm

Scheduling problems are known to be NP hard even without the extra complexity emanating from the QT restriction. Therefore, a heuristic CE algorithm for solving this problem has been devised and is described next. The algorithm operates as follows:

- Initialize a probability vector
- *Generate possible trajectories (solutions)

- Obtain objective function value for each trajectory
- Obtain "elite sampling"
- Update the probabilities vector using a smoothing factor α, $p_n = \alpha p_{current} + (1-\alpha) p_{n-1}$
- Identify the best schedule in the sample
- Check for stopping criterion. If met – stop, and print the best schedule. Otherwise, go to (*).

This process is illustrated in Figure 3.

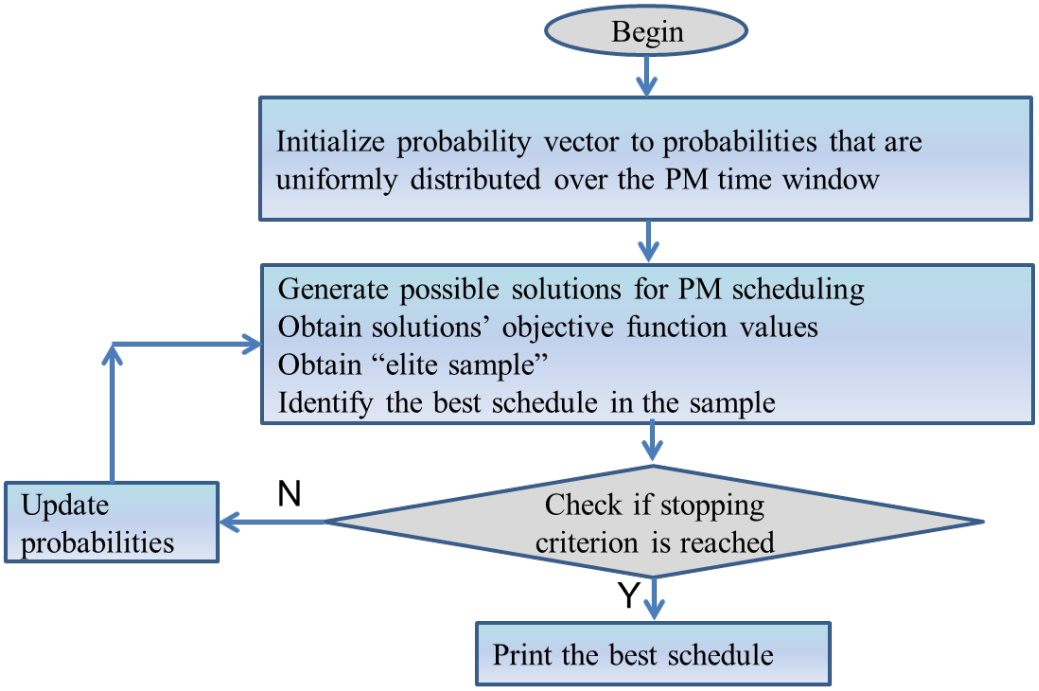



Figure 3: Heuristic solution via the CE algorithm.

The probability vector is initialized to probabilities that are uniformly distributed over the PM time windows. The probability vector defines the probability for each variable *y* within a PM time window to get the value 1 (recall that $y_{j,oj,t}$ is 1, if PM $o_j$ starts at time t on tool j). Next, a preset number of possible trajectories for PM scheduling are generated according to the probability vector. Trajectory is defined as a set of *y*'s that get a value 1 and determine each PM start time. Then, the value of the objective function is calculated for each trajectory. The values of all start times for the PM's are given as parameters to a solver, so it can obtain a solution by solving the MILP model. Once that has been completed, an "elite sampling" – i.e. best 1%, but not less than 30 samples, are obtained and the probabilities vector is updated based on the frequency of each *y* variable within the "elite sampling", using a smoothing coefficient. This process is repeated until a pre-set stopping criterion is met. The stopping criterion has been set to be: no change in the objective function for the best sample of the "elite sampling" during five consecutive iterations.

### 4.2 Parameter Calibration

Before applying the CE algorithm, the first step is to calibrate the values for the following initialization parameters:

- $\alpha$ - the smoothing factor for updating the probabilities from one generation to another. Tested for values of 0.3, 0.5 and 0.7.
- The multiplying factor: number of iterations in each generation. Tested for values 1, 3 and 5.
- "Elite sampling" size: the percentage of the iterations by which the probabilities are updated in each generation. Tested for values 1% and 5%.

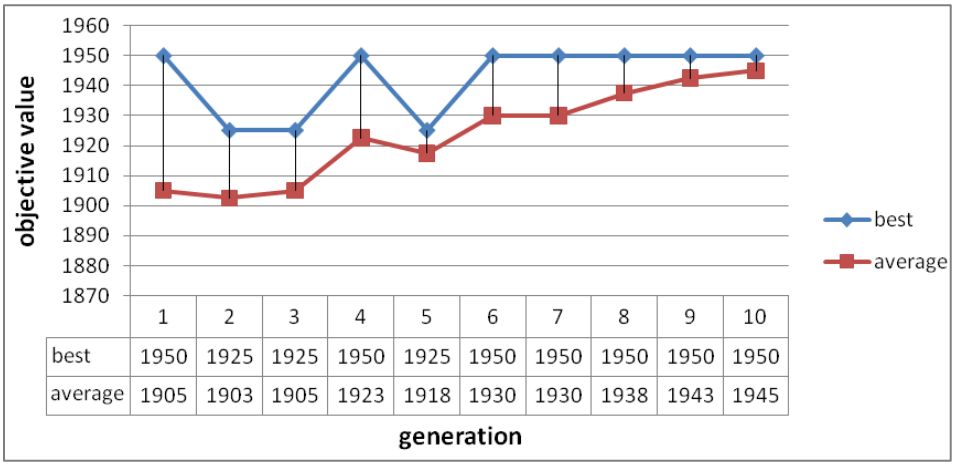
The values for the smoothing factor and the multiplying factor were selected within a relatively broad range in order to ensure that optimal solutions are attained, but this has to be verified as part of the calibration. The CE algorithm was programmed in VBA with an interface to the LINDO API solver in order to obtain the objective function value for each solution generated by the CE. In this solution approach, the solver optimizes a relaxed problem, in which, each PM start time is pre-defined.

In order to ensure that the small scale problems resemble a real production line setting, the input data includes the following characteristics: 4 operation steps, 3 toolsets with 2 tools each, 5 PM's, 2 QT restrictions and incoming WIP of 25 wafers every 2 hours for a planning horizon of 24 hours, with all tools performing at a utilization range of between 60 to 90 percent. For a subset of small scale problems, changing the parameter values had very little effect on the solution quality. With all values, the optimal solution was attained in every generation. The only difference affected by the parameters values was the run time, which obviously increased when the multiplying factor (i.e. number of iterations) was higher. However, for medium scale problems, changing the parameter values had a significant effect on the solution quality. For most problems, the optimal solution (1950) was attained only when using the values of $\alpha$=0.5 and multiplying factor =5. The optimal solution was attained with both elite sampling size of 1% and 5%, but converged and reached the solution much faster when using 1%. These results are depicted in Table 3.

Table 3: Optimality with medium scale calibration problems.

| | parameters | | | solution performance | | | |
|---|---|---|---|---|---|---|---|
| index | alfa | iteration multiplier | % elite | obj | seconds | lindo iterations | gen's to solution |
| 1 | 0.3 | 1 | 0.01 | 1900 | 57.00637 | 217656 | 5 |
| 2 | 0.5 | 1 | 0.01 | 1900 | 55.97133 | 214839 | 5 |
| 3 | 0.7 | 1 | 0.01 | 1900 | 55.49842 | 216226 | 3 |
| 4 | 0.3 | 5 | 0.01 | 1925 | 390.5691 | 1512632 | 7 |
| 5 | 0.5 | 5 | 0.01 | 1950 | 569.9406 | 2174264 | 10 |
| 6 | 0.7 | 5 | 0.01 | 1900 | 277.1378 | 1075279 | 5 |
| 7 | 0.3 | 3 | 0.01 | 1925 | 407.6167 | 1566136 | 12 |
| 8 | 0.5 | 3 | 0.01 | 1900 | 342.3899 | 1305840 | 10 |
| 9 | 0.7 | 3 | 0.01 | 1900 | 169.3499 | 649908 | 5 |
| 10 | 0.3 | 5 | 0.05 | 1925 | 1905.353 | 7396879 | 34 |
| 11 | 0.5 | 5 | 0.05 | 1950 | 907.0367 | 3489946 | 16 |
| 12 | 0.7 | 5 | 0.05 | 1925 | 332.1399 | 1294334 | 6 |

Based on this analysis, the parameters values were set to: $\alpha$=0.5, multiplying factor =5, and 1% for elite sampling size. Figure 4 shows the improvement in the objective function value from generation to generation of the best sample and of the average elite sample, when using the selected parameter values, as they both converge to the optimal solution. In Figure 5, we illustrate how the CE algorithm progresses in finding the optimal start time for a specific PM activity. It shows how initially, at generation #1, there are positive probabilities to consider the PM start time for time 0, 1, 2, …,7, but as generations progress, from the 7$^{th}$ generation onward, the PM start time is considered only for time 2 or 4 and eventually converges to start at time 2.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| best | 1950 | 1925 | 1925 | 1950 | 1925 | 1950 | 1950 | 1950 | 1950 | 1950 |
| average | 1905 | 1903 | 1905 | 1923 | 1918 | 1930 | 1930 | 1938 | 1943 | 1945 |

Figure 4: Objective function values by best and average elite samples.

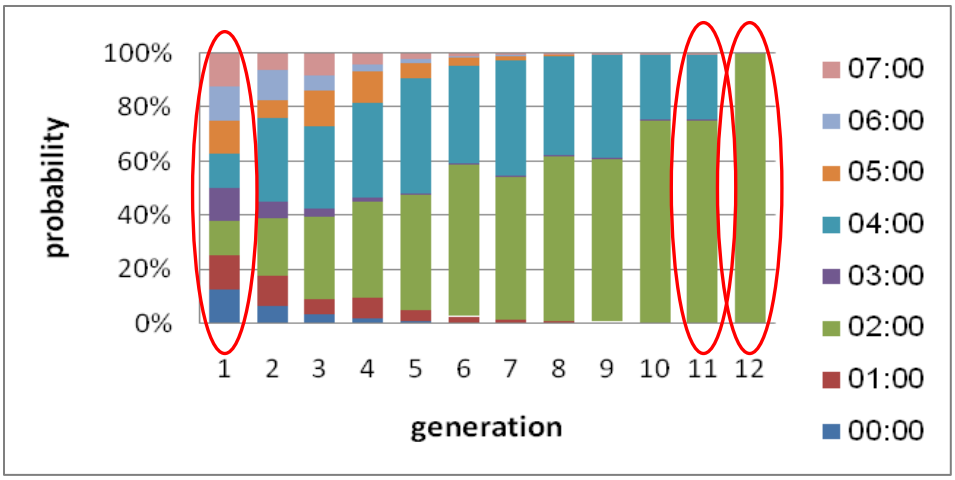



Figure 5: PM probabilities vector convergence over generations.

### 4.3 Numerical Results

A design of experiments with small, medium, and large scale problems (differing in the number of binary variables) has been conducted and the results are provided in detail in Table 4 and summarized in Table 5. Ten test cases were generated for each type. The number of generations varied between 5 to 17 in the small and medium types, and up to more than 60 in some of the large scale test cases.
For the large scale problems, the number of PM's was extended up to 10 and the planning horizon was extended to a full week (168 hours).

As can be observed, the MILP (optimal) solution provides solutions in a reasonable amount of time (i.e. up to a few minutes) for both small and medium scale problems. However, for large scale problems, which contain more than 200 binary variables, MILP becomes limited and the runtime to solution increases significantly to a few hours (and in two of the test cases, the optimal solution was not found even after 48 hours.)

The CE solution approach require longer time for the small and medium scale problems because of its initialization parameter values that take longer to converge – but it proves itself on the large scale problems where a solution is attained much faster; as reflected by the runtime ratio, which represents the ratio of the average runtimes of the MILP relative to the CE. Additionally, in all of the test problems that the MILP

failed to reach a solution even within 48 hours, the CE approach provided a solution within 4 hours on average. The quality of the CE solutions is satisfactory, with an average deviation of less than 1% from the MILP solutions. One interesting characteristic to note is that both approaches had high variance in their runtimes over the different problems. This is likely to occur due to the big effect of the nature of the QT restrictions on the amount of elite solutions. When the QT restrictions are tighter, the amount of elite solutions is reduced, which makes it harder to reach a solution.

Table 4: Details of the test cases for small, medium, large scale problems (CE vs. MILP).

| problem type | | method | # of gen's | number of lindo iterations | runtime (seconds) | runtime ratio | objective value | % deviation from optimal |
|---|---|---|---|---|---|---|---|---|
| *small* | *average* | MILP | | 265377.20 | 39.33 | 4.87 | 1524.60 | 0.00% |
| | | CE | 7.10 | 542079.10 | 126.76 | | 1524.60 | |
| | *stdev* | MILP | | 209401.48 | 30.74 | 2.93 | 463.44 | 0.00% |
| | | CE | 4.28 | 348565.37 | 80.46 | | 463.44 | |
| *medium* | *average* | MILP | | 548394.70 | 125.48 | 12.57 | 2035.00 | 0.49% |
| | | CE | 9.60 | 1748450.30 | 426.58 | | 2025.00 | |
| | *stdev* | MILP | | 596493.52 | 135.55 | 15.18 | 620.08 | 1.19% |
| | | CE | 2.84 | 846353.15 | 204.03 | | 617.28 | |
| *large* | *average* | MILP | | 27647974.80 | 14695.98 | 0.61 | 4847.60 | 0.93% |
| | | CE | 23.43 | 22134858.86 | 9534.86 | | 4755.43 | |
| | *stdev* | MILP | | 14937089.81 | 7006.09 | 0.40 | 35.93 | 0.92% |
| | | CE | 18.15 | 16753330.80 | 7425.20 | | 97.97 | |

Table 5: Comparison between the two solution approaches (CE vs. MILP).

| Problem Scale | Binary Variables | Method | Avg. No. of generations | Runtime (seconds) | Runtime ratio | %deviation from optimal |
|---|---|---|---|---|---|---|
| **small** | 50 | MILP | | 39.33 | 4.87 | 0.00% |
| | | CE | 7.10 | 126.76 | | |
| **medium** | 100 | MILP | | 125.48 | 12.57 | 0.49% |
| | | CE | 9.60 | 426.58 | | |
| **large** | 200 | MILP | | 14187.15 | 0.61 | 0.93% |
| | | CE | 27.25 | 10963.15 | | |

## 5 CONCLUSIONS AND FURTHER WORK

In this paper, we have formulated the PM-QT problem, of scheduling PM activities on tools within QT restrictions, such that overall throughput is maximized and the QT restrictions are not violated. We have shown that this problem can be formulated as an MILP model without an apparent efficient solution algorithm and thus proposed a metaheuristic approach based on the cross-entropy (CE) concept for its efficient solution. Using small and medium scale problems, we have demonstrated how to calibrate the CE

initialization parameters and then engaged it in solving large real-life industrial instances of the problem. We have shown that its solutions are indeed efficient in runtime reductions with almost no compromise of solution quality (less than 1% difference between MILP and CE solutions for large scale problems.)

Further work is still needed to evaluate other potential heuristic optimization techniques such as simulated annealing and genetic algorithms. Additionally, we recommend learning more about the performance of the CE convergence relative to other potential stopping criteria. Another item that was not explicitly considered in this work is the impact of equipment unscheduled downtime (i.e. breakdown). It may be that adding this consideration to the model may alter the quality of solutions and trigger further innovative approaches for obtaining efficient solutions.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the contributions to this work by two senior students at Ben-Gurion University, Roi Bareli and Anat Harel as well as the support of Prof. Gad Rabinowitz of Ben-Gurion University and Avri Altman of Intel.

## REFERENCES

Alardhi, M., R. G. Hannam, A. W. Labib. 2007. "Preventive Maintenance Scheduling for Multi-cogeneration Plants with Production Constraints." *Journal of Quality in Maintenance Engineering* 13: 276-292.

Altman, A., and A. Kalir. 2009. "A Solution to the Problem of Scheduled Maintenance of a Time-limit Constrained Segment in Semiconductor Manufacturing." *Intel Technical Report*.

Burda, R. 2008. "Fab Scheduling and Dispatch Methods Used to Meet Process Time Window Requirements." *5th International Symposium of Semiconductor Manufacturing Initiatives (ISMI)*, Austin, TX, USA.

Crespo Marquez, A., J. N. D. Gupta, and J. P. Ignizio. 2006. "Improving Preventive Maintenance Scheduling in Semiconductor Fabrication Facilities." *Production Planning & Control* 17(7): 742-754.

Margolin, L. 2002. Cross-entropy Method for Combinatorial Optimization. *Master's thesis*, Technion Institute of Technology, Israel.

Moghaddam, K. S., and J. S. Usher. 2009. "Maintenance Scheduling of Multi-component Systems Using Multi-objective Simulated Annealing." *Proceedings of Industrial Engineering Research Conference*, Louisville, USA.

Moghaddam, K. S., and J. S. Usher. 2011. "Preventive Maintenance and Replacement Scheduling for Repairable and Maintainable Systems Using Dynamic Programming." *Computers & Industrial Engineering* 60(4): 654–665.

Mosely, A., T. Teyner, and R. Uzsoy. 2004. "Maintenance Scheduling and Staffing Policies in a Wafer Fabrication Facility." *IEEE Transactions on Semiconductor Manufacturing* 11: 316-323.

Raza, S. A., and A. M. Al-Turki. 2007. "A Comparative Study of Heuristic Algorithms to Solve Maintenance Scheduling Problem." *Journal of Quality in Maintenance Engineering* 13: 398-410.

Robinson, J. K. 1998. Capacity Planning in a Semiconductor Wafer Fabrication Facility with Time Constraints Between Process Steps. *PhD diss.,* U. Mass. Amherst, USA.

Rubinstein, R. Y., and D. P. Kroese. 2004. *The Cross-entropy Method: A Unified Approach to Monte-Carlo Simulation, Randomized Optimization and Machine Learning*, Springer Verlag.

Suresh, K., and N. Kumarappan. 2006. "Combined Genetic Algorithm and Simulated Annealing for Preventive Unit Maintenance Scheduling in Power System." *IEEE Power Engineering Society General Meeting*, 5 pp.

Uzsoy, R., C. Y. Lee, and L. A. Martin-Vega. 1992. "A Review of Production Planning and Scheduling Models in the Semiconductor Industry, Part I: System Characteristics, Performance Evaluation and Production Planning." *IIE Transactions* 24: 47-60.

Van Dijkhuizen, G., and A. V. Harten. 1998. "Two Stage Generalized Age Maintenance of a Queue-like Production System." *European Journal of Operational Research* 108: 363-378.

Van Sickle, D. L., and E. F. Hertzler. 2006. "300mm Time Constrained Queue Loop Management." *International Symposium of Semiconductor Manufacturing (ISSM)*, 57-60.

Yao, X., E. Fernandez-Gaucherand, M. C. Fu, and S. I. Marcus. 2004. "Optimal Preventive Maintenance Scheduling in Semiconductor Manufacturing." *IEEE Transactions on Semiconductor Manufacturing* 17: 345-356.

## AUTHOR BIOGRAPHIES

**ADAR A. KALIR** received the B.Sc. and M.Sc. degrees in industrial engineering and management from Tel-Aviv University, Israel, and the Ph.D. degree in industrial and systems engineering from Virginia Tech. He is a Senior Principal Engineer with the Fab/Sort Manufacturing Division, Intel Corporation. In his position, he is leading projects for manufacturing operations optimization and production cycle time improvements throughout Intel's factories worldwide. He is also an Adjunct Professor with Ben-Gurion University, Israel. He also serves as a Co-chair for the IEEE TC-SMA (Semiconductor Manufacturing Automation). His email address is adar.kalir@intel.com.

**ISRAEL TIRKEL** is a faculty member in the department of Industrial Engineering and Management, at Ben-Gurion University of the Negev, Israel. He has worked for Intel Corporation, in Israel and the USA, for twenty-three years in senior management positions of fab operations. He received his B.Sc. with distinction in 1983, M.Sc. with distinction in 2009, and Ph.D. in 2011 at Industrial Engineering and Management, from Ben-Gurion University of the Negev. His research interests are semiconductors manufacturing operations, with focus on yield optimization, cycle-time forecasting, equipment performance, and capacity planning. He is an Associate Editor of IEEE Transactions on Semiconductor Manufacturing. His email address is tirkel@bgu.ac.il.