Production Planning & Control,
Vol. 18, No. 3, April 2007, 217–225

Parallel machine scheduling subject to auxiliary resource constraints

E. CAKICI and S. J. MASON*

Department of Industrial Engineering, University of Arkansas, Arkansas

This paper is motivated by scheduling photolithography machines in semiconductor manufacturing wherein reticle requirements are the auxiliary resource constraints. As the problem is NP hard, two different heuristic solution approaches are developed. The performance of our network-based mathematical model and heuristics are evaluated through an extensive set of problem instances. The best performing heuristic method typically produces solutions that are 1.72% above optimal. If this method is used as the seed solution for a Tabu search-based post processing algorithm, schedules that are 0.78% above the optimal solution, on average, are possible.

Keywords: Machine scheduling; Optimization; Heuristics; Semiconductor manufacturing

1. Introduction

1.1 *Semiconductor manufacturing*

The manufacturing of integrated circuits (IC) on silicon wafers is a complex production process. Between 250–500 process steps are performed on 50–120 different types of equipment to produce a typical IC ('wafer fab' in figure 1). Complex process steps, batch processes, re-entrant flows and expensive equipment are typical features of the semiconductor manufacturing process. In designing semiconductor wafer fabs, companies strive to produce a large number of chips in the least amount of time possible with minimum cost. Production scheduling is one useful way in which companies work towards this goal. However, the large number of processing steps, product types, specific processing equipment capabilities, product priorities, and other factors cause wafer fab scheduling to be an extremely difficult task. Effective scheduling can produce noticeable results, as an improvement in the semiconductor manufacturing process can provide huge financial gains (e.g. a single silicon wafer of ICs can be worth over $100 K).

Products travel through semiconductor wafer fabs in lots. Each lot typically consists of 25 wafers. Photolithography is one of the main process steps in the wafer fabrication stage of semiconductor manufacturing. The objective of the photolithography process is the accurate and precise definition of a three-dimensional pattern on the wafer. During the photolithography process, the IC pattern is transferred from a photo mask ('reticle') onto photosensitive polymer, which replicates the pattern in the underlying layer (figure 2). Exposure tools ('steppers') transfer the pattern onto the wafer by projecting light through the reticle to expose the wafer. The exposed wafer is then developed by removing polymerised sections of photo resist from the wafer.

Exposure tool processing depends on both the lot and the correct reticle being available simultaneously. Integrated circuits are built by repeatedly constructing 'layers' with desired properties on the silicon wafer's surface. Every layer of each product can require its own unique reticle. Therefore, reticles can be thought of as 'auxiliary resource' constraints in the photolithography process. Reticle requirements are typically both product- and layer-dependent, and a reticle must be on the stepper for the duration of a lot's processing. As a

*Corresponding author. Email: mason@uark.edu

Production Planning and Control
ISSN 0953–7287 print/ISSN 1366–5871 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/09537280601035836

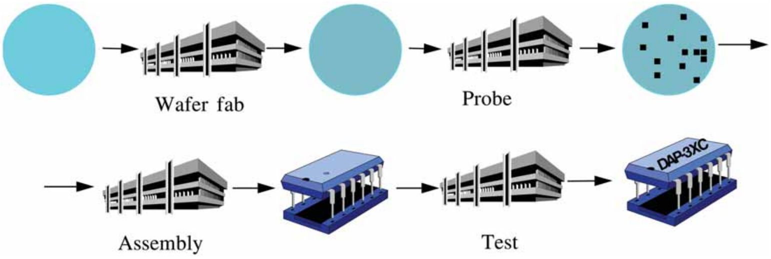

Figure 1. The semiconductor manufacturing process.

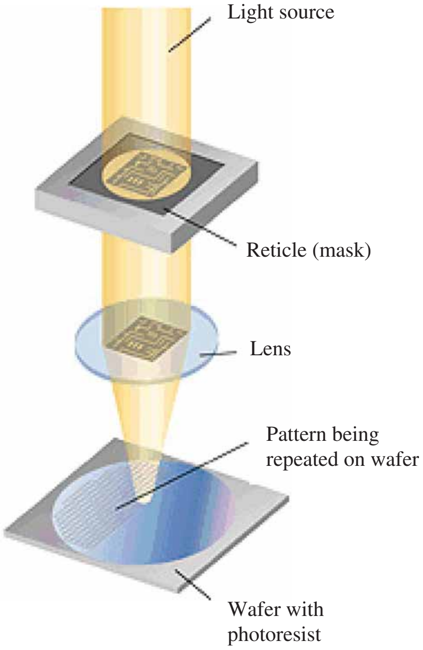

Figure 2. The photolithography process (adapted from http://www.just2good.co.uk/).

set of reticles for a single product can cost well over \$150 K, the relatively small number of reticles present in a wafer fab constrains the number of steppers that can process a given product layer simultaneously (Akcali and Uzsoy 2000).

Generally, more advanced products require 20 to 30 layers of processing. Therefore, 20 to 30 reticle changes must be performed to process a specific lot. Setup issues arising from reticle requirements can affect the productivity of photolithography tools and the entire wafer fab (as \$10 million plus steppers are typically a fab's bottleneck tool group (Park *et al.* 1999)). As 35 to 45% of a wafer fab's work-in-process (WIP) typically resides in the photolithography area (Lee *et al.* 2002) effective scheduling of the photolithography process can lead to substantial improvements in overall wafer fab performance.

1.2 *Literature review*

1.2.1 Photolithography scheduling. Akcali and Uzsoy (2000) investigated the photolithography work centre scheduling problem by investigating the setting of lot start and finish times over the duration of a production shift. A sequential procedure that divides the problem into capacity allocation and lot sequencing sub-problems is presented. In the capacity allocation phase, a greedy heuristic is developed which gives priority to the lot that has the largest number of the 'same type' lots waiting. These lots are allocated to the machine with the highest capacity in terms of unallocated capacity available for processing. However, it is assumed that an operation which requires a specific reticle type can be assigned to at most Λ machines, where Λ is the existing number of reticles of the corresponding type that are available. Simulation experiments show that their sequential procedure yields satisfactory results. It is also shown that stepper capability, which is determined by the operations that can be processed by the stepper, and the time horizon over which the problem is solved, affect cycle time, flow time, and the number of setups performed per day.

Akcali *et al.* (2001) examined the effects of various process control mechanisms on the photolithography process via another simulation study. Three test-run policies and two dedication policies are examined. A test-run is usually required after some specified number of lots have been processed on a stepper, thereby causing a reticle change to be required. In the dedicated assignment case, a lot is processed by the same stepper at every layer. Alternatively, a lot can be processed by any stepper at any layer in the flexible dedication case.

The first-in, first-out dispatching rule is applied such that if there is an idle stepper with the required reticle, the current lot is assigned to that stepper. Otherwise, the lot is assigned to any idle stepper and a reticle change is scheduled. If there is no idle stepper, the current lot queues. When a stepper becomes idle, the stepper buffer is checked, with the first lot that requires the same reticle that is currently on the stepper becoming assigned to the stepper. If there is no such lot, the first lot in the buffer is assigned to the stepper and a reticle change occurs. Of the three test policies, 'Policy 3 (P3)—Test Wafer is reworked with its Lot' is found to be the best approach. In this policy, the first wafer that is processed by the stepper is inspected while the rest of the lot is processed. If the first wafer fails the test, the first wafer is reworked with the entire lot.

Park *et al.* (1999) studied reticle management issues. They investigated different storage and inspection policies and investigate the relationship between cycle time and product mix with a simulation model. Machine balance is achieved when a random machine selection rule is applied. It is shown that when product mix is diversified, cycle time decreases due to decreased reticle requirements. In the case of a homogeneous product mix, lots often need to wait due to reticle unavailability, in addition to machine availability.

Lee *et al.* (2002) investigated scheduling rules to assign WIP to steppers to maximise production volume for some desired cycle time level. After prioritising lots by an index computed from the dispatching rules, the highest-ranking lot is allocated to the most suitable stepper among the available steppers by considering setup times and available capacity. The information regarding which stepper can process which lot is stored in a stepper-process capability table. In the authors' heuristic approach, the number of reticles available is assumed to be infinite, which is very unrealistic. It is shown that 'pull' type scheduling rules such as the apparent tardiness cost (ATC) rule of Vepsalainen and Morton (1987) are more effective in bottleneck circumstances when compared to other approaches.

Given target WIP levels, Kim *et al.* (2002) determined how many lots of each device/product type should be processed over an eight-hour shift by allocating them to eligible steppers. The objectives investigated by Kim *et al.* (2002) are to (1) minimise the difference between target and actual WIP and (2) maximise stepper utilisation by minimising the total number of setups. A mixed-integer-programming (MIP) model and three heuristic approaches are developed. The unique feature of their study is the consideration of setup times, as well as an alternative objective function. Based on solution quality and computation times, the linear programming and adjustment (LPA) approach is the most promising method proposed by the authors. In the LPA approach, the relaxed MIP model is solved quickly and then adjusted to make it integer-feasible.

Arisha and Young (2004) integrated simulation and artificial intelligence for lot scheduling on photolithography toolsets. The scheduling problem here is to assign a specific, available tool to process the job. The selection of the best tool for job processing is based on scores calculated for each available tool by evaluation of some selection criteria, such as lot priority, previous layer, previous product, and buffer status. The authors' simulation model provides insights regarding system performance: the uniform lot distribution assumption is false due to high system variability.

1.2.2 Multiple-resource constrained scheduling. Another area similar to photolithography stepper scheduling with reticle availability considerations is multiple-resource constrained scheduling (MRCS). MRCS is applied to problems in which two or more resources constrain the output of a system, as opposed to machine-limited systems in which machines are the only constraining resource. In MRCS systems, jobs can only be processed if both the machine and the associated secondary resources are available.

In dual-resource constrained systems (DRCS), two resources are considered to be constraining factors for job processing; this is similar to the stepper scheduling problem with auxiliary resource (i.e. reticle) constraints. In the literature, labour and machines typically represent the two constraining resources of concern. This type of DRCS is commonly termed a 'labour and machine-limited system.' Bokhorst *et al.* (2004) introduced the 'who-rule' for DRCS, focusing on the effect of the 'who-rule' on labour flexibility. The 'who-rule' is a labour allocation rule which selects one worker out of a group to be transferred to a work centre. 'Who rules' are applied in two cases if more than one skilled worker is available:

1. When a worker becomes available for transfer.
2. When a job arrives at an empty work centre.

Bokhorst *et al.* (2004) examined the following four 'who' rules in their experiments:

1. *Random* (*RND*) *rule*: Choose a worker randomly.
2. *Most efficient* (*MEF*): Assign worker who is most efficient at performing the task.
3. *Priority* (*PRIO*): Assign the least flexible worker (in terms of number of skills).
4. *Longest idle time* (*LIT*): Assign worker who has been waiting the longest.

'When' and 'where' rules are also quite common in the literature as 'labour allocation' rules. A 'when' rule determines when to transfer a worker from the work centre at which he/she currently works to another work centre. The 'where' rule determines to which work centre a worker needs to be transferred (Bokhorst *et al.* 2004).

Treleven (1989) presented a detailed review of DRCS research in which job shops are constrained by machine and labour availability. Dispatching rules and labour assignment rules examined in the DRCS literature are summarised. Treleven (1989) discussed labour flexibility and cross-training issues as well. Gargeya and Deane (1996) extended this study by reviewing research in MRCS and auxiliary resource-constrained environments. Auxiliary resources are 'resources required to make labour and machines productive' (Gargeya and Deane 1996). In contrast to homogeneous secondary resources in DRCS (i.e. resources that can process any job in queue), auxiliary resources are heterogeneous in that every job requires a unique resource.

Gargeya and Deane (1999) presented a simulation study for dynamic job shop scheduling in the presence of auxiliary resource constraints. They developed dispatching rules to prioritise jobs and auxiliary resource assignment rules for assigning resources to machines. Their contingency-based scheduling (CBS) approach gives scheduling priority to the most constrained (i.e. critical) resource at any given time. Priority is given to jobs that require the critical auxiliary resource as well. The CBS approach is compared with combinations of dispatching and tool assignment rules; such as first come first served (FCFS) and least number of jobs in queue at next machine (LNJQNM) rules. Simulation results suggest that the CBS approach performs better than its competitors in terms of the average time a job spends in the system.

A key characteristic of labour-constrained DRCS is that the number of workers is less than the number of machines, which implies that attention should be given to cross-training and labour allocation rules. By using labour allocation rules, one worker is selected to be transferred to the machine to which a job has been assigned. The common way to approach the dual and multiple resource constrained job shop problem is to use priority rules for assigning jobs to machines and assignment rules for assigning auxiliary resources to jobs at the machines (Nonas and Olsen, 2005). A number of different DRCS simulation studies have been reported in the literature, such as El-Maraghy *et al.* (2000) who apply a genetic algorithm to the DRCS problem to minimise a wide array of scheduling performance measures, such as mean flow time, maximum weighted tardiness and worker utilisation.

Kempf *et al.* (1998) studied single batch-processing machine scheduling with job families as motivated by burn-in operations in the final testing stage of the semiconductor manufacturing process. In their study, load board availability constraints are considered as secondary constraints. The authors presented four heuristic approaches in attempting to minimise makespan and to minimise the sum of completion times. The heuristics first try to find 'good' batches for each job family, and then a dispatching rule is applied to sequence the jobs on the machine. Because of computation time limitations, heuristics are evaluated in terms of lower bounds. Another algorithm which runs each of the four heuristics and selects the schedule with the best objective function value provides the most promising solutions in an acceptable amount of computation time.

1.3 *Problem statement*

A review of the available literature indicates that while a substantial amount of research has been done in machine-only and MRCS, there is a dearth of research that explicitly considers auxiliary resource constraints in scheduling. The research studies motivated by the photolithography process of semiconductor manufacturing are somewhat similar to resource allocation problems due to (1) researcher assumptions and (2) the solution methods employed. Moreover, the majority of previous research efforts use discrete event simulation to study problems to examine the effect of control policies on various performance criteria. This type of simulation model can be expensive and time-consuming to develop, debug, and run (El-Maraghy *et al.* 2000).

In previous research, the general approach for managing reticles is that reticles must be shared among machines and that it is not possible to use the same reticle in different machines in non-overlapping time periods. In this paper, photolithography scheduling motivates our investigation into parallel machine scheduling in the presence of auxiliary resource constraints to minimise total weighted completion time. This is important research, as steppers are intended to be the bottleneck resource in wafer fabrication facilities due to their high capital cost (> $10 million plus). In this paper, (1) reticles are allowed to be used in different machines in non-overlapping time periods and (2) jobs are investigated with non-zero ready times—both of these characteristics are inherent in semiconductor wafer fabs.

2. Mathematical programming formulation

First, the pertinent notation for mathematical formulation is introduced:

J The set of jobs or production lots.
L The set of processing layers.
δ_l The set of jobs which require layer $l \in L$ processing.

An actual wafer fab environment is emulated in which jobs arrive for processing to the stepper tool group at various points in time during a production shift.

Each arriving job requires processing at some specified process layer, and it is assumed that all jobs can be processed by any stepper, regardless of a job's associated layer. Job $j \in J$ has the following parameters associated with it:

p_j The processing time of job $j \in J$.
r_j The ready (release) time of job $j \in J$.
w_j The weight (priority) of job $j \in J$.

A total of m identical steppers operating in parallel are considered. Finally, the time at which job $j \in J$ finishes its required processing is denoted by C_j. Given this preliminary notation, a network-based optimisation model formulation is introduced.

2.1 *Network formulation*

First, a dummy job 0 is introduced whose processing time, ready time, and weight are each set equal to 0. In the network formulation of the parallel machine scheduling problem subject to auxiliary resource constraints, job 0 is required to be both the first and the last job processed on each machine. In this way, it indicates both the starting and finishing of job processing on each machine.

Binary decision variable x_{ij} is defined to assist with job sequencing such that $x_{ij} = 1$ if job $i \in J$ immediately precedes job $j \in J$ on the same machine; otherwise, $x_{ij} = 0$. Due to reticle availability concerns, it must be ensured that jobs requiring the same unique reticle do not have overlapping time periods in which they are being processed on different machines. Therefore, another binary decision variable e_{ij} is introduced, where $e_{ij} = 1$ if job $i \in J$ finishes its processing before job $j \in J$ starts its processing; otherwise, $e_{ij} = 0$ ($\forall i \in \delta_l$, $\forall j \in \delta_l$, $\forall l \in L : i \neq j$).

The objective function of interest in this paper is to minimise total weighted completion time ($TWCT$) of all jobs, where

$$TWCT = \sum_{j:j \in J} w_j C_j.$$

In terms of the network model, its constraints are presented in a manner typically associated with the travelling salesman problem. Jobs are assigned to each of the m available machines, subject to each machine starting and ending its schedule with job 0:

$$\sum_{j \in J: j \neq 0} x_{0j} \leq m \tag{1}$$

$$\sum_{j \in J: j \neq 0} x_{j0} \leq m \tag{2}$$

Scheduling constraints dictate machine assignment and job sequencing for jobs on the same machine:

$$\sum_{j \in J: j \neq i} x_{ji} = 1 \quad \forall i \in J : i \neq 0 \tag{3}$$

$$\sum_{j \in J: j \neq i} x_{ij} = 1 \quad \forall i \in J : i \neq 0 \tag{4}$$

In order to process job j immediately after job i on the same machine, job i's completion time C_i is determined by the following relationship:

$$C_i - C_j + \left(M - \min_{k \in J: k \neq 0} \{r_k + p_k\} + p_j \right) x_{ij} \leq M - \min_{k \in J: k \neq 0} \{r_k + p_k\} \quad \forall i \in J, \forall j \in J : j \neq 0, i \neq j \tag{5}$$

In constraint (5),

$$M \geq \max_{j \in J} \{r_j\} + \sum_{j \in J} p_j$$

(i.e. big M). In order to improve the tractability of the network formulation, constraints (6) and (7) are additional valid inequalities that are added to the model:

$$C_i \geq r_i + p_i + \sum_{j \in J: i \neq j} \left(\max(0, r_j + p_j - r_i) \right) x_{ji} \quad \forall i \in J : i \neq 0 \tag{6}$$

$$C_i \leq M - \sum_{j \in J: i \neq j} p_j x_{ij} \quad \forall i \in J : i \neq 0 \tag{7}$$

Finally, constraints (8) and (9) ensure that if two jobs require the same reticle type, one of the jobs should complete its processing before the other job's processing starts:

$$C_i - C_j + \left(M - \min_{k \in J: k \neq 0} \{r_k + p_k\} + p_j \right) e_{ij} \leq M - \min_{k \in J: k \neq 0} \{r_k + p_k\} \quad \forall i \in \delta_l, \forall j \in \delta_l, \forall l \in L : i \neq j \tag{8}$$

$$e_{ij} + e_{ji} \geq 1 \quad \forall i \in \delta_l, \forall j \in \delta_l, \forall l \in L : i \neq j \tag{9}$$

2.2 *Problem complexity*

In the scheduling notation scheme of Lawler *et al.* (1982), the research problem of interest in this paper is denoted as $Pm|r_j, aux|\sum w_j C_j$. This problem reduces to the single machine $1|r_j|\sum w_j C_j$ problem, which is NP hard even if $w_j = 1$ for all jobs $j \in J$ (Lenstra *et al.* 1977). In addition, reticle availability concerns further constrain the feasibility conditions of the scheduling problem of interest, thereby making the problem even more difficult. Therefore, a number of heuristics are developed in an attempt to produce near-optimal solutions in an acceptable amount of computation time.

3. Heuristics for $Pm|r_j, \mathrm{aux}|\Sigma w_j C_j$

This section describes the development of effective heuristics for generating high quality, feasible solutions for the scheduling problem of interest. The first two heuristics presented below are based on common approaches found in the scheduling literature. Next, two two-phase heuristic approaches to minimise total weighted completion time when auxiliary resources constraints exist in a parallel machine environment are presented. In the first 'construction' phase, it is assumed that jobs requiring the same reticle type are assigned to the same machine. Next, the 'improvement' phase two utilises adjacent pairwise interchanges to produce better solutions (i.e. lower *TWCT*) through workload balancing.

3.1 *Heuristic H1*

Let ψ denote the machine that becomes idle at time t and Ω represent the time at which the first machine other than ψ completes its required workload. Initially, $\psi = 1$ and $t = 0$. Define two sets of jobs: candidate jobs (θ) and unscheduled jobs (Π).

Step 1: *Initialisation*. Determine ψ and t.

Step 2: *Job ranking*. Set $\theta = \Pi$. For each job $j \in \theta$, compute ranking index $I_j(t)$:

$$I_j(t) = \left\{ \begin{array}{ll} \dfrac{w_j}{r_j - t + p_j} & r_j > t \\ \dfrac{w_j}{p_j} & r_j \le t \end{array} \right\}$$

The job with the maximum $I_j(t)$ value is assigned to machine ψ. Let this assigned job be job k. Compute the parameter α as follows:

$$\alpha = \left\{ \begin{array}{ll} t & r_k \le t \\ r_k & r_k > t \end{array} \right\}$$

Step 3: *Feasibility check*. If job k can be feasibly assigned to machine ψ at time α, go to Step 4. Otherwise, go to Step 5.

Step 4: *Job to machine assignment*. Assign job k to machine ψ, scheduling its processing to begin at time α. $\Pi = \Pi / k$. Go to Step 1.

Step 5: Candidate job update. $\theta = \theta \backslash k$. If $\theta = \{\emptyset\}$, go to Step 6. Otherwise, go to Step 2.

Step 6: *Time update*. If no job $j \in \Pi$ can be assigned to machine ψ at time t, the value of t must be updated to a future point in time at which conditions are feasible. Feasibility conditions change when (a) a new job arrives at the tool group and (b) another machine in the tool group becomes idle.

Since we update α with regard to job ready time in Step 2, we do not need to consider condition (a). Therefore, based on (b), set $t = \Omega$ and go to Step 2.

3.2 *Heuristic H2*

The purpose for the development of H2 is to examine another ranking index for use in Step 2 of H1. In H2, we compute the ranking index $I_j(t)$ for each job $j \in \theta$ as

$$I_j(t) = \left\{ \begin{array}{ll} 0 & r_j > t \\ \dfrac{w_j}{p_j} & r_j \le t \end{array} \right\}.$$

If no unscheduled job is ready at time t, then $I_j(t) = \{w_j / r_j - t + p_j\}$. The time update in Step 6 of H1 will only be valid if no unscheduled job is ready at time t. Hence, it is required to consider condition (a). If a job arrives to the tool group, update t to the earliest arrival time. Otherwise, set $t = \Omega$. All other steps in H1 are identical in H2.

4. Computational study

4.1 *Problem generation and parameters setting*

A variety of problem instances are examined to evaluate the efficacy of each heuristic in minimising *TWCT* on parallel machines subject to auxiliary resource constraints. First, two different levels are considered for the number of layer types, sampling from a discrete uniform distribution on the interval [1,v], $v \in \{3,6\}$. Experiments are also performed for both two and three machines operating in parallel (table 1). For each job, an integer processing time is generated from a discrete uniform distribution on [45, 75]. One-half of the job ready times are generated discrete integers on the

Table 1. Experimental design.

Factors	Levels	Level description
Number of machines (m)	2	2, 3
Number of jobs (n)	2	10, 15
Number of layers (l)	2	Discrete uniform DU[1, v], where $v \in \{3,6\}$

uniform interval [1, 360], while the remaining 50% of the jobs have $r_j = 0$. Further, job weights are distributed as discrete integer values on [1, 20].

Obtaining satisfactory feasible solutions is quite challenging as both the number of different layer types processed and the number of available reticles affects both *TWCT* solution quality and the required computation time. As optimal solution values are required for heuristic evaluation purposes, two different numbers of job levels are evaluated in our experimental design: 10 jobs and 15 jobs. A total of 10 problem instances are generated for each of the $2 \times 2 \times 2 = 8$ experimental combinations, thereby resulting in a total of 80 problem instances that will be evaluated by both proposed heuristics.

4.2 *Computational results*

Optimisation models are implemented in AMPL and solved by CPLEX to evaluate the solution performance of the optimisation model. The two heuristics are coded in Visual Basic for evaluation purposes. Let $TWCT(I, M, N, L, H)$ denote the total weighted completion time resulting from applying heuristic H to problem instance I, where instance I is characterised by M machines, N jobs, and L layer types. The performance ratio

$$PR(I, M, N, L) = \frac{TWCT(I, M, N, L, H) - TWCT^{*}(I, M, N, L)}{TWCT^{*}(I, M, N, L)} \times 100\%$$

is computed, where $TWCT^{*}(I, M, N, L)$ is the known optimal solution for instance I. The average performance ratio value computed over all 10 instances of each experimental factor combination is given in table 2. The first column lists an (m, n, l) triplet that corresponds to n jobs being scheduled on m machines for l different layer types. An asterisk (*) in the triplet signifies an average is computed across all instances for both levels of the experimental factor. Finally, the values in parentheses denote the number of times that a given heuristic produced the best overall solution for a problem instance at each factor level.

Table 2. Average performance ratio results.

(m, n, l)	H1	H2
(2, *, *)	4.70% (3)	1.23% (8)
(3, *, *)	5.70% (6)	2.22% (15)
(*, 10, *)	4.94% (8)	1.37% (17)
(*, 15, *)	5.46% (1)	2.08% (8)
(*, *, 3)	5.50% (7)	2.07% (17)
(*, *, 6)	4.90% (2)	1.38% (8)
Overall	5.20% (9)	1.72% (25)

Both heuristics require less than one second to resolve a problem instance. In contrast, the network optimisation model requires an average of 1.54 hours, with a best (worst) case of 0.0625 seconds (14.51 hours) to solve one of the problem instances of interest to optimality. The experimental results suggest that H2 produces schedules that are substantially better than H1 in terms of *TWCT*. In an attempt to promote an even better heuristic performance, a metaheuristic approach that can be used to post-process the schedule produced by H2 is examined.

4.3 *Post processing algorithm*

Tabu search (TS) is a metaheuristic procedure that was developed to find optimal or near-optimal solutions for a wide range of optimisation problems, including scheduling problems. Algorithm MTS is developed; a modified TS algorithm that attempts to search for optimal solutions within a pre-specified neighbourhood. In MTS, our neighbourhood is defined as all pairwise interchanges. As in conventional TS, MTS treats some number of the most recent job swaps (moves) as Tabu, meaning they cannot be reversed for a corresponding number of future iterations, to avoid local optimal solutions and cycling.

However, unlike conventional TS, MTS does not allow for moves to non-improving solutions. Preliminary experiments reveal that whenever MTS is allowed to move to feasible but non-improving solutions, solution quality degrades and the required computation time increases, often without ever having MTS return to a comparable objective function value. Therefore, algorithm MTS renews the schedule when a superior solution is found, compared with the best solution to date, in effect continuing its search as if this new schedule was in fact the initial schedule. Finally, in order to explore the neighbourhood more effectively, the criterion for deciding which jobs to interchange is based on the jobs' schedule position on the machines—jobs that are close to one another in the schedule (in terms of start time, for example) are selected to be interchanged.

Algorithm MTS

Step 1: *Initialisation*. Let Λ denote the current iteration and Γ denote the desired total number of iterations. Select an initial sequence (e.g. the one generated by heuristic H2).

Step 2: *Job selection*. Select candidate jobs for interchange.

Step 3: *Move assessment*. If the move is forbidden by the Tabu list or does not satisfy the choice criterion, go to Step 6. Otherwise, go to Step 4.

Step 4: *Update parameters*. Enter the acceptable move on the Tabu List. Swap the position of the selected jobs, leaving all other jobs' positions alone. Now, we must determine the starting and completion times of the selected jobs and the jobs following them on the same steppers. Let ω denote the set of selected jobs. In order to feasibly place these jobs on a given machine, a placement heuristic is required (see 'Placement heuristic' below).

Step 5: *Objective function assessment*. If the objective function value resulting from the proposed swap is better than the best objective function value found so far, then update the best schedule accordingly and go to Step 6.

Step 6: $\Lambda = \Lambda + 1$. If $\Lambda = \Gamma$, then STOP. Otherwise, go to Step 2.

Placement heuristic

Step 1: Find the job $j \in \omega$ that has an immediately preceding job $\varphi \notin \omega$ with minimum completion time.

Step 2: If job φ can be feasibly assigned at the completion time t of the preceding job, then $C_{\varphi} = t + p_{\varphi}$ and go to Step 5; else go to Step 3.

Step 3: If $r_{\varphi} > t$ and job φ can be feasibly assigned at time r_{φ}, then $C_{\varphi} = r_{\varphi} + P_{\varphi}$ and go to Step 5. Otherwise, go to Step 4.

Step 4: If $r_{\varphi} \leq \Omega$ and job φ can be feasibly assigned at time Ω (i.e., the completion time of the next finished job $j \notin \omega$), then $C_{\varphi} = \Omega + P_{\varphi}$ and go to Step 5; otherwise, set the objective function value of the current solution equal to big M and STOP.

Step 5: $\omega = \omega/\varphi$. If $\omega = \leq \{\emptyset\}$, then calculate the objective function value of the new solution and STOP; otherwise, go to Step 1.

Table 3 displays the experimental results when the post-processing algorithm MTS is seeded with the schedule from H2. For an added computational expense of six seconds (i.e. a total of seven seconds) the average *TWC* performance ratio decreases from 1.72% above optimal to 0.78% above optimal. In fact, H2 + MTS produces the best heuristic solution in 76 of 80 (95%) of the test cases. These results suggest that decision makers can produce even better heuristic solutions to the parallel machine scheduling problem with auxiliary resource constraints if they can afford a slight increase in required computation time. These improved solutions can hopefully help semiconductor wafer fab personnel to utilise more effectively their most critical production resources, the photolithography steppers.

Table 3. PR improvement due to algorithm MTS.

(m, n, l)	H2	H2 + MTS
(2, *, *)	1.23% (8)	0.73% (38)
(3, *, *)	2.22% (15)	0.83% (38)
(*, 10, *)	1.37% (17)	0.40% (37)
(*, 15, *)	2.08% (8)	1.15% (39)
(*, *, 3)	2.07% (17)	1.06% (36)
(*, *, 6)	1.38% (8)	0.50% (40)
Overall	1.72% (25)	0.78% (76)

5. Conclusions and future research

The critically important problem of scheduling photolithography machines in semiconductor wafer fabrication facilities is investigated in this paper. Existing approaches and assumptions described in the literature are expanded, and new, effective heuristic solution procedures are developed. The schedule produced by our most effective heuristic, H2, is improved upon further through the use of a post-processing local search procedure. This procedure selects appropriate neighbourhoods according to job positions in order to reduce the computational effort required to reach improving solutions. In this placement heuristic, care is taken to only search for improving job schedule positions for a limited amount of time, as unnecessary forced machine idle times can adversely increase the *TWCT* objective function value.

In our future research efforts, our metaheuristic algorithms will be extended to determine methods for designing solution neighbourhoods more effectively. In addition, some wafer fabs have multiple reticles available for one or more process layers. Additional research will focus on extending our initial research findings to this case, as added auxiliary resources should provide for the creation of more effective production schedules. Due-date related objective performance measures, such as total weighted tardiness, will also be examined, along with the corresponding changes to the experimental design to investigate various levels of due date tightness and due date range. The ATC dispatching rule of Vepsalainen and Morton (1987) can be applied to feasibly sequence unscheduled jobs whenever a machine becomes idle under a due-date related performance measure.

References

Akcali, E. and Uzsoy, R., A sequential solution methodology for capacity allocation and lot scheduling problems for photolithography. *IEEE/CPMT International Electronics Manufacturing Technology Symposium*, 2000, 374–381.

Akcali, E., Nemoto, K. and Uzsoy, R., Cycle-time improvements for photolithography process in semiconductor manufacturing. *IEEE Trans. Semicon. Manuf.*, 2001, **14**(1), 48–56.

Arisha, A. and Young, P., Intelligent simulation-based lot scheduling of photolithography toolsets in a wafer fabrication facility, in *Proceedings of the 2004 Winter Simulation Conference*, 2004, pp. 1935–1942.

Bokhorst, J.A.C., Slomp, J. and Gaalman, G.J.C., On the who-rule in dual resource constrained (DRC) manufacturing systems. *Int. J. Prod. Res.*, 2004, **42**(23), 5049–5074.

El-Maraghy, H., Patel, V. and Abdallah, I.B., Scheduling of manufacturing systems under dual-resource constraints using genetic algorithms. *J. Manuf. Syst.*, 2000, **19**(3), 186–201.

Gargeya, V.B. and Deane, R.H., Scheduling research in multiple resource constrained job shops: A review and critique. *Int. J. Prod. Res.*, 1996, **34**(8), 2077–2097.

Gargeya, V.B. and Deane, R.H., Scheduling in the dynamic job shop under auxiliary resource constraints: a simulation study. *Int. J. Prod. Res.*, 1999, **37**(12), 2817–2834.

Kempf, K.G., Uzsoy, R. and Wang, C., Scheduling a single batch processing machine with secondary resource constraints. *J. Manuf. Syst.*, 1998, **17**(1), 37–51.

Kim, S., Yea, S. and Kim, B., Shift scheduling for steppers in the semiconductor wafer fabrication process. *IIE Trans.*, 2002, **34**, 167–177.

Lawler, E.L., Lenstra, J.K. and Rinnooy Kan, A.H.G., Recent developments in deterministic sequencing and scheduling: a survey. In *Deterministic and Stochastic Scheduling*, edited by M.A.H. Dempster, J.K. Lenstra and A.H.G. Rinnooy Kan, pp. 35–74, 1982 (Dordrecht: Reidel).

Lee, H.Y., Park, J. and Kim, S., Experimental study on input and bottleneck scheduling for a semiconductor fabrication line. *IIE Trans.*, 2002, **34**, 179–190.

Lenstra, J.K., Rinnooy Kan, A.H.G. and Brucker, P., Complexity of machine scheduling problems. *Ann. Disc. Math.*, 1977, **1**, 343–362.

Nonas, S.L. and Olsen, K.A., Optimal and heuristic solutions for a scheduling problem arising in a foundry. *Comput. Oper. Res.*, 2005, **32**, 2351–2382.

Park, S., Fowler, J., Carlyle, M. and Hickie, M., Assessment of potential gains in productivity due to proactive reticle management, in *Proceedings of the 1999 Winter Simulation Conference*, 1999, pp. 856–864.

Treleven, M., A review of the dual resource constrained system research. *IIE Trans.*, 1989, **21**(3), 279–287.

Vepsalainen, A. and Morton, T.E., Priority rules for job shops with weighted tardiness costs. *Manage. Sci.*, 1987, **33**, 1035–1047.

Eray Cakici is a Logistics Engineer II at Transplace. He received his MSc degree in Industrial Engineering from the University of Arkansas in 2005 and a BSc in Industrial Engineering from Baskent University, Turkey, in 2004. He is currently a PhD student at the University of Arkansas. His doctoral studies mainly focus on transportation logistics and production scheduling.

Scott J. Mason is an Associate Professor and the Associate Department Head of Industrial Engineering at the University of Arkansas, as well as the Chair of Industrial Engineering Graduate Studies. He received his Bachelor's and Master's degrees from The University of Texas at Austin and his PhD in Industrial Engineering from Arizona State University. His research and teaching interests include production planning and control, scheduling and semiconductor manufacturing. He is a member of INFORMS and a senior member of the Institute for Industrial Engineers.