Daily Scheduling for R&D Semiconductor Fabrication

Da-Yin Liao, *Member, IEEE*, Shi-Chung Chang, *Member, IEEE*, Kuo-Wei Pei, and Chi-Ming Chang, *Member, IEEE*

Abstract— **This paper presents the development of a daily scheduling tool, Electronic Research & Service Organization Fab Scheduler (ERSOFS), for a research and development (R&D) pilot line of semiconductor wafer fabrication. An integer programming problem formulation is first given, which captures the salient features such as high variety and very low volume, cyclic process flows, batching at diffusion machines, single mask for each photolithography operation, loop test and engineering splitting and merging of wafer lots. A solution methodology based on Chang and Liao's approach [8] for scheduling flexible flow shops is then extended to this class of problems. The solution methodology is implemented and validated in an R&D fab. Results indicate that ERSOFS efficiently generates schedules of high quality. The rescheduling function of ERSOFS provides fast and smooth adjustments of schedules to cope with the high production uncertainties in an R&D fab. Analysis of the algorithmic properties demonstrates the potential of ERSOFS for application to larger fabs.**

I. Introduction

INTEGRATED circuit wafer fabrication is a business of high investment, high technology, and fierce competition. It involves perhaps the most complex manufacturing process ever used. In an IC fab, there may be dozens of fabrication processes. Each process may contain 200–300 processing steps and more than 100 machines are involved. There exist high uncertainties in operations due to frequent machine failures and fluctuation of yield rates. Production planning and control of IC wafer fabrication is thus quite complicated and particularly difficult. It is still a very challenging research topic to develop sound production planning and control in an IC fab [13], [23].

Major production control issues in an IC fab include 1) *wafer release* of raw wafers into the fab, 2) *daily scheduling*, and 3) *lot dispatching* to determine which lot to process when a machine becomes available. Wafer release [12], [18] aims at controlling the wafer-in-process (WIP) level and fabrication cycle times. It is calculated using a day or a week as a time unit over a long time horizon of two to four months. As a byproduct, daily or weekly wafer outputs, i.e., production targets, can also be determined. Lot dispatching [10] ranks lot priorities of processing and is done on the shop floor to respond properly to fab status in real time. Daily scheduling bridges between these two aforementioned functions. It breaks daily production targets into a production schedule in a time scale of 1–3 h over a day, which then serves as a guideline for dispatching. The need for effective coordination of daily operations in a fab is significant.

Wein [24] has pointed out that scheduling has a significant impact on the performance of semiconductor wafer fabrication. Bai *et al.* [2] presented a hierarchical production planning and scheduling system for a semiconductor wafer fab by using hierarchical decomposition and production flow control methods where wafer movements are treated as continuous flows. Similarly, a fluid network model was adopted by Connors *et al.*, for scheduling semiconductor lines of high-production volume [9]. Lu *et al.* [20] analyzed several distributed policies for scheduling a large semiconductor manufacturing facility. They identified the two best policies: one for minimizing the mean delay, or equivalently, the mean work-in-process, and the other for minimizing the variance of delay, which are considered to be important performance measures for semiconductor manufacturing. Lozinski *et al.* [17] used bottleneck starvation indicators to implement a bottleneck starvation avoidance policy for shop floor scheduling of IC fabs. Bitran *et al.* [6] developed a scheduling system for a wafer fabrication facility. They decompose the facility into many smaller shops with homogeneous product mix which leads to a simple scheduling problem and enables the use of relatively simple heuristics. Although there have been investigations focusing their attention on the scheduling problems of IC fabs where various specialty wafers of small volume and customer-specific orders are manufactured [5], few of the existing results address the distinct production control problems in a research and development (R&D) IC fab.

There are many commercially available *area* (intrabay) dispatching/scheduling tools, but there is a lack of effective *fab* (interbay) schedulers for semiconductor fabrication. The short interval scheduling (SIS) package, included in the management information system Comets developed by Consilium Inc., implements many priority rules considering the WIP level at each machine area and/or the WIP level of its immediately upstream or downstream machines. The PROMIS system of PROMIS Systems Corporation comprises integrated production planning and dispatching modules. The philosophy behind these modules is to achieve a balanced line. For each processing step, target inventories are set. If the inventories are at the target levels, then the line is considered in balance. Production planning determines production targets to achieve

Manuscript received December 10, 1993; revised May 29, 1996. This work was supported in part by the National Science Council, R.O.C., under Grants NSC81-0416-E002-567 and NSC82-0416-E002-366, and by ERSO of ITRI under Contracts T0-81014 and S0-83002.

D.-Y. Liao and S.-C. Chang are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

K.-W. Pei and C.-M. Chang are with the Vanguard International Semiconductor Corporation, Hsinchu, Taiwan, R.O.C.

Publisher Item Identifier S 0894-6507(96)08114-6.

line balance while dispatching tries to meet these targets. However, practitioners need to break these production targets into schedules of finer resolution for daily shop floor control. It is still a challenging issue both in theory and in practice to develop an effective daily scheduling methodology which dynamically schedules with global information of the fab.

In this paper, the daily scheduling problem of an R&D IC fab is investigated. An R&D IC fab has the salient characteristics of 1) wafers of very small volume but high varieties; 2) many dedicated machines without backup; 3) high production uncertainties such as frequent machine failures and tune-ups, engineering experiments, lot holdings and releases, process changes, inspections, and reworks; and 4) little historical data available for a new process. Similar to other IC fabs, in an R&D fab, a few wafers of the same processing requirements are stored in a cassette as a lot and as a unit of processing. The fabrication of different lots may require processing by the same machine. There are many revisits to a machine by the same lot due to the layered nature of IC fabrication. That is, there are cycles in production flow paths. Lots with the same operating conditions are usually batched together for a diffusion operation at a furnace since it takes a relatively long processing time. For each photolithography step, there is only a single mask available for processing. Although there may be a few photolithography machines available, only one lot with the required mask can be processed by one machine at a time.

In an R&D pilot line, cycle time reduction is very important for speeding up the learning curve of process development [7]. Although Wein [24] claimed that wafer release has the largest effect on cycle time, Lu *et al.* [20] have shown that short interval scheduling/dispatching also has a significant effects on reducing both the mean and variance of cycle time. In this paper, wafer release and out schedules are assumed given. The daily scheduling function tries to schedule effective wafer movements such that the right kind of wafers are processed at the right time in order to meet the delivery schedule and to reduce the cycle times of fabrication.

To reflect the effectiveness of wafer movements, each lot is given a weighting factor for each processing step completed in the day. The weighting factor for a lot is determined based on the slack time (due date—estimated residual cycle time of the lot), and it may vary day by day. The daily scheduling problem for an R&D pilot line is then to maximize, under a given set of daily wafer release schedules, the total weighted movements of wafer lots in the fab while satisfying constraints of 1) equipment capacity, 2) batching, 3) single-mask, and 4) precedence relationship of fabrication processes. Such a daily scheduling problem is formulated as an integer programming problem in this paper.

There are relatively high random disturbances in an IC fab. Typical disturbances include machine failure, material shortage, lot holding/releasing and production process change, etc. As these uncertainties may have major effects on the daily operations of a fab, it is very important to cope with these uncertainties in daily scheduling.

Our scheduling methodology extends the approach developed in [8] for scheduling flexible flow shops to the scheduling of R&D IC pilot lines. It consists of three parts: 1) an efficient algorithm for developing a near-optimal daily schedule under nominal conditions (i.e., a nominal daily schedule); 2) fast rescheduling for timely adjustment of the nominal daily schedule to cope with disturbances; and 3) periodic rescheduling using the "rolling horizon scheme" and the nominal scheduling algorithm of 1). The fast rescheduling in 2) applies a neighborhood search to the nominal schedule [1], exploits the structure of the nominal fab scheduling approach, and aims at a quick and reasonably graceful response. Fig. 1 depicts the schematic diagram of nominal scheduling and fast rescheduling and Fig. 2 demonstrates the rolling horizon philosophy for periodic rescheduling.

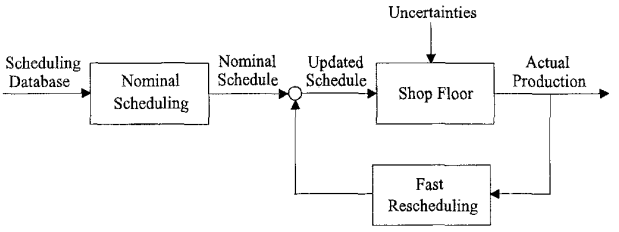


Fig. 1. Open-loop feedback optimal control for scheduling with uncertainties.

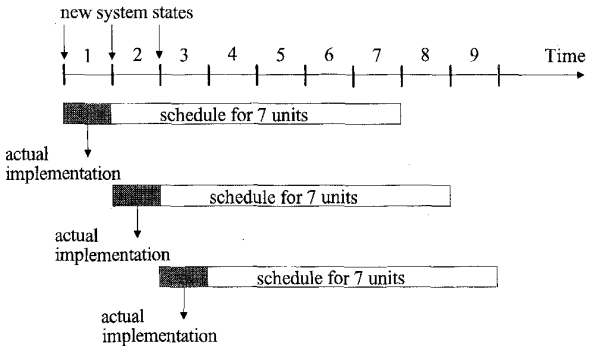


Fig. 2. Rolling horizon philosophy for peridical rescheduling.

A fab scheduler, ERSOFS, that implements our scheduling methodology is developed for daily or short-term scheduling of the Submicron Laboratory of Electronics Research and Service Organization (ERSO), Industrial Technology Research Institute. This tool has been validated by using field data from Sept. 15–Oct. 16, 1993. Results indicate that ERSOFS obtains schedules of high quality (<5% from the optimal) using very reasonable amounts of CPU time (<8 min on a SUN SPARC II workstation) for application to daily scheduling. ERSOFS provides schedules that are distinctly better than those generated by field engineers: ERSOFS uniformly schedules more movements for each hot lot (lots of first priority) than those of the actual schedules. The movements predicted by ERSOFS's schedule are better than actual production which was performed by engineers using the *ad hoc* scheduling. Analysis of algorithmic properties of ERSOFS demonstrates the potential of the methodology for application to mass-production fabs. Numerical experimentation results indicate that our fast rescheduling algorithms are computationally very efficient (<15 s) and still result in good schedules with smooth adjustments from the original one.

The remainder of this paper is organized as follows. In Section II, the daily scheduling problem of R&D fabs is first formulated. Development of a baseline solution methodology for the scheduling problem is presented in Section III. Further algorithmic developments for coping with production uncertainties are given in Section IV. Section V briefly describes the implementation of the fab scheduler ERSOFS, its validation results and the assessment of its potential. Finally, Section VI concludes this paper.

II. Scheduling Problem Formulation

Consider an IC fab which fabricates semiconductor wafers for the purpose of research and development (R&D). There are machine groups of various functionality. Each machine group may consist of a few homogeneous machines. Wafers of the same processing requirements are stored in a cassette as a lot, which has a maximum of 24 wafers. There are usually tens of wafer types in an R&D pilot line, each having a small volume of lots. The process flow of each lot can be organized as a sequence of stages. A stage comprises a few processing steps, whose processing requirements and conditions are specified by recipes. Different lots and stages may use the same machine group. There are many revisits to a machine group in the process flow of a lot due to the layered nature of semiconductor fabrication. That is, there are cycles in production flow paths among machine groups. As there are relatively large buffer spaces on the shop floor, buffer space availability does not pose a significant constraint on production flows.

Let us define some notation for modeling such an R&D IC pilot line.

Notation:

S_i	Wafer type index.
i	Total number of stages for type-i wafers.
$(i,\ s)$	Stage index for type-i wafers, $s = 1, \cdots, S_i$.
P_{is}	Processing time of stage $(i,\ s)$.
m	Machine group index.
M_{is}	Machine group where stage $(i,\ s)$ can be processed.
T	The scheduling horizon.
t	Time index, $t = 1, \cdots, T$.
C_{mt}	Capacity of machine group m at time t.
l_{it}	Number of type-i wafer lots released at the beginning of time t.
r	Recipe index.
P_r	Processing time of recipe r.
R_{is}	Recipe for processing stage $(i,\ s)$.
N_r	Diffusion machine group where diffusion recipe r can be processed.
$\overline{B}(\underline{B})$	Maximum (minimum) number of lots in a batch for a diffusion machine.
ψ_{is}	Weighting factor for processing a lot at stage $(i,\ s)$.
X_{ist}	Number of type-i lots ready for processing stage $(i,\ s)$ at the beginning of time t.
U_{ist}	Number of type-i lots to be loaded onto machines for processing stage $(i,\ s)$ at time t.
b_{rt}	Number of batches to be formed for processing recipe r at time t.
$DIFF$	Set of diffusion machine groups.
$PHOTO$	Set of photolithography machines.

Among the above, the daily wafer release schedules l_{it}, machine capacities C_{mt}, process flow data and the weighting factors ψ_{is} are given as inputs, while variables u_{ist} and b_{rt} correspond to the scheduling decisions.

Based on field engineers' descriptions and data availability, we assume in this paper that

1) Setup time for changing lot types of production at each machine can be estimated and incorporated as part of the processing time of each lot.
2) Setup cost is negligible.
3) Lot transportation time and cost from one machine group to the other are negligible.
4) There are no rework and scrap of wafers during the day to be scheduled.

In the process flow of type-i wafers, lots loaded for stage $(i,\ s-1)$ processing at time $t - P_{i(s-1)}$ will be finished after $P_{i(s-1)}$ time units and constitute the inflow of the buffer for stage $(i,\ s)$ at time t. These inflow lots plus those originally in the buffer for stage $(i,\ s)$ processing at the beginning of time t minus the lots loaded for processing stage $(i,\ s)$ at time t form the lots ready for processing stage $(i,\ s)$ at the beginning of the next time, $t+1$. The flows of lots therefore satisfy the following flow balance equations.

Flow Balance Equations for type-i wafers, $\forall\, i$:

$$X_{i1(t+1)} = X_{i1t} - u_{i1t} + l_{it}, \quad \forall\, t \tag{2.1a}$$

$$X_{is(t+1)} = X_{ist} - u_{ist} + u_{i(s-1)(t-P_{i(s-1)})}, \quad \forall\, s = 2, 3, \cdots, S_i, \quad \forall\, t \tag{2.1b}$$

$$X_{i(S_i+1)(t+1)} = X_{i(S_i+1)t} + u_{iS_i(t-P_{iS_i})}, \quad \forall\, t \tag{2.1c}$$

with the wafer release schedules $\{l_{it}, \forall\, t\}$, and the initial wafers in process (WIP) $\{X_{i1t}, \forall\, t\}$ and $\{u_{is(-P_{is}+1)}, u_{is(-P_{is}+2)}, \cdots, u_{is0}, \forall\, s\}$ given. Note that wafer releases form the inflows of the first stage $(i,\ 1)$, $\forall\, i$. A pseudo stage $(i,\ S_i+1)$ is used to represent the stock of completed type-i lots with variable $X_{i(S_i+1)t}$ denoting the cumulative number of finished lots up to time t. The precedence relationship among operations of each process flow is captured implicitly in these flow balance equations.

Since a diffusion operation takes a relatively long processing time and a diffusion machine is designed to process many wafers at a time, lots of the same operating conditions, i.e., lots requiring the same recipe, are usually batched together for diffusion. For a recipe r, $\sum_{(i,s),R_{is}=r} u_{ist}$ is the total number of lots that are batched together for processing at time t. As there are limitations on both minimum ($\underline{B}$) and maximum ($\overline{B}$) numbers of lots to form a batch, the total number of lots batched for recipe r processing must satisfy

Batching Constraints:

$$\underline{B} \cdot b_{rt} \le \sum_{\substack{(i,\ s) \\ R_{is}=r}} u_{ist} \le \overline{B} \cdot b_{rt}, \quad \forall\, r \quad \text{and} \quad \forall\, t. \tag{2.2}$$

The processing capacity of diffusion machines is also expressed in the unit of batches. For recipe r, the b_{rt} batches commencing their diffusion operation on diffusion machines need P_r time units to complete the processing. There is a total of $\sum_{r,N_r=m} \sum_{\tau=t-P_r+1}^{t} b_{r\tau}$ batches being processed by diffusion machine group m. This quantity can not exceed its capacity at each time, i.e.,

Machine Capacity Constraints (Diffusion Machines):

$$\sum_{\substack{r \\ N_r=m}} \sum_{\tau=t-P_r+1}^{t} b_{r\tau} \le C_{mt}, \quad \forall m \in DIFF \quad \text{and} \quad \forall t. \tag{2.3a}$$

Similarly, for a nondiffusion stage (i, s), the quantity $\sum_{(i,s),\, M_{is}=m} \sum_{\tau=t-P_{is}+1}^{t} u_{is\tau}$ must not exceed the processing capacity of machine group m.

Machine Capacity Constraints (Nondiffusion Machines):

$$\sum_{\substack{(i,s) \\ M_{is}=m}} \sum_{\tau=t-P_{is}+1}^{t} u_{is\tau} \le C_{mt}, \quad \forall m \notin DIFF \quad \text{and} \quad \forall t. \tag{2.3b}$$

There is only one mask available for each photolithography stage, which in turn limits the machine capacity for processing it to at most one machine at a time, i.e.,

Single Mask Constraints:

$$u_{ist} \le 1, \quad \forall m = M_{is} \in PHOTO \quad \text{and} \quad \forall t. \tag{2.4}$$

Frequent lot splitting and lot merging for process development purposes are two important features in an R&D IC pilot line. Let $\mathcal{M}$ be a set of stages where several lots are merged into one lot. Define $S^{\mathcal{M}}_{i's'} \equiv \{(i, s)\}$ as the set of processing stages whose finished lots will be merged into one type-i' lot for further processing of stage (i', s'). Let Y_{ist} be the number of lots which have completed stage (i, s) and are ready for stage (i', s') processing at time t. When a lot is formed and loaded for processing at stage (i', s'), it takes one lot from the buffer of each stage $(i, s) \in S^{\mathcal{M}}_{i's'}$. So,

Flow Balance Equations for Merging:

$$Y_{is(t+1)} = Y_{ist} - u_{i's't} + u_{is(t-P_{is})}, \quad \forall (i, s) \in S^{\mathcal{M}}_{i's'} \text{ and } (i', s') \in \mathcal{M}, \quad \forall t. \tag{2.5}$$

Furthermore, the number of merged lots cannot exceed what are available for merging,

Merging Constraints:

$$u_{i's't} \le Y_{ist}, \quad \forall (i, s) \in S^{\mathcal{M}}_{i's'} \text{ and } (i', s') \in \mathcal{M}, \quad \forall t. \tag{2.6}$$

The splitting of a lot is just the reverse of merging, where one lot becomes a few lots of different types. Let $\mathcal{S}$ be a set of stages where one lot is split into several lots after being processed. Define $S^{\mathcal{S}}_{i's'} \equiv \{(i, s)\}$ as the set of stages whose input lots are from a splitting stage (i', s'). When a lot is split by stage (i', s'), the buffer level of each $(i, s) \in S^{\mathcal{S}}_{i's'}$ increases by one after the splitting, i.e.,

Flow Balance Equations for Splitting:

$$X_{is(t+1)} = X_{ist} - u_{ist} + u_{i's'(t-P_{i's'})}, \quad \forall (i, s) \in S^{\mathcal{S}}_{i's'} \text{ and } (i', s') \in \mathcal{S}, \quad \forall t. \tag{2.7}$$

The production flow of type-i' lot created after a merging or splitting stage then follows the same set of flow balance equations as (2.1). Our model here does not include unexpected splitting or conditional splitting that depends on the processing result of a stage.

A loop test bears much resemblance to the planned splitting and merging. Processing of loop test lots is initiated by the completion of certain stages of a few different regular production lots, which can be viewed as a pseudo merging. When a loop test process finishes, regular production lots may resume their individual processing flows, which corresponds to a splitting. The only difference with merging/splitting is that a loop test uses its own wafer lots and the relevant production lots are on hold. The techniques for modeling merging and splitting can therefore be applied to model the production flow of loop test lots after a minor extension.

Unlike mass production IC fabs, engineering experimentation and inspection stages are frequently added to the original processes in a pilot line on a daily basis. Such a feature can be easily handled by updating the process flow database and requires no extra modeling efforts. However, the processing time data for these stages is usually estimated roughly by experienced engineers.

As the production unit is either of a lot or a batch, the following integrality constraints should also be satisfied.

Integrality Constraints:

$$u_{ist},\ X_{ist},\ \text{and } b_{rt} \text{ are nonnegative integers}, \quad \forall (i, s), \quad \forall r \text{ and } \forall t. \tag{2.8}$$

In a pilot line, importance or priorities of production are different among different wafer types and stages during one day. Our objective of daily scheduling is to maximize the total weighted production (or moves) of wafer lots in the fab, where a *move* is defined to be a completion of one stage of a lot. The daily scheduling problem is then to find an allocation of processing capacity over one day that maximizes the weighted moves while satisfying all production constraints. Mathematically, it is formulated as

$$\max_{\mathbf{u},\mathbf{b}} \sum_{(i,s)} \sum_{t} \psi_{is} u_{ist} \quad \text{subject to (2.1)–(2.8)}$$

or equivalently,

$$(P) \qquad \min_{\mathbf{u},\mathbf{b}} -\sum_{(i,s)} \sum_{t} \psi_{is} u_{ist} \quad \text{subject to (2.1)–(2.8).}$$

Note that the selection of weighting factors ψ_{is} can also depend on the desired effective moves and can be achieved by human and long term considerations. It makes it easier for production engineers to identify the desired effective moves during the day.

III. A Baseline Solution Methodology

The scheduling problem (P) formulated in Section II is an integer programming problem of NP-hard computational complexity [22]. Chang *et al.* [8] have developed an effective approach for scheduling flexible flow shops by using Lagrangian relaxation and network flow techniques. A simple but detailed example illustrating the basic idea of the formulation and solution methodology can be found in [8]. As an IC fabrication line is basically a flexible flow shop but with a reentrant feature, this section develops a near-optimal and computationally efficient solution algorithm for the daily scheduling problem (P) by extending the approach of [8]. To convey the key ideas and simplify the discussion, merging and splitting constraints (2.5)–(2.7) are excluded from the following developments.

A. Lagrangian Relaxation and Decomposition

One major difficulty for solving the scheduling problem (P) is caused by the competition among different wafer lots and stages for limited processing resources. If there were infinite machine capacity and there was no need to form batches, each lot could be scheduled independently and the scheduling problem then becomes quite trivial. Motivated by such an observation, we apply Lagrangian relaxation [8] and [14] to relax the machine capacity constraints (2.3a) and (2.3b) and the batching constraints (2.2), and form the Lagrangian function as

$$
\begin{aligned}
&-\sum_{(i,s)}\sum_{t}\psi_{is}u_{ist}\\
&+\sum_{\substack{m\\ m\notin DIFF}}\sum_{t}\lambda_{mt}\cdot\left[\sum_{\substack{(i,s)\\ M_{is}=m}}\sum_{\tau=t-P_{is}+1}^{t}u_{is\tau}-C_{mt}\right]\\
&+\sum_{\substack{m\\ m\in DIFF}}\sum_{t}\pi_{mt}\cdot\left(\sum_{\substack{r\\ N_r=m}}\sum_{\tau=t-P_r+1}^{t}b_{r\tau}-C_{mt}\right)\\
&+\sum_{r}\sum_{t}\mu_{rt}\cdot\left[\underline{B}\cdot b_{rt}-\sum_{\substack{(i,s)\\ R_{is}=r}}u_{ist}\right]\\
&+\sum_{r}\sum_{t}\nu_{rt}\cdot\left[\sum_{\substack{(i,s)\\ R_{is}=r}}u_{ist}-\overline{B}\cdot b_{rt}\right]
\end{aligned}
$$

where $\{\lambda_{mt}\}$, $\{\pi_{mt}\}$, $\{\mu_{rt}\}$, and $\{\nu_{rt}\}$ are the associated Lagrange multipliers which are nonnegative real numbers.

Let $\mathbf{u}_i \equiv \{u_{imt}, \quad \forall m \text{ and } t\}$, $\lambda \equiv \{\lambda_{mt}, \quad \forall m \notin DIFF\}$, $\pi \equiv \{\pi_{mt}, \quad \forall m \in DIFF\}$, $\mu \equiv \{\mu_{rt}\}$, and $\nu \equiv \{\nu_{rt}\}$. Define for type-i wafers

$$
\begin{aligned}
&PL_i(\mathbf{u}_i, \lambda, \mu, \nu) \equiv\\
&\sum_{s}\sum_{t}\left[\sum_{\substack{m\\ m=M_{is}}}\sum_{\tau=t}^{t+P_{is}-1}\lambda_{m\tau}-\psi_{is}+\sum_{\substack{r\\ r=R_{is}}}(\nu_{rt}-\mu_{rt})\right]\\
&\cdot u_{ist}, \qquad (3.1)
\end{aligned}
$$

and for recipe r at time t

$$
BL_{rt}(b_{rt}, \pi, \mu, \nu) \equiv \left(\sum_{m=N_r}\sum_{\tau=t}^{t+P_r-1}\pi_{m\tau}+\underline{B}\cdot\mu_{rt}-\overline{B}\cdot\nu_{rt}\right)b_{rt}. \qquad (3.2)
$$

The dual problem obtained after relaxing problem (P) is

(D)

$$
\begin{aligned}
\max_{\substack{\lambda\geq 0,\ \pi\geq 0\\ \mu\geq 0,\ \nu\geq 0}}\Bigg\{&\Phi(\lambda, \pi, \mu, \nu) \equiv \sum_{i}\min_{\mathbf{u}_i}PL_i(\mathbf{u}_i, \lambda, \mu, \nu)\\
&+\sum_{r}\sum_{t}\min_{b_{rt}}BL_{rt}(b_{rt}, \pi, \mu, \nu)\\
&-\sum_{\substack{m\\ m\notin DIFF}}\sum_{t}\lambda_{mt}C_{mt}-\sum_{\substack{m\\ m\in DIFF}}\sum_{t}\pi_{mt}C_{mt}\Bigg\}
\end{aligned}
$$

subject to (2.1), (2.4), and (2.8).

Note that for a given set of Lagrange multipliers (λ, π, μ, ν), there are two classes of independent subproblems in (D), which correspond to production scheduling and determination of batch sizes without capacity limitation respectively:

1) production scheduling subproblem for type-i wafers

$(PS-i)$

$$
\min_{\mathbf{u}_i}PL_i(\mathbf{u}_i, \lambda, \mu, \nu)
$$

subject to (2.1), (2.4) and (2.8); and

2) batch size determination subproblem for recipe r at time t

$(BA-rt)$

$$
\min_{b_{rt}}BL_{rt}(b_{rt}, \pi, \mu, \nu)
$$

subject to (2.8).

After the relaxation, competition for machines among type-i lots of different stages due to the reentrant feature of production flow no longer exists in a subproblem $(PS-i)$ either. Each subproblem is a much simpler scheduling/batching problem than (P).

B. Solution Algorithms for Subproblems

1) Network Flow Algorithm for Production Scheduling in $(PS-i)$: The set of flow balance equations (2.1) of $(PS-i)$ render themselves naturally to a network representation, which is a graph consisting of nodes and arcs. Let a node n_{st} correspond to the buffer for stage (i, s) at time t, $s = 1, \cdots, S_i+1$ and $t = 1, \cdots, T$. Since it takes P_{is} time units of processing for a lot to go from the buffer of stage (i, s) to the buffer of stage $[i, s+1)$, an arc $[n_{st}, n_{(s+1)(t+P_{is})}]$ is formed to represent a path for the production flow u_{ist} between these two buffers. The number of lots that are carried over in the buffer of stage (i, s) from time t to time $t+1$ is $X_{is(t+1)}$. These lots flow through an arc $[n_{st}, n_{s(t+1)}]$. In addition, a

source node **S** and a sink node **T** are added to the network. Node **S** is connected by arcs to node n_{s1} with a flow equal to the initial buffer level X_{is1}, $\forall s$, to node $n_{s(P_{is}-\tau)}$ with a flow equal to $u_{is(-\tau)}$, $\tau = 0, \cdots, P_{is} - 1$ and $\forall\, s$, and to node n_{1t} with a flow equal to released quantity l_{it}, $\forall t$. Node **T** serves as a sink of flows, which is connected by an arc from a node $n_{(S_i+1)t}$ with flow $u_{iS_i(t-P_{iS_i})}$. The single mask constraints (2.4) form bounds on arc flows that correspond to production flows of photolithography stages. It can be easily verified that the flow conservation at each node represents one of the flow balance equations.

As the cost function of $(PS - i)$ is linear, by properly summing the cost coefficients, the associated arc cost for arc $(n_{st}, n_{(s+1)(t+P_{is})})$ is $\sum_{m=M_{is}} \sum_{\tau=t}^{t+P_{is}-1} \lambda_{m\tau} - \psi_{is} + \sum_{r=R_{is}} (\nu_{rt} - \mu_{rt})$, $\forall\, s$ and $\forall t$, and zero for other arcs. Thus, subproblem $(PS - i)$ is essentially a minimum cost linear network flow (MCLNF) problem, whose integer optimal solution can be obtained by polynomial time algorithms [21]. Our implementation adopts the RELAX code developed by Bertsekas *et al.* [4] to solve $(PS - i)$.

2) Algorithm for Batch Size Determination in $(BA - rt)$: Each subproblem $(BA - rt)$ is a simply constrained, linear integer programming problem. Under a given set of $\{\pi_{mt}\}$, μ_{rt}, ν_{rt}, and u_{ist}, a batch size b_{rt} is determined according to the complementary slackness condition [19] of $(BA - rt)$ as follows:

1) if $\left(\sum_{m=N_r} \sum_{\tau=t}^{t+P_r-1} \pi_{m\tau} + \underline{B} \cdot \mu_{rt} - \overline{B} \cdot \nu_{rt}\right) > 0$ then $b_{rt} = 0$;

2) if $\left(\sum_{m=N_r} \sum_{\tau=t}^{t+P_r-1} \pi_{m\tau} + \underline{B} \cdot \mu_{rt} - \overline{B} \cdot \nu_{rt}\right) < 0$ then $b_{rt} = C_{mt}$;

3) if $\left(\sum_{m=N_r} \sum_{\tau=t}^{t+P_r-1} \pi_{m\tau} + \underline{B} \cdot \mu_{rt} - \overline{B} \cdot \nu_{rt}\right) = 0$ then $b_{rt} = \left\lceil \frac{\sum_{\substack{(i,s) \\ R_{is}=r}} u_{ist}}{\overline{B}} \right\rceil$

where $\lceil x \rceil$ is the largest integer smaller or equal to x.

Namely, an integer-valued b_{rt} in $[0, C_{mt}]$, where $m = N_r \in DIFF$, is chosen so that $(\sum_{m=N_r} \sum_{\tau=t}^{t+P_r-1} \pi_{m\tau} + \underline{B} \cdot \mu_{rt} - \overline{B} \cdot \nu_{rt}) \cdot b_{rt} = 0$ and the Lagrangian function is minimized. When $(\sum_{m=N_r} \sum_{\tau=t}^{t+P_r-1} \pi_{m\tau} + \underline{B} \cdot \mu_{rt} - \overline{B} \cdot \nu_{rt}) = 0$, b_{rt} is set to the possibly smallest integer to take the advantages of small batch production.

C. Subgradient Algorithm for the Dual Problem

The dual function $\Phi(\lambda, \pi, \mu, \nu)$ is not differentiable because of the integrality constraints (2.8). A subgradient method of [11] and [15] is adopted to iteratively solve the dual problem (D). Let $\Theta \equiv [\lambda, \pi, \mu, \nu]$ be a vector of Lagrange multipliers. Multiplier vector Θ is then updated by

$$\Theta^{k+1} = \Theta^k + \alpha^k \mathbf{g}(\Theta^k) \tag{3.3}$$

between the kth and $(k+1)$th iterations, where $\mathbf{g}(\Theta^k)$ is the subgradient of Φ and α^k is the step size of the kth iteration with

$$\alpha^k = \gamma \frac{\overline{\Phi}^* - \Phi(\Theta^k)}{\mathbf{g}(\Theta^k)^T \mathbf{g}(\Theta^k)} \tag{3.4}$$

$\overline{\Phi}^*$ being an estimate of the optimal dual cost and γ being a real number in [0, 2]. The subgradient iteration terminates when the resultant step size α is sufficiently small or a fixed number of iterations has been achieved. Definition and calculation of the subgradient $\mathbf{g}(\Theta)$ are given in Appendix A.

D. Construction of a Good Feasible Schedule

Theoretically, even when the optimal solution to the dual problem (D) is obtained, it may still result in an infeasible schedule, i.e., some of the batching constraints (2.2) or capacity constraints (2.3) cannot be satisfied by the dual optimal solution. This is because of the integer decision variables involved. However, the dual cost, i.e., the minimal cost of a relaxed problem from (P), does provide a lower bound to the optimal cost of (P). To complete our solution methodology, an iterative heuristic algorithm is further developed to adjust the dual solution to a near-optimal, feasible schedule by taking advantages of the marginal cost interpretation of Lagrange multipliers and the network structure of the flow balance equations.

Key ideas of the heuristic algorithm are briefly summarized as follows. The algorithm checks all recipes and all machine groups for each time unit in an ascending order over the whole time horizon to see whether their respective batching or capacity constraints are satisfied. When a violation of left-hand part of a batching constraint (2.2) occurs, i.e., b_{rt} is too large for the available lots to batch, b_{rt} is reduced to the largest value needed for satisfying the constraint. A violation of diffusion machine capacity constraints (2.3a) occurs when too many batches are scheduled to the machine. To resolve it, the number of scheduled batches is reduced to the corresponding machine capacity. As there may be a reduction of a few batches, our heuristic reduces batches starting from those of the lowest weights of batched lots.

When there is a constraint violation with the right-hand side of batching constraints (2.2), too many lots are put into a batch. Lots are removed from the batch until the constraint is satisfied. Similarly, a violation of the nondiffusion machine capacity constraints (2.3b) results from the too many lots scheduled. The heuristic removes excess lots starting from those of the lowest weights. Removed excess lots have to be rescheduled for processing while meeting all the constraints.

The removal and rescheduling of excess lots are done through the *PULL* and *PUSH* procedures.

PULL Consider the material flow network of the lot to be removed. Focus on the upstream and downstream subnets of the arc with the excess flow. Pull the excess amount of flows out of both subnets in

a way that results in minimum production cost changes, i.e., by solving a MCLNF problem for each subnet. Then update arc flows of the material flow network according to the removal.

PUSH Consider the material flow network of the lot to be rescheduled and modify the arc capacities of the flow network: the capacity of an arc belonging to the feasibility-checked portion of the schedule is set to the residual capacity of the corresponding machine group; otherwise, set an arc capacity to the difference between the corresponding machine capacity and the arc flow of this network only. The lot to be rescheduled is then routed through the modified network by solving a MCLNF problem of it.

Note that the capacity setting in the *PUSH* procedure ensures that the rescheduling of a lot causes no new violations of constraints for the feasibility-checked portion of the schedule. Although violations may still exist in the portion yet to be checked, they will eventually be resolved as our heuristic algorithm iterates over all the time units in an ascending order. The heuristic thus guarantees the feasibility of the final solution but not the optimality. Further descriptions of this heuristic algorithm for constructing a good feasible schedule (*CGFS*) are given in Appendix B.

Once a feasible schedule is obtained, the corresponding cost of the objective function is an upper bound on the optimal cost, while the dual cost serves as a lower bound. The difference between the optimal cost and the lower bound is known as the *duality gap* which provides a measure of the optimality of the feasible solution; the smaller the gap, the closer the feasible schedule to the optimal.

Remark: To focus on the key ideas, the handling of merging and splitting constraints (2.5)–(2.7) are omitted from the previous descriptions of our solution methodology. Note that (2.5)–(2.7) describe the coupling between lot types before and after a merging or splitting stage. Similar application of the Lagrange relaxation technique described in Section III-A again decouples the production flow of these different lot types and facilitates the same decomposition of the original scheduling problem into independent, single lot type scheduling subproblems. The *CGFS* heuristic is extended to also guarantee the solution feasibility with respect to constraints (2.5)–(2.7) by following exactly the same framework of the *CGFS* in Section III-D. Interested readers may refer to [16] for more details.

IV. Fast Rescheduling

Consider uncertainties in *machine availability and lot holding/release,* which are frequently encountered in an R&D fab and have significant effects on production schedules. When an unexpected event of these uncertainties occurs and makes the original schedule infeasible, rescheduling is needed to maintain schedule feasibility in a timely, economical and smooth way; shuffling the original production around the shop floor due to rescheduling is definitely undesirable. These requirements are very similar to the functions provided by the *CGFS* heuristic algorithm. Ideas and steps of the *CGFS* algorithm that exploits the economical interpretation of Lagrange multipliers and the network structure of production flows therefore constitute the backbone of our fast rescheduling algorithm developments.

In the following algorithm developments, values of Lagrange multipliers associated with the nominal schedule are stored and available as inputs. Let t be the time when an uncertain event occurs. Note that since the production schedule before time t has been realized, rescheduling can only be for the part of schedule over duration $[t, T]$.

Machines may unexpectedly become unavailable due to failure or other adverse events (e.g., out of photoresistance for a photolithography machine). Once it happens, the originally scheduled production may exceed the available machine capacity. The excess production loads are then adjusted as follows:

1) Fast Rescheduling for Capacity Loss: Assume that machine group m lost part of its capacity at time t and this will last until time $t + d$.

Step U.1) Update the capacity of machine group m during $[t, t + d]$.

Step U.2) If m is a diffusion machine group, we first apply *Step 3.1* of *CGFS* to check if machine group m is over-loaded. If so, apply *Step 3.2* of *CGFS* to reduce the number of batches scheduled to machine group m at time t. For lots in these reduced batches, apply *Steps 2.1* and *2.2* of *CGFS* to reschedule them. If m is not a diffusion machine group, apply *Step 4.1* of *CGFS* to check the feasibility of the original schedule under the adjusted capacity and *Step 4.2* to reschedule the excess lots if any.

When the failed machines are repaired and become available before the expected time, rescheduling is also needed to bring these machines back on line.

2) Fast Rescheduling for Capacity Gain: Assume that part of the capacity of machine group m unexpectedly becomes available at time t.

Step A.1) Update the capacity of machine group m.

Step A.2) To utilize the newly available capacity, the original schedule is adjusted by constructing and solving the MCLNF problem of the production flow network of each lot. As such an adjustment may cause violations of constraints, apply *CGFS* to maintain the feasibility of the new schedule.

Sometimes, lots are held by engineers in an R&D fab due to unexpected needs for experimentation, inspection or loop test. The production flows of the held lot can be easily removed from the schedule by applying the *PULL* procedure. However, removal of lots makes some machine capacity available. Ideas similar to those of handling capacity gain are used to reschedule the utilization of such emerging capacity.

3) Fast Rescheduling for Holding of a Lot: Assume that lot i is held at time t.

Step H.1) Apply *PULL* to remove production flows of lot i from the schedule after time t.

Step H.2) Apply *Step A.2* to the remaining lots.

The held lots may be released unexpectedly earlier than scheduled.

4) Fast Rescheduling for Releasing of a Lot: Assume that the held lot i is released at time t.

Step R.1) To insert the released lot i into the original schedule, construct and solve the MCLNF problem of the material flow network of lot i. Apply *CGFS* to maintain the feasibility of the new schedule.

V. Industrial Applications

A. Fab Scheduler—ERSOFS

Our scheduling methodology is implemented into a computer tool, Electronic Research & Service Organization Fab Scheduler (ERSOFS), for daily or short-term scheduling of the Submicron Laboratory of ERSO. Functional blocks of ERSOFS are depicted in Fig. 3. Nominal fab scheduler (NFS) implements our baseline scheduling algorithm. The input interface of ERSOFS draws the scheduling data from the management information system PROMIS[1] and generates four input data files for NFS: *lot.dat* containing lot information such as lot ID, current status and priority; *machine.dat* for information of machine ID, type, forecast capacity and functionality; *process.dat* providing process data and the process-machine map; *weight.dat* containing the weighting coefficients that are automatically generated according to priorities of lots. Weighting coefficients for hot lots are manually keyed into file *hot.dat*. Scheduling results of NFS are presented to users in four files through an output interface: *schedule.lot* and *schedule.eqp* present the schedules generated by NFS by individual lots and individual machines, respectively (see Tables II and III for illustrations); *schedule.wip* and *schedule.utl* outputting the end-of-day WIP distribution and the predicted machine utilization (loading) based on the NFS schedules.

There was no fab scheduler in ERSO before ERSOFS was installed. Experienced shop floor supervisors determined the daily production targets every morning. Technical operators of individual machines were responsible for dispatching lots when a machine became idle. Both the supervisors and operators used empirical rules to schedule and dispatch lots for processing.

B. Industrial Validation

ERSOFS has been tested on data collected from the field from Sept. 15, 1993–Oct. 16, 1993, and is now executed twice a day by the Submicron Laboratory of ERSO: once at 2:00 a.m. and again at 2:00 p.m. During the period of testing, there were about one hundred machines. The number of machines available for production at any one time generally varies due to machine failures and process engineering developments. Except for two types of machines where there are two machines in each type, all the remaining types have only single machine, i.e., the fab is essentially a single line production facility. Machine status and expected capacity are input by fab supervisors with preventive maintenance schedule considered. The weighting coefficients are set to 10 000 for all stages of hot lots. For a normal lot, the weighting coefficient is set to 100 for its first stage to process in the day and then discounted by 60% for the every following stages. Such a setting of weighting coefficients tries to rush all hot lots and to equalize the numbers of moves during a day for normal lots.

[1] PROMIS is a trademark of PROMIS Systems Corporation.

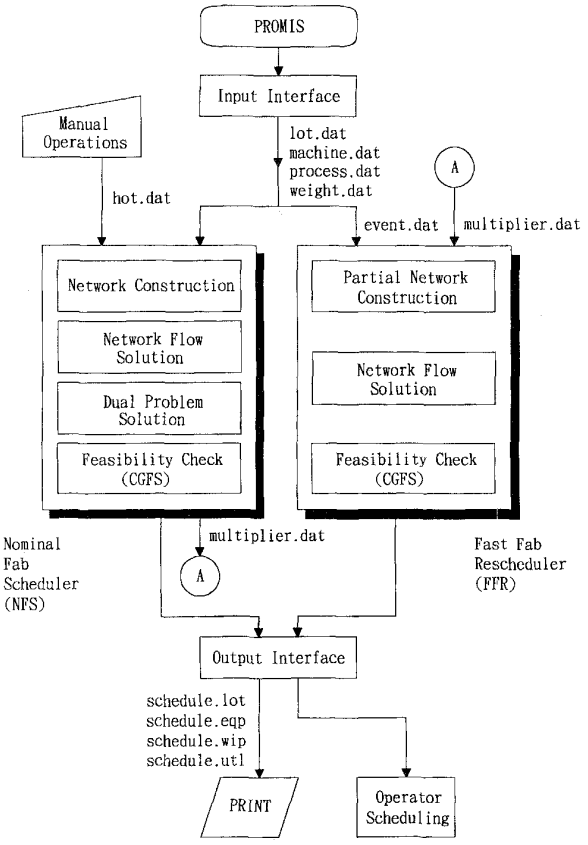


Fig. 3. Functional block diagram of fab scheduler—ERSOFS.

TABLE I
Scheduling Data Set

Date	9/15	9/17	9/18	9/22	9/23	9/24	9/29	10/1	10/2
Total # of Lots	113	113	91	78	118	125	118	66	93
# Hot Lots	10	14	10	15	16	15	15	10	15
Date	10/4	10/5	10/6	10/8	10/12	10/13	10/15	10/16	
Total # of Lots	98	101	93	98	100	96	115	104	
# of Hot Lots	11	12	12	13	13	11	12	6	

Table I lists 17 sets of test data, where each set includes three data items: the date, the total number of lots and the number of hot lots. The scheduling time horizon is set to 24 time units with 1 h as a unit. All of our experiments are performed on a SUN SPARC II workstation.

Numerical results of the 17 data sets are listed in Table IV with those of the actual fab schedules for comparisons. All the resultant relative duality gaps are below 5%, and most of them are below 2%. Our baseline algorithm is therefore considered near-optimal for the decision criterion of problem (P). The scheduled costs of all the cases are much better than the costs

TABLE II
SCHEDULING RESULTS BY INDIVIDUAL LOTS

LOT_ID: L233096.1 pieces: 17 <HOT LOT>
PRODUCT TYPE: TA56112-FULL-2.01

EQUIPMENT	PROCEDURE	RECIPE	TIME	PROC TIME
P5500	D7-PCT-PH-2.01	FALPPC.01	15-OCT 14:42	2
OST	D7-PCT-PH-2.01	FINAD0.01	15-OCT 15:42	1
C8000A	D7-PCT-IM-2.01	FBKHT6.01	15-OCT 15:42	1
GSD	D7-PCT-IM-2.01	FIMPCC.01	15-OCT 15:42	4
L3200	D7-PCT-IM-2.01	FASHRP.01	15-OCT 23:42	1
PRS1	D7-PCT-IM-2.01	FRM521.01	16-OCT 00:42	3
OST	D7-PCT-IM-2.01	FINAR0.01	16-OCT 02:42	1
P5500	D7-NCT-PH-2.01	FALNPC.01	16-OCT 02:42	2
OST	D7-NCT-PH-2.01	FINAD0.01	16-OCT 03:42	1
C8000A	D7-NCT-IM-2.01	FBKHT6.01	16-OCT 03:42	1
GSD	D7-NCT-IM-2.01	FIMNCA.01	16-OCT 03:42	4
L3200	D7-NCT-IM-2.01	FASHRP.01	16-OCT 05:42	1
PRS1	D7-NCT-IM-2.01	FRM521.01	16-OCT 06:42	3

TABLE III
SCHEDULING RESULTS BY INDIVIDUAL MACHINES

EQUIPMENT: FT-III

LOT_ID	PROCEDURE	RECIPE	TIME
L233111.1	S7-VIA-ET-1.01	MYS717.01	15-OCT 12:42
L233156.1	D7-P1-ET-3.01	MYD711.01	15-OCT 12:42
L233057.3	S5-CT-ET-2.01	MYS542.01	15-OCT 12:42
L233183.1	D5-P1-ET-3.01	MYD515.01	15-OCT 15:42
L233183.1	D5-P1-ET-3.01	MYD511.01	15-OCT 15:42
L233075.1	S7-VIA-ET-1.01	MYS717.01	15-OCT 17:42
L236099.1	D7-P1-ET-1.01	ZDM208.01	15-OCT 17:42
L233107.1	S7-VIA-ET-1.01	MYS717.01	15-OCT 19:42
L233151.1	S5-BC-ET-3.01	MYS538.01	15-OCT 21:42
L233142.1	D7-P1-ET-2.02	MYD711.01	16-OCT 06:42
L233150.1	D5-P1-ET-3.01	MYD515.01	16-OCT 06:42
L233150.1	D5-P1-ET-3.01	MYD511.01	16-OCT 06:42
L233153.1	S5-BC-ET-3.01	MYS538.01	16-OCT 09:42
L233168.1	D7-P1-ET-2.02	MYD711.01	16-OCT 09:42
L233175.1	D5-AA-SNET-3.01	MYD501.01	16-OCT 09:42

of the actual schedules. Also the scheduled moves by ERSOFS outperform the actual moves in all cases. Computation times of all the cases tested are less than 8 min on a SUN SPARC II workstation and about 6 min on a VAX6410 computer under the PROMIS environment of ERSO. For the purpose of daily scheduling, it is considered acceptable for real application.

In an R&D fab, it is highly desirable to rush hot lots through in time. After comparing the empirical schedules with those generated by ERSOFS, we find that ERSOFS uniformly schedules more moves for each hot lot than those of the actual schedules. Furthermore, the scheduled total moves of normal lots from ERSOFS are more than those of the actual schedules. In addition, the numbers of moves in a day among normal lots tend to be equalized by ERSOFS. The differences between ERSOFS and the empirical schedules may be caused by several reasons: 1) ERSOFS schedules globally for the fab and achieves good coordination and resource utilization over the whole line; 2) empirical scheduling by human supervisors is limited by utilizing local information and myopic decision-making; 3) ERSOFS rounds off the processing time in the unit of an hour, which may result in differences with the empirical results; 4) there is lack of sufficient historical data on processing time, partly because it is a one-year-old fab and partly because of its R&D nature. The superiority of ERSOFS over empirical schedules due to reasons 1) and 2) has been clearly observed from the schedule for diffusion machines, where the empirical schedules frequently spent unnecessary time waiting for lots to form a batch or wasted machine capacity by processing batches smaller than the normal size. Either case results in low machine utilization and poor productivity.

In Table IV, the scheduled moves of ERSOFS are significantly more than those of actual schedules in several days. These differences were caused by unexpected failures of some key machines and the machine capacity deviated much from what had been forecasted at the time of scheduling by ERSOFS. For example, two photolithography machines and one testing machine, which are bottleneck machines, failed after the scheduling time of Oct. 2. Due to the feature of single line production, production may severely stall once a key machine fails. To handle such uncertainty, a fast rescheduling part is further developed in Section V-D by exploiting the structure of NFS so that ERSOFS becomes a complete daily scheduling tool.

TABLE IV
FIELD SCHEDULING RESULTS

Date	Scheduled Cost	Actual* Cost	Lower Bound	Duality† Gap(%)	CPU Time (Seconds)	S/A‡
9/15	-2,343,307	-1,619,518	-2,357,934	0.620	452.82	1.110
9/17	-2,761,731	-1,647,394	-2,778,362	0.599	471.59	1.158
9/18	-1,777,142	-405,146	-1,830,037	2.890	326.85	2.606
9/22	-2,245,479	-1,587,079	-2,267,647	0.978	308.18	1.034
9/23	-2,086,194	-1,644,105	-2,104,044	0.848	331.26	1.031
9/24	-2,711,213	-1,450,732	-2,832,851	4.294	420.63	1.199
9/29	-2,435,717	-1,418,971	-2,451,207	0.632	445.07	1.552
10/1	-1,340,337	-1,521,462	-1,385,273	3.244	164.36	1.227
10/2	-1,476,381	-217,953	-1,492,120	1.055	297.21	5.558
10/4	-1,881,394	-1,564,583	-1,888,981	0.402	325.88	1.073
10/5	-1,968,445	-1,703,347	-1,989,853	1.076	342.20	1.002
10/6	-2,349,128	-1,426,232	-2,383,055	1.424	329.14	1.274
10/8	-2,139,367	-1,576,986	-2,181,971	1.953	336.58	1.136
10/12	-1,493,512	-1,569,736	-1,517,736	1.596	380.24	1.070
10/13	-1,249,505	-1,517,813	-1,259,986	0.832	314.97	1.105
10/15	-2,197,486	-1,584,749	-2,224,689	1.223	473.17	1.399
10/16	-1,367,319	-389,578	-1,378,511	0.812	448.43	4.761

$*$: obtained by applying the cost measure of (P) to the actual schedule

† : Duality Gap $= \frac{\text{Scheduled Cost} - \text{Lower Bound}}{\text{Lower Bound}} \times 100\%$

‡ : S/A $= \frac{\text{ERSOFS Scheduled Moves}}{\text{Empirically Scheduled Moves}}$

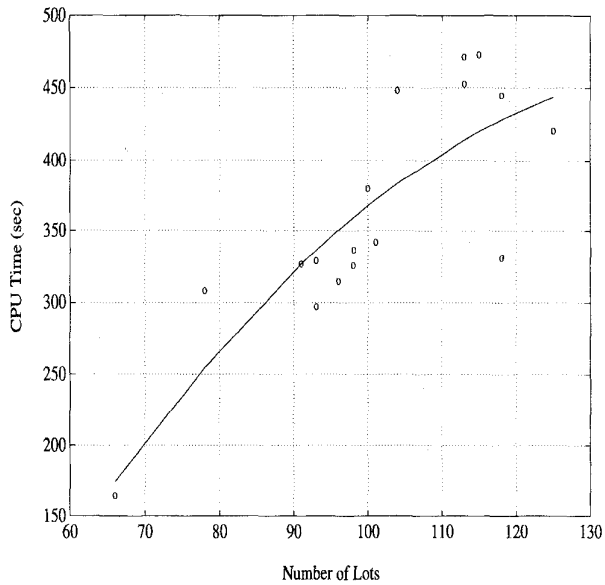


Fig. 4. CPU time of ERSOFS versus number of lots.

C. Algorithmic Features

To assess the potential of extending our methodology to larger fabs, we analyze the computational features of the NFS algorithm. Major computational loads of the NFS algorithm lie in solving the subproblem $(PS-i)$'s and the subgradient iterations. It is known that the computational complexity of the RELAX code for solving a MCLNF problem of $(PS-i)$ is $O(N^3 \log NC)$ [4], where N is proportional to the number of stages of type-i lots and the scheduling time horizon and C is the range of arc cost coefficients. The convergence of subgradient iterations slows down as the number of Lagrange multipliers increases, i.e., slows down with respect to the increase of the number of machine groups, number of batches and the scheduling time horizon.

In the R&D pilot line, each type of wafers is of a small production volume ranging from one to four lots. These different lots of the same type are usually distributed in the wafers-in-process of nearby processing stages at the beginning of a day. Each lot may only go through a few (no more than 5, empirically) consecutive stages of processing during one day because of the processing, setup and waiting times. These facts imply that only a small portion of the processing stages (e.g., 10 out of 100) need to be considered in each subproblem $(PS-i)$. Namely, the network for representing $(PS-i)$ to an hourly resolution has approximately 24×10 nodes and at most 4 units (lots) of flows on it. Each subproblem $(PS-i)$ can therefore be solved very efficiently.

As the number of wafer types increases, the number of subproblems increases, but not the dimension of each subproblem or the Lagrange multipliers. So the computational time probably increases linearly with respect to the number of wafer types. As each type of wafer consists of one to four lots in an R&D fab, we therefore project that the computational time probably increases linearly with respect to the number of lots. Numerical results in Section V-B support this conjecture. Fig. 4 depicts the relationship between computation time versus the number of lots, where a polynomial fit of degree 3 to the data in Section V-B is used. It indicates that computation time is approximately a linear function of the number of lots.

D. Fast Rescheduling Numerical Results

The fast rescheduling algorithms are implemented into the fast fab rescheduler (FFR) module and is integrated with the NFS according to the philosophy in Fig. 1 to make ERSOFS a complete scheduling tool. As shown in Fig. 3, the input interface of ERSOFS triggers the rescheduling FFR through file *event.dat*, which specifies the uncertainty to cope with, its occurrence instance and the duration. The Lagrange multipliers obtained from solving for the nominal schedule by NFS are stored in file *multiplier.dat*, which is input by FFR for fast rescheduling under small disturbances and by the periodic rescheduling of NFS as a starting point of iterations to help in reducing the computational time [8].

To evaluate the effectiveness of these fast rescheduling algorithms, the data set of Oct. 15 (in Section V-B) is taken as the nominal scenario and the following uncertainties are assumed to occur respectively:

Step S1) The machine OST fails at 2:00 p.m. and it is not back to normal until 8:00 p.m.

Step S2) The failed machine OST is repaired and becomes available at 7:00 p.m., i.e., 1 h earlier than expected.

Step S3) Lot L233 077.1 is unexpectedly held at 8:00 a.m. by engineers after its completion of processing at machine S7080.

TABLE V
TESTING RESULTS OF RESCHEDULING FOR UNCERTAINTIES

Scenario		S1	S2	S3	S4
Fast Rescheduling Algorithm	CPU Time (sec)	10.31	7.49	3.15	14.67
	Cost1	-1,041,311	-257,062	-2,196,327	-2,254,703
Re-Solving By The Baseline Algorithm	CPU Time (sec)	223.81	207.67	189.54	227.10
	Cost2	-1,042,638	-259,739	-2,210,752	-2,265,881
	Dual Cost	-1,053,069	-261,823	-2,224,013	-2,286,459
	Duality Gap	0.99%	0.80%	0.60%	0.90%
$\frac{(cost1-cost2)}{cost2} \times 100\%$		0.13%	1.03%	0.63%	0.49%

Step S4) Hot lot L233 099.5 is unexpectedly released to machine PR5000 at 11:00 a.m. to continue its process.

Table V lists the rescheduling results of these four test scenarios. Note that the initial multipliers for direct rescheduling by the baseline scheduling algorithm are set to the values of the dual solution from scheduling the nominal scenario. From these results we can see that the fast rescheduling algorithms are very effective for handling small disturbances: the computation times required are well within the limitation for real-time application (less than 15 s) and the adjusted schedules are very close to those by direct application of the baseline algorithm.

VI. CONCLUSIONS

We have developed a daily scheduling tool, ERSOFS, for R&D semiconductor fabrication. Our problem formulation has captured the salient features such as high variety and very low volume, cyclic process flow, batching at diffusion machines, single mask for each photolithography operation, loop test and engineering splitting and merging of wafer lots. A solution methodology based on Lagrangian relaxation and network flow techniques has been developed, implemented and validated. Field testing results have demonstrated that ERSOFS efficiently generates schedules with high quality. The rescheduling function of ERSOFS has provided fast and smooth adjustments of schedules to cope with the high production uncertainties in an R&D fab. Analysis of the algorithmic properties have also demonstrated the potential of ERSOFS for application to large fabs.

The four output reports of ERSOFS provide a guideline for daily shop floor control in the fab; it helps to identify the bottleneck machines and forecast the WIP distribution. The schedule also serves as an input for the operator scheduling module of ERSO, which considers the availability of technical operators and tries to meet the hourly schedules determined by ERSOFS.

APPENDIX A
SUBGRADIENTS OF DUAL FUNCTION

The subgradients of the dual function $\Phi(\lambda, \pi, \mu, \nu)$ with respect to Lagrange multipliers λ_{mt}, π_{mt}, μ_{rt}, and ν_{rt} are given as

$$\mathbf{g}^1_{mt}(\mathbf{u}^*, \mathbf{b}^*) \equiv \frac{\partial}{\partial \lambda_{mt}} \Phi(\lambda, \pi, \mu, \nu)$$
$$= \sum_{\substack{(i,s) \\ M(i,s)=m}} \sum_{\tau=t-P_{(i,s)}+1}^{t} u_{(i,s)\tau} - C_{mt}, \quad \forall m \notin DIFF, \text{ and } t,$$

$$\mathbf{g}^2_{mt}(\mathbf{u}^*, \mathbf{b}^*) \equiv \frac{\partial}{\partial \pi_{mt}} \Phi(\lambda, \pi, \mu, \nu)$$
$$= \sum_{\substack{r \\ N(r)=m}} \sum_{\tau=t-P_r+1}^{t} b_{r\tau} - C_{mt}, \quad \forall m \in DIFF, \text{ and } t,$$

$$\mathbf{g}^3_{rt}(\mathbf{u}^*, \mathbf{b}^*) \equiv \frac{\partial}{\partial \mu_{rt}} \Phi(\lambda, \pi, \mu, \nu)$$
$$= \underline{B} \cdot b_{rt} - \sum_{\substack{(i,s) \\ R(i,s)=r}} u_{(i,s)t}, \quad \forall r \text{ and } t,$$

$$\mathbf{g}^4_{rt}(\mathbf{u}^*, \mathbf{b}^*) \equiv \frac{\partial}{\partial \nu_{rt}} \Phi(\lambda, \pi, \mu, \nu)$$
$$= \sum_{\substack{(i,s) \\ R(i,s)=r}} u_{(i,s)t} - \overline{B} \cdot b_{rt}, \quad \forall r \text{ and } t.$$

APPENDIX B
HEURISTIC ALGORITHM FOR CONSTRUCTING A GOOD FEASIBLE SCHEDULE (*CGFS*)

Initialize with the schedule obtained from solving the dual problem.

Do for t for all time horizon in an ascending order

Do for r for all recipes

Step 1.1) Check if the left-hand side of the batching constraint of recipe r at time t is violated.

Step 1.2) If so, reduce the batch b_{rt} by $b_{rt} \equiv \lceil \sum_{\substack{(i,s) \\ R_{is}=r}} u_{ist}/\overline{B} \rceil$.

Step 2.1) Check if the right-hand side of the batching constraint of recipe r at time t is violated.

Step 2.2) If so, call procedure *PULL* to remove the lot with the smallest weights and procedure *PUSH* to reschedule it. Repeat this step until the violation is resolved.

Enddo

Do for m for all diffusion machine groups

Step 3.1 Check if the capacity constraint of diffusion machine group m is violated at time t.

Step 3.2 If so, reduce the batch of recipe r whose weighted moves $\sum_{\substack{(i,s) \\ R_{is}=r}} \psi_{is} u_{ist}$ are smallest. Repeat this step until the violation is resolved.

Enddo

Do for m for all nondiffusion machine groups

Step 4.1) Check if the capacity constraint of nondiffusion machine group m is violated at time t.

Step 4.2) If so, call procedure *PULL* to remove the lot with the smallest weights and procedure *PUSH* to reschedule it. Repeat this step until the violation is resolved.

Enddo

Enddo.

ACKNOWLEDGMENT

The authors would like to thank C.-C. Chien, C.-S. Cheng, S.-R. Yen, S.-L. Sun, C.-F. Hung, and the engineers of ERSO, ITRI for their data collection and valuable discussions.

REFERENCES

[1] B. Bona, P. Brandimarte, C. Greco, and G. Menga, "Hybrid hierarchical scheduling and control systems in manufacturing," *IEEE Trans. Robot. Automat.,* vol. 6, no. 6, 1990.

[2] S. X. Bai, N. Srivatsan, and S. B. Gershwin, "Hierarchical real-time scheduling of a semiconductor fabrication facility," in *Proc. 9th IEEE Int. Electron. Manuf. Tech. Symp.,* Washington, DC, Oct. 1990.

[3] D. P. Bertsekas, *Dynamic Programming.* Englewood Cliffs, NJ: Prentice-Hall, 1987.

[4] D. P. Bertsekas and P. Tseng, "Relaxation methods for minimum cost ordinary and generalized network flow problems," *Oper. Res.,* vol. 36, no. 1, pp. 93–114, 1988.

[5] G. R. Bitran and D. Tirupati, "Planning and scheduling for epitaxial wafer production facilities," *Oper. Res.,* vol. 36, no. 1, pp. 34–49, 1988.

[6] ——, "Development and implementation of a scheduling system for a wafer fabrication facility," *Oper. Res.,* vol. 36, no. 3, pp. 377–395, 1988.

[7] C. G. Cassandras and Y.-C. Ho, "Yield learning and production planning for semiconductor manufacturing systems," in *Proc. 4th Int. Conf. Comp. Integrated Manuf. Automation Tech.,* Oct. 1994, pp. 385–390.

[8] S.-C. Chang and D.-Y. Liao, "Scheduling flexible flow shops with no setup effects," *IEEE Trans. Robot. Automat.,* vol. 10, no. 2, pp. 112–122, 1994.

[9] D. Connors, G. Feigin, and D. Yao, "Scheduling semiconductor lines using a fluid network model," in *Proc. 3rd Int. Conf. Comp. Integrated Manuf.,* May 1992, pp. 174–183.

[10] J. E. Dayhoff and R. W. Atherton, "Signature analysis of dispatch schemes in wafer fabrication," *IEEE Trans. Comp., Hybrids, Manufact. Technol.,* vol. CHMT-9, no. 4, pp. 518–525, 1986.

[11] M. L. Fisher, "Lagrangian relaxation method for solving integer programming problems," *Manage. Sci.,* vol. 27, pp. 1–18, 1981.

[12] C. R. Glassey and M. G. C. Resende, "Closed-loop job release control for VLSI circuit manufacturing," *IEEE Trans. Semiconduct. Manufact.,* vol. 1, no. 1, pp. 36–44, 1988.

[13] J. J. Golovin, "A total framework for semiconductor production planning and scheduling," *Solid State Technol.,* pp. 167–170, May 1986.

[14] D. J. Hoitomt, P. B. Luh, E. Max, and K. R. Pattipati, "Scheduling jobs with simple precedence constraints on parallel machines," *Cont. Sys. Mag.,* vol. 10, no. 2, pp. 34–40, 1990.

[15] M. Held, P. Wolfe, and H. Crowder, "Validation and subgradient optimization," *Math. Program.,* vol. 6, pp. 62–88, 1974.

[16] D.-Y. Liao, *ERSOFS Reference Manual.* Taipei, Taiwan: National Taiwan University, June 1994.

[17] C. Lozinski and C. R. Glassey, "Bottleneck starvation indicators for shop floor control," *IEEE Trans. Semiconduct. Manufact.,* vol. 1, no. 4, pp. 147–153, 1988.

[18] R. C. Leachman, M. Solorzano, and C. R. Glassey, "A queue management policy for the release of factory work orders," *J. Manuf. Oper. Manage.,* 1988.

[19] D. G. Luenberger, *Linear and Nonlinear Programming.* Reading, MA: Addison-Wesley, 1984.

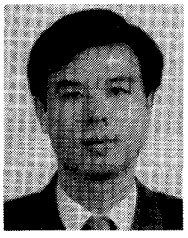
[20] S. H. Lu and P. R. Kumar, "Distributed scheduling based on due dates and buffer priorities," *IEEE Trans. Automat. Contr.,* vol. 36, no. 12, pp. 1406–1416, 1991.

[21] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity.* Englewood Cliffs, NJ: Prentice-Hall, 1982.

[22] M. M. Syslo, N. Deo, and J. S. Kowalik, *Discrete Optimization Algorithms with PASCAL Programs.* Englewood Cliffs, NJ: Prentice Hall, 1983.

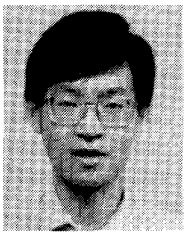
[23] R. Uzsoy, C.-Y. Lee, and L. A. Martin-Vega, "A review of production planning and scheduling models in the semiconductor industry—part I: System characteristics, performance evaluation and production planning," *IEE Trans.,* vol. 24, no. 4, pp. 47–60, Sept. 1992.

[24] L. M. Wein, "Scheduling semiconductor wafer fabrication," *IEEE Trans. Semiconduct. Manufact.,* vol. 1, no. 3, pp. 115–130, 1988.



Da-Yin Liao (S'90–M'91) received the B.S. degree in mechanical engineering and the M.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C., in 1989, 1991, and 1994, respectively.

He is currently serving as a Second Lieutenant in the Chinese Army, Taiwan, R.O.C. His research interests include computer-integrated manufacturing, production scheduling, and production management for semiconductor manufacturing.



Shi-Chung Chang (S'83–M'87) received the B.S.E.E. degree from National Taiwan University, Taipei, Taiwan, R.O.C., in 1979, and the M.S. and Ph.D. degrees in electrical and systems engineering from the University of Connecticut, Storrs, in 1983 and 1986, respectively.

From 1979 to 1981, he served as an Ensign in the Chinese Navy. He worked as a technical intern at the Pacific Gas and Electric Co., San Francisco, CA, in the summer of 1985. During 1987, he was a member of the Technical Staff, Decision Systems Section, Alphatech, Inc., Burlington, MA. He is currently with the Electrical Engineering Department, National Taiwan University. His research interests include optimization theory and algorithms, operation scheduling and control of large-scale systems, parallel computing, high-speed networks, and distributed decision making. He has been a principal investigator and consultant to many industry and government funded projects in the above areas, and has published more than 70 papers.

Dr. Chang is a member of Eta Kappa Nu and Phi Kappa Phi.

Kuo-Wei Pei was born in Taiwan, R.O.C., in 1959. He received the B.S. degree in physics and the M.S. degree in applied physics from the National Cheng Kung University, Tainan, and Chung Cheng Institute of Technology, Chung-Li, respectively.

In 1991, he joined the Electronics Research and Service Organization, Industrial Technology Research Institute, Taiwan, R.O.C., as an engineer in the VLSI process engineering and manufacturing. Currently, he is a Section Manager, Manufacturing Department, Vanguard International Semiconductor Corporation, Hsinchu, Taiwan, R.O.C.

Chi-Ming Chang (M'92) received the M.S. and Ph.D. degrees in industrial engineering from Texas Technical University, Lubbock, in 1982 and 1986, respectively.

From 1986 to 1990, he was an Assistant Professor, Louisiana Technical University, Ruston, where he taught and published in the areas of manufacturing systems engineering, manufacturing control, and simulation. In 1990, he was attracted by the Submicron Process Technology Project and joined Electronics Research and Service Organization, ITRI, Taiwan, R.O.C. As Deputy Director, Semiconductor Manufacturing Engineering Division, he was responsible for the operation of the 4-in wafer fabrication R&D pilot line and later for the start-up of the 8-in wafer fabrication R&D pilot line. Upon completion of the Submicron Project in Dec. 1994, he joined Vanguard International Semiconductor Corp., Hsinchu, Taiwan, R.O.C., the off-shoot company of the Submicron Project, as Director of Product Manufacturing Engineering Division.