Optimization-based Litho Machine Scheduling with Load Balancing and Reticle Expiration

Bing Yan, Hsin Yuan Chen, Peter B. Luh, Fellow, IEEE, Simon Wang, Joey Chang

Abstract — The increasing demand for on-time delivery is forcing semiconductor manufacturers to seek more efficient schedules of machines and other resources to meet targets while making effective use of excess capacities. This paper presents the modeling and solution methodology of lithography machine scheduling with load balancing and reticle expiration consideration for the bottleneck process of semiconductor fabrication. A mixed-integer optimization model is established with capacity constraints and processing requirements. The objective is to meet daily targets, to keep load balancing, to avoid reticles shortage, and to reduce the number of machine setups. The problem formulated above is solved by the branch-and-cut method. Although the formulation is linearized, the practical size problem is still difficult to solve because the convex hull is hard to obtain. To overcome efficiency difficulty, some constraints are modified. To further improve efficiency, a two-phase model is established. The higher phase is to reduce the problem range by relaxing some constraints, and the lower phase is for scheduling. Numerical testing shows that the method can generate high quality schedules within reasonable time.

I. INTRODUCTION

With fierce competition, the increasing demand for on-time delivery is forcing semiconductor manufacturers to seek more efficient schedules of machines and other resources to meet targets while making effective use of excess capacities. In a fab, about 35% ~ 45% of the lots’ waiting time and work-in-process (WIP) are observed in front of the bottleneck stage, e.g., litho stage [1]. Lithography is the process of transferring a pattern to the photoresist surface of a wafer by selective exposure to light through the reticle. After washing away unexposed regions, the pattern is left as a layer on the wafer after removing the remaining photoresist. The number of litho machine is limited and the lifetime of reticle is limited, so wafers will re-visit litho machine for multiple-layer processing. Machine load imbalance and reticle shortage will affect target achievement. For the litho machine scheduling problem under consideration, the key issue is how to assign litho machines and reticles to process layers.

Effective scheduling of litho machines with load balancing and reticle expiration considerations, however, is challenging. The litho process is so sensitive that some layers must be processed on the same machine. To avoid overload or starvation, stacking machines need to be assigned with load balancing based on the estimated WIP information. In addition, based on the property of reticle, it needs to be repaired outside the fab after processing certain number of lots. To prevent reticle shortage in fabrication, reticles that can process the same layer need to be scheduled to avoid expiring at the same time based on their remaining lifetime.

In this paper, a litho machine scheduling model is established in Section 3 based on previous work. For load balancing, the idea here is to minimize the load difference between every stacking machine and the average. Stochastic method is used to calculate the total load based on estimated WIP and historical cycle time data. In terms of reticle expiration, the novel idea is to keep the reticles expire gradually with same time interval. That is to keep the remaining lifetime difference between two reticles at the certain time interval. Resource capacity and processing requirements are basic constraints. Modeling of load balancing and reticle expiration are major components in the model. The objective is to meet daily targets, to keep load balancing, to avoid reticle shortage, and to reduce the number of machine setups.

The problem is solved by the branch-and-cut method, a method for combinatorial optimization of mixed-integer linear problems. To solve the problem in a computationally efficient manner, entire formulation should be linear. The method has been implemented by using IBM ILOG CPLEX, and two examples are presented to demonstrate efficiency and performance of the method in Section 4. Although the formulation is linear, the practical size problem is still difficult to solve because the convex hull of the problem is hard to obtain as explained in the small example. To overcome efficiency difficulty, some constraints are modified. The second example based on practical data is to demonstrate that our method can generate near-optimal schedules of practical sizes.

To further improve efficiency, a two-phase model is established in Section 5. The higher phase is to reduce the problem range by relaxing some constraints, and the lower phase is for scheduling. The convex hull of the first phase is analyzed in the small example. A practical size example is to compare the efficiency of one-phase and two-phase model. The numerical testing shows the method can generate high quality schedules within reasonable time.
II. LITERATURE REVIEW

Developing effective scheduling methods and approaches for semiconductor manufacturing system is very challenging because of its large-scale and complex re-entrant characteristics. In the past few decades, semiconductor manufacturing scheduling has attracted much attention. Some methods such as rules of earliest due date, shortest processing time and critical ratio are widely discussed. As the bottleneck process, approaches for optimizing lithography process are a matter of particular interest and recorded in some publications, including heuristic rules [2], mathematical programming techniques [3], and simulation techniques [4]. For load balancing, hierarchical approaches were developed based on the two boundary algorithm for WIP balancing [5, 6]. Scheduling methods were proposed for lot release, WIP balancing, and load balancing based on a modified two boundary method [1]. Balance rate was defined as the ratio of WIP difference between actual and target, and the machine with high rate was dispatched first. To tackle the load balancing issue in the semiconductor manufacturing system, a novel model, Resource Schedule and Execution Matrix, was presented to easily schedule the wafer lots by using a simple two-dimensional matrix representing activities of tasks in the factory [7]. The wafer lot with the biggest wait steps was assigned to the smallest load litho machine. For simplifying the simulation, each wafer lot was assumed to have the same process steps and quantity, and each layer stage was assumed to have the same process time. A new WIP balancing concept was presented, which directly considered load levels of bottleneck machines for higher throughput [8]. Also, a mixed-integer programming model is developed and solved by CPLEX, but the model can only decide the quantity of wafer lots processed on litho machine. By applying dynamic programming method to the machine constraint in the litho machine, a load balance allocation function was developed, focusing on allocating the first layer of each input lot to maintain the load balance [9]. A load balancing approach performed on the basis of a detailed simulation model was discussed [10]. The wafer lots assignment of the litho machines was decided at the time when the wafer lots were leased to the manufacturing system to improve the load balancing problem in the photolithography area. Another simulation-based approach focusing on dispatching policies was explored to primarily investigate the objective cycle time [11]. An approach combining simulation and artificial intelligence was presented, and a Neural Network approach using simulations was proposed to model the photolithography scheduling problem [12]. The primary goal was the minimization of WIP, setup time and throughput time. Reticle expiration issue was not discussed much in the literature of semiconductor manufacturing scheduling.

III. PROBLEM FORMULATION

The formulation is built on previous work about litho machine scheduling with multiple reticles and setups [13]. It has the following new features: (a) machine preventive maintenance (PM) requirements; (b) layer load requirements; (c) reticle expiration considerations to avoid that reticles expire at the same time; and (d) load balancing to avoid overload or starvation of stacking machines. The objective is to meet daily targets, to keep load balancing, to avoid reticle shortage, and to reduce the number of machine setups. In the formulation, denote the discrete time slot index $k$ ($1 \leq k \leq K$), the machine index $m$ ($1 \leq m \leq M$), reticle index $r$ ($1 \leq r \leq R$), product index $p$ ($1 \leq p \leq P$), and layer index $l$ ($1 \leq l \leq L$). Based on the characteristics of reticles, let $R_g$ denote the set of reticles that can process layer $l$ of product $p$.

A. Modeling of resource capacity constraints

The key decision variables are $\{\delta_m(k)\}$, binary, where $\delta_m(k) = 1$ indicates that machine $m$ is combined with reticle $r$ to process layers at time slot $k$; and $\delta_m(k) = 0$, otherwise.

1) Machine Capacity Constraints

Each machine can only be combined with one reticle to process layers at any time slot, i.e.,

$$\sum \delta_m(k) \leq 1, \forall k, \forall m.$$  

2) Reticle Capacity Constraints

Each reticle can only be combined with one machine to process layers at any time slot, i.e.,

$$\sum m \delta_m(k) \leq 1, \forall k, \forall r.$$  

3) Machine-Reticle Matching Requirements

Machine $m$ cannot use the reticles involved in set $R_m$, i.e.,

$$\delta_m(k) = 0, \forall k, \forall m, and r \in R_m.$$  

4) Machine PM requirements

Litho machine needs to do preventive maintenance to prevent fault and guarantee service every certain time duration. For the scheduling problem under consideration, begin and complete time of PM on particular machine is known before scheduling, therefore machine PM scheduling is not involved. During PM, machine is not available, e.g.,

$$\delta_m(k) = 0, k \in [b_m^T, e_m^T], m \in M^{PM} and \forall r.$$  

In the above, $M^{PM}$ denotes the set of machines need to do PM within the scheduling horizon, and $b_m^{PM}, e_m^{PM}$ denote begin and complete time of PM duration on machine $m$.

B. Processing requirements

1) Processing time requirements

Let $T_{mr}$ denote the time required to process the unfinished lot left over from previous scheduling horizon on machine $m$ and reticle $r$, and $T_{mr}'$ the time required to process the unfinished lot left over for next horizon. The former ones are known, and the later ones are integer decision variables. Each processing period must be assigned required amount of raw processing time $T_{mr}$ on machine $m$ with reticle $r$, i.e.,

$$\sum \delta_m(k) - T_{mr} + T_{mr}' = N_{mr} \times T_{mr}, \forall r, \forall m;$$  

$$0 \leq N_{mr} \leq N_m^{UB}, N_{mr} = (K - T_{mr}')/T_{mr} + 1, \forall r, \forall m.$$  

In the above, the number of lots processed on machine $m$ and reticle $r$ within scheduling horizon is denoted by $N_{mr}$, integer decision variable, and $N_m^{UB}$ denote its upper bound.

Since the last lot might be unfinished, the decision variables $T_{mr}'$ must smaller than processing time, i.e.,
0 \leq T_{mr}^l < T_{mr}^l - 1, \forall r, \forall m, \forall l, (6)

2) Layer load requirements

When the capacity exceeds the target much within scheduling horizon, the model tends to allocate machines and reticles to process the most priority layer with the extra capacity. However, this may lead to imbalance among the layers of the same product. To avoid this, the layer load requirements set an upper bound for each layer, and it’s certain times of the target. The upper bound $T_{pl}^{UB}$ for layer $l$ of product $p$ will be the bigger one between certain times of target $T_{pl}$ and the smallest integer that bigger than target, i.e.,

$$T_{pl}^{UB} = \max\{1.2 \times (T_{pl}^{UB} \div 25), \lceil \frac{T_{pl}^{UB}}{25} \rceil \}, \forall p, \forall l, (7)$$

In the above, $\lceil T_{pl} \rceil$ denotes ceiling function and $\max$ denotes max function. Although both of them are nonlinear, they will not increase complexity because no decision variables are involved. The unit for target and upper bound is wafer and lot respectively, and one lot contains 25 wafers. The layer load of every layer cannot exceed its upper bound, e.g.,

$$N_{pl} \leq T_{pl}^{UB}, \forall p, \forall l. (8)$$

In the above, $N_{pl}$ denotes the number of lots (might not be integer) for layer $l$ of product $p$ within scheduling horizon, and it can be obtained from $N_{mr}$ and $T_{mr}$ as follows,

$$N_{pl} = \sum_{m \in \mathcal{M}_p} \sum_{r \in \mathcal{R}} (N_{mr} + T_{mr}^l \div T_{mr}^l), \forall p, \forall l. (9)$$

C. Modeling of machine setup

The set of binary decision variables $y_{mr}(k)$ is used to obtain the machine setup status. The relationships between $y_{mr}(k)$ and main decision variables $\delta_{mr}(k)$ can be shown as follows,

$$y_{mr}(k + 1) \geq \delta_{mr}(k) \div (k + 1) - \delta_{mr}(k), \forall m, \forall r, k \in \{1, K - 1\};$$

$$y_{mr}(k + 1) \geq \delta_{mr}(k) \div (k + 1), \forall m, \forall r, k \in \{1, K - 1\}. (10)$$

D. Modeling of load balancing

1) Load balancing

The litho process is so sensitive that some layers must be processed on the same machine. For example, layers A and B of product D must be processed on the same machine, and the allocation of layer A will fix the process of B. To avoid overload or starvation, the load among stacking machines should be kept balanced. The idea here is to minimize the load difference between stacking machines and the average. Denote the total load on machine $m$ and average load from layer $A$ to $B$ by $TL_{m}^{A:B}$ and $AL^{A:B}$, and the load difference $LD_{m}^{A:B}$ on stacking machine $m$ can be decired as follows,

$$LD_{m}^{A:B} = TL_{m}^{A:B} - AL^{A:B}. (11)$$

2) Total load of the stacking machine

For the scheduling problem under consideration, the estimated number of WIP lots that will flow to the stacking machine is known in advance based on the current process conditions. In addition, the process statuses, e.g., which layer needs to be processed next and how long time since the lot left previous layer stage, are also known. To overcome the uncertainty difficulty, our novel idea is to use stochastic method to obtain total load. Based on historical data, the cycle time between two successive layer stages and the probability can be obtained. Let $WIP_{mr}^{A:B}$ denote the number of WIP which was processed layer $A$ $d$ days ago and needs to be processed layer $B$ on machine $m$. Let $P(C_{T_{mr}^{A:B}})$ denote the probability that the cycle time is $d$ days from layer A to B. The total load to be processed layer B in the 7th day on stacking machine $m$ can be calculated as follows,

$$TL_{mr}^{A:B} = \sum_{j} WIP_{mr}^{A:B} \times P(C_{T_{mr}^{A:B}}) + N_{mr}^l \times P(C_{T_{mr}^{A:B}}). (12)$$

In the above, $N_{mr}^l$ denotes the number of lots that is scheduled to process layer A on machine $m$ within the scheduling horizon. It might be load in the future, therefore it is also in the load calculation. The total load formulation for multiple layers in one stacking group is similar.

E. Modeling of reticle expiration considerations

1) Reticle remaining lifetime

Based on the property of reticle, it needs to be repaired outside the fab after processing certain number of lots, and it usually takes several days. To avoid reticle shortage in fabrication, reticles that can process the same layer are not expected to expire at the same time. In ideal case, their expiration dates should be equally spaced, which means they should expire gradually with same time interval. Here reticle remaining lifetime is used to consider expiration, which represents how many lots one reticle can process before repairing. Let $R_{r}^0$ and $R_{r}$ denote the remaining lifetime of reticle $r$ before and after scheduling respectively. The relationship between them can be easily obtained as follows,

$$R_{r} = R_{r}^0 - \sum_{m \in \mathcal{M}_r} N_{mr}, \forall r. (13)$$

2) The remaining lifetime difference between two reticles

To keep the expiration dates of reticles equally spaced, our novel idea is to keep the remaining lifetime difference between two reticles the expected gap. However, not all remaining lifetime differences of every two reticles are desired to compare with the gap. Only the difference of the two reticles, which can process the same layer and whose remaining lifetime is most close to each other, is reasonable and useful. To overcome this difficulty and find the right comparison pairs, a sequence will be established for every group of reticles that can process the same layer based on their remaining lifetime. Each reticle will be given a ranking number, the smaller the longer remaining lifetime.

F. Objective function

The objective function has four terms, to meet daily targets, to keep load balancing among stacking machines, to avoid reticles shortage, and to reduce the number of machine setups.

For meeting targets, let $W_{pl}^R$ and $W_{pl}^P$ denote the reward (over target) weight and penalty (under target) weight for layer $l$ of product $p$. The second term is to keep load balancing among stacking machines, and let $W_{m}^L$ denotes the weight for load balancing, $SG$ stacking groups, and $SM_{p}$ the stacking machines within stacking group $g$. For reticle remaining lifetime gap part, let $W_{R}^{RR}$ and $W_{R}^{RP}$ denote the reward (over gap $G_{p}$) and penalty (under gap) weight. The last term is to reduce the number of setups with weight $W_{S}$. In sum, the objective function can be described as follows,

$$\sum_{p} \sum_{l} (W_{pl}^R \times \min(N_{pl} \times 25 - T_{pl}^l, 0)) - W_{pl}^P \times \max(N_{pl} \times 25 - T_{pl}^l, 0)$$

$$+ W_{L}^L \times \sum_{g \in SG} \sum_{m \in SM_{p}} |LD_{m}| + \sum_{p} \sum_{l} W_{R}^{RR} \times \max(\sum_{p} N_{pl}^l - G_{p}, 0) + \sum_{p} \sum_{l} W_{R}^{RP} \times \min(\sum_{p} N_{pl}^l - G_{p}, 0)$$

$$+ W_{S}^S \times \sum_{g \in SG} \sum_{m \in SM_{p}} |LD_{m}| + \sum_{p} \sum_{l} W_{R}^{RR} \times \max(\sum_{p} N_{pl}^l - G_{p}, 0) + \sum_{p} \sum_{l} W_{R}^{RP} \times \min(\sum_{p} N_{pl}^l - G_{p}, 0). (14)$$
\[ (-W^{\min} \times \min(R_r - R_p - G_{pi}, 0) - W^{\max} \times \max(R_r - R_p - G_{pi}, 0)) \\
+ W^{\min} \times \sum_{m} \sum_{r} y_{mr}(k) \times 2. \]

(14)

IV. SOLUTION METHODOLOGY AND NUMERICAL RESULTS I

The problem is solved by the branch-and-cut method, a method for combinatorial optimization of mixed-integer linear problems. To solve the problem in a computationally efficient manner, entire formulation should be linear. The nonlinear terms in objective function can be linearized by special ordered set techniques \([14]\) and replacing absolute value by new decision variables, similar to the modeling of machine setup. The method presented above has been implemented by using the optimization package IBM ILOG CPLEX Optimization Studio Version 12.2. Testing has been performed on a PC with 1.60GHz Intel (R) i7 CPU and 4G RAM. Two examples are presented to demonstrate efficiency and performance of the method developed in Section 4. Although the formulation is linear, the practical size problem and performance of the method based on the small example need to be evaluated. The convex hull analysis with a simple example

A. The convex hull analysis with a simple example

The following simple two-machine and two-reticle litho scheduling problem is used to analyze the convex hull. The number of time slots is also two, and the raw processing time is one time slot for simplicity. The problem is solved with resource capacity constraints and layer load requirements, the objective is to meet targets and reduce the number of setups.

Without layer load requirements, the optimal integer solution is \(\delta_{mr}(k) = (0, 0, 1, 1, 0, 0)\), \(y_{mr} = (0, 0, 0, 0, 0)\). To the purpose of visualization, \(\delta_{m2r1}(k)\) and \(y_{mr}\) are selected to plot a 3-D figure shown in Fig. 1. The other decision variables are fixed as optimal integer solution. It is obvious the convex hull of all feasible solutions to the original problem is ACBD and the optimal solution is D (1, 1, 0).

Then with layer load requirements, the optimal integer solution is \(\delta_{mr}(k) = (0, 0, 1, 1, 0, 0)\), \(y_{mr} = (0, 0, 0, 0, 0)\). \(\delta_{m2r1}(k)\) and \(y_{mr}\) are still selected to plot a 3-D figure as in Fig. 2. It is obvious the convex hull of all feasible solutions to the original problem is ABC. After relaxing the integrality constraints, all decision variables can take any value within \([0, 1]\), and the optimal solution to the relaxed problem is D (0.6, 0.6, 0). A cover cut \(\delta_{m2r1}(1) + \delta_{m2r1}(2) \leq 1\) can be generated from the layer load constraints \(\delta_{m2r1}(1) + \delta_{m2r1}(2) \leq 1.2\), and the optimal solution to the relaxed problem is (0.5, 0.5, 0).

In general, to avoid non-integer coefficients, the upper bound of layer load can be revised as follows,

\[ T_{\text{rel}} = \left[ \max \left( 1.2 \times (T_{\text{pi}} + 25), T_{\text{pi}} + 25 \right) \right] \forall p, \forall l. \]

However, the convex hull ABC still cannot be obtained by adding more cuts on the feasible region. This difficulty is caused by the interactions between decision variables \(\delta_{mr}(k)\) and \(y_{mr}\). Branching will be performed for every dimension based on the continuous optimal solution, and the total number of branching operations grows exponentially when the problem size increases, which leads to the low efficiency.

Similarly, the processing time requirements (5) with multiple decisions \((\delta_{mr}(k), N_{mr}, T_{mr})\) also increase the difficulty to obtain the convex hull. To overcome this difficulty, processing time requirements can be relaxed by ignoring the unfinished lot at the end of the scheduling horizon denoted by \(T_{\text{mr}}\) as follows,

\[ N_{mr} = \sum_{r} (\delta_{mr}(k) - T_{mr}) = T_{mr}, \forall r, \forall m. \]

In the above, \(N_{mr}\) still means the number of lots processed on machine \(m\) and reticle \(r\) within the scheduling horizon, but is not a decision variable and not integer.

B. Practical size example

This example is to demonstrate our method can generate near-optimal solutions for practical size litho machine scheduling problems, and show the insights obtained. In this problem, seven layers of one product are to be scheduled on eleven machines and seventy-one reticles within 411 time slots. The problem is first solved with linearized objective function (14) and constraints (1-13). Then it is solved with different modification of constraints involving layer load and processing time requirements. The following results are to compare the efficiency and performance of different formulations.

Fig. 2. The feasible region and convex hull with layer load
TABLE I  
TESTING RESULTS FOR DIFFERENT FORMULATIONS

<table>
<thead>
<tr>
<th>Formulation</th>
<th>Stop time</th>
<th>Stop Gap</th>
<th>CPU time</th>
<th>Gap</th>
<th>Number of nodes processed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without layer load (1-6, 9-14)</td>
<td>5m</td>
<td>5%</td>
<td>302s</td>
<td>5.3%</td>
<td>62</td>
</tr>
<tr>
<td>All constraints (1-14)</td>
<td>5m</td>
<td>5%</td>
<td>303s</td>
<td>104%</td>
<td>228</td>
</tr>
<tr>
<td>New layer load (1-6, 8-15)</td>
<td>5m</td>
<td>5%</td>
<td>301s</td>
<td>179%</td>
<td>494</td>
</tr>
<tr>
<td>New layer load and no processing time requirements (1-4, 8-16)</td>
<td>5m</td>
<td>5%</td>
<td>279s</td>
<td>3.4%</td>
<td>1206</td>
</tr>
</tbody>
</table>

In the above, stop time and stop gap (relative difference between the objective of optimal relaxed solution and current integer solution) are stop criteria. The optimization will stop when the CPU time reaches upper bound or the gap falls below stop gap. CPU time and gap means the actual optimization running time and gap. From the results, without layer load requirements, the efficiency is good, and the gap can be reduced to 5% within five minutes. After adding these requirements in, it is obvious that the resulting gap is far from the previous one. With the modified layer load requirements, although relative gap is higher, the best integer solution is getting better. At last, when the layer load and processing time requirements are both modified, the results are very encouraging. In sum, it can be shown that linear formulation cannot guarantee high efficiency and good results. It is helpful to analyze and characterize the entire formulation, find the troubling constraints and modify correspondingly.

To show our method can generate near-optimal solutions for practical size problems, the results analysis of the problem solved with new layer load and no processing time requirements is also presented. The scheduling results meet all targets except for one layer, because there are no available reticles. For load balancing, standard deviation of stacking machine total load before scheduling is 15, and after is 11. The total load among stacking machines from layer A to C before and after scheduling is compared in Fig. 3. It can be seen that after the scheduling, load trends to be balancing. From the results of reticle remaining lifetime before and after scheduling, the gap between two near reticles becomes bigger after the scheduling, which will avoid reticles that can process same layer to expire at the same time.

V. SOLUTION METHODOLOGY AND NUMERICAL RESULTS II

To further improve the efficiency, a two-phase model is established. The higher phase is to reduce the problem range by relaxing some constraints, and the lower phase is for scheduling. Two examples are presented to demonstrate the efficiency and performance of this two-phase model. The convex hull of the first phase is analyzed in the small example. A practical size example is to compare the efficiency of one-phase model and two-phase model. The numerical testing shows the method can generate high quality schedules within reasonable time.

A. Two-phase model

In the first phase, the constraints with the machine setup modeling are removed, and the number of running reticles is used to control machine setup. So the objective function is to meet daily targets, to keep load balancing, to avoid the reticle shortage, and to reduce the number of running reticles. A new set of binary variables \( d_{mr} \) is used to detect whether reticle \( r \) is assigned to machine \( m \) within the scheduling horizon.

\[
d_{mr} \leq N_{mr} \leq N_{mr}^{UB} \times d_{mr}, \quad \forall m, \forall r. \quad (17)
\]

In the above, when \( N_{mr} \) is 0, the value of \( d_{mr} \) is 0; when \( N_{mr} \) is nonzero, the value of \( d_{mr} \) is 1. Therefore the objective function can be modified as follows,

\[
\sum \sum \left( -W_{pl} \times \min(N_{pr} \times 25 - T_{pr}, 0) \right) - \left( -W_{pl} \times \max(N_{pr} \times 25 - T_{pr}, 0) \right)
\]

\[
+ W \times \sum \sum |LD_{mr}| \left( -W_{mr} \times \min(R_{mr} - R - G_{pl}, 0) \right)
\]

\[
+ W_{mr} \times \sum \sum d_{mr}. \quad (18)
\]

For the first phase, the decisions involve the number of lots assigned to the machine and reticle, and it will be solved with objective function (18) and constraints (1-4, 10 and 16). The results \( N_{mr} \) from the first phase will be the input data as \( N_{mr}^{UB} \) for the next phase. For the second phase, the decision variables involve the processing number and time of lots scheduled to the machine and reticle that are assigned in the previous phase. The objective is to increase the productivity, to finish the target as soon as possible, and to reduce the number of running reticles and machine setups, i.e.,

\[
W_{mr} \times \sum \sum \delta_{mr}(k) + W \times \sum \sum \delta_{mr}(k) \times k
\]

\[
W_{mr} \times \sum \sum DR_{mr} + W_{mr} \times \sum \sum y_{mr}(k) + 2. \quad (19)
\]

The second phase model will be solved with objective function (19), constraints (1-4, 10 and 16), and one more constraint to connect with the first phase results as follows,

\[
N_{mr} \leq N_{mr}^{UB} \text{ for } m \text{ and } r \text{ assigned in the first level.} \quad (20)
\]

B. The convex hull analysis with a simple example

This example is same as the simple example presented in previous section, but the objective is to meet targets and reduce the number of running reticles. The problem is solved with resource capacity constraints, layer load requirements,
reticle running detection constraints (17), and a cover cut \( \delta_m(2) \leq 1 \).

The optimal integer solution is \( \delta_w(\ell) = (0, 0, 1, 1, 0, 0, 0), \)
\( d_w = (0, 1, 1, 0, 0), \) and \( \delta_m(2) \) are selected to plot a
3-D figure shown in Fig. 4. The other decision variables are
fixed as optimal integer solution. It is obvious the convex
hull of all feasible solutions to the original problem is ACE
and the optimal solution is A (1, 0, 1) (or E). After relaxing
the integrality constraints, all decision variables can take any
value within [0, 1]. All feasible solutions to the relaxed
problem are in polyhedron ABCDE, and the optimal solution
is B (1, 0, 0.5) (or D). In objective function, meeting target is
prior to reducing the number of running reticles, therefore
d\( m(2) \) will take the value of 1 due to integrality requirements.
The optimal solution A (1, 0, 1) (or E) is obtained.

\[ \text{Fig. 4. The feasible region and convex hull with cover cut} \]

C. Practical size example

This example is to compare the efficiency and performance
of one and two phase model in CPLEX. In this problem,
seven layers of one product are to be scheduled on eleven
machines and seventy-one reticles within 411 time slots.

**TABLE II**

<table>
<thead>
<tr>
<th>Model</th>
<th>Phase</th>
<th>Stop time</th>
<th>Stop Gap</th>
<th>CPU time</th>
<th>Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>One phase (1-4, 8-16)</td>
<td>First</td>
<td>300s</td>
<td>5%</td>
<td>279s</td>
<td>3.4%</td>
</tr>
<tr>
<td></td>
<td>Second</td>
<td>120s</td>
<td>0.1%</td>
<td>25s</td>
<td>0.1%</td>
</tr>
<tr>
<td>Two-phase</td>
<td></td>
<td>240s</td>
<td>0.5%</td>
<td>2s</td>
<td>0.1%</td>
</tr>
</tbody>
</table>

From the results, it can be seen that the efficiency of
two-phase model is better than one phase model. From the
detailed results, the performance of two-phase model is also
better, for example, the number of machine setups is smaller
than one phase model results.

VI. CONCLUSIONS

This paper investigates the litho machine scheduling in
semiconductor manufacturing, and two aspects of efforts are
involved: modeling and optimization.

In the modeling aspect, a novel mathematical model is
developed to solve litho machine scheduling with load
balancing and reticle expiration considerations based on
previous work. The formulation involves machine PM and
layer load requirements, modeling of load balancing and
reticle expiration, and objective function besides previous
resource constraints, processing time requirements and
modeling of machine setup.

Then litho machine scheduling problem is optimized by
using IBM ILOG CPLEX Optimization Studio. Based on
convex hull analysis, some constraints are modified, and the
efficiency and performance comparison among different
formulations are shown in testing results. In additional, the
results analysis about load balancing is also presented.

To further improve the efficiency, a two-phase model is
presented. After convex hull analysis, the efficiency and
performance comparison of one-phase model and two-phase
model is shown in the testing results.

REFERENCES

dbottleneck scheduling for a semiconductor fabrication line,” IIE Trans.,

side constraints,” European Journal of Operational Research, vol. 109,

hierarchical production scheduling environment for semiconductor wafer
fabrication,” IEEE Trans. on Robotics and Automation, Vol. 19, Issue 4,

Approach to Solve the Capacity Allocation Problem for Photolithography Area,” Lecture Notes in Computer Science, Vol.
3982, 2006.

2004.

2003.

photolithography machines in the semiconductor manufacturing
system,” Journal of Information Science and Engineering, vol. 24, no. 2,

maximization in semiconductor fabrication,” IEEE Trans. on Semicond.

allocation system,” IEEE Trans. on Semicond. Manuf., vol. 18, Issue 4,

of load-balancing problems in the photolithography area of a
semiconductor wafer fabrication facility,” in Proceedings of the 33rd

photolithography toolsets in a wafer fabrication facility,” in Proceedings

Improvements for Photolithography Process in Semiconductor

“Optimization-based Litho Machine Scheduling with Multiple Reticles
and Setups,” in IEEE 7th Conference on Automation Science and
Engineering, Trieste, Italy, Aug., 2011.

Ordered Sets,” Mathematical Programming, Vol. 10, No. 1, pp. 52-69,
1976.