A TAXONOMY OF SCHEDULING PROBLEMS IN SEMICONDUCTOR DEVICE TEST OPERATIONS

TALI FREED

ROBERT C. LEACHMAN

KENNETH H. DOERR

ABSTRACT
Semiconductor device test facilities differ not only by production volume and tester brands. The complexity of the devices and the characteristics of the testers affect the scheduling methodologies as well. Goals and strategies vary from one firm to another, leading to a variety of objectives and performance measures. Due to random yield lot size is variable and lot priorities are common. Changeover times are oftentimes sequence-dependent. Since semiconductor device testing systems are very costly, scheduling methods that increase the throughput of the facility are financially significant. In this paper we describe a variety of semiconductor device testing environments, develop mathematical formulations for their scheduling problems, and suggest solution methods. The paper is intended to serve as a basis for the development of scheduling systems for a variety of semiconductor device testing facilities.

1. INTRODUCTION
The semiconductor manufacturing process consists of the wafer fabrication sub-process (front-end) and the device packaging and testing sub-process (back-end). In the front-end silicon wafers are chemically processed to generate electronic devices. In the back-end the wafers are sawn (sliced) into devices, and the devices are packaged, branded, and tested. A detailed description of the semiconductor manufacturing process can be found in [1], [2], and [3]. The device test portion of the process is the focus of this paper.

Semiconductor device test facilities differ not only by production volume and tester brands. The complexity of the devices and the characteristics of the testers vary, and affect the scheduling methodologies. Firm goals and strategies dictate its objectives and performance measures. Multiple test operations, sequence-dependent changeover times, random lot size and lot priorities are common. Furthermore, test operations sometimes have the unique characteristic that the lot processing rate depends on other lots assigned to the same tester.

Due to the high cost of test systems scheduling methods that increase the throughput of the facility are financially significant. In this paper we describe a variety of semiconductor device testing environments, develop mathematical formulations for their scheduling problems, and suggest solution methods.

2. LITERATURE REVIEW
Previous research on various aspects of planning and scheduling the complete semiconductor manufacturing process can be classified into three major categories: 1) performance evaluation methods, 2) production planning models, and 3) shop-floor control techniques. The reader is referred to the review papers [2], [3] for a more detailed discussion on these methodologies.

The area of shop-floor control of semiconductor manufacturing operations can also be classified into three major categories: 1) dispatching rules and input regulation strategies, 2) optimal control and knowledge-based systems, and 3) deterministic scheduling algorithms. The reader is referred to [3] for a detailed review of models that belong to the first two categories, as well as for a review of deterministic scheduling methods for burn-in ovens.

Most of the deterministic scheduling algorithms have been designed for wafer fab applications and are not applicable for the fundamentally different test operations (e.g., [4], [5], [6], [7], [8]). Scheduling test operations has been the subject of a series of papers ([9], [10], [11], [12], [13], [14], [15], [16], [17]), that focus on finding good heuristics for solving a dynamic, real-time scheduling problem. The test area in most of these papers is modelled as a job shop, with precedence constraints and deterministic lead times. Some of the papers also consider the sequence-dependency of setup times. [15], [16], and [17] focus on equipment and hardware requirements. [15] and [16] use integer programming with Lagrangian relaxation to solve the scheduling problem, and [17]...
uses Petri nets. There are many test facilities, however, that do not fall into these categories. In this paper, we attempt to describe and model a wide range of inherently different semiconductor device test environments.

3. COMPLEXITIES OF THE SEMICONDUCTOR DEVICE TESTING PROCESS

The production unit in the back-end is a lot of devices. Due to random yield earlier in the process, test lot sizes can range from 1 to 10,000 devices per lot. A lot may be tested once or several times, depending on its complexity and future use. Most commercial applications require a single test operation.

Even when the devices require several test operations, it is sometimes reasonable to assume that a single operation is required. For example, when the work procedures require that all of the lot test operations should be performed sequentially on the same tester, the lot can be viewed as requiring a single processing operation, whose duration is the sum of the test times and changeover times. This approach is particularly plausible when changeover times between test operations are not very significant, and lot priorities and orderly production are of major importance. In this paper we examine only testing environments where single test operations are performed. Models and solution techniques for scheduling problems with multiple test operations are discussed in [1], [9]-[18].

A test operation can typically be performed by several testers. In this paper we model single-tester as well as multiple-tester environments. In the case of multiple testers, we assume that the testers are identical, with identical hardware and software configuration, identical set of device types they can test, and identical processing rates.

Certain types of testers may have up to four testing heads (test-stations). Multiple-head testers, known in the industry as “multiplexing testers”, can be configured to test different device types simultaneously. Lot lead times and tester throughput are then significantly affected by the combination of lots tested concurrently. While multiple-head testers are preferred for testing simpler, mass-produced devices, semiconductor companies that manufacture complex devices with long test times often use single-head testers. The use of single-head testers affords decreased tester scheduling complexity and simplified work procedures, although tester idle time is larger. In this paper we develop models for single-head testers only. The problem of scheduling multiple-head testers is discussed in detail in [1] and [18].

In order to increase throughput, some testers are capable of testing several devices in parallel on the same head. For example, if the test time per device when tested by itself is 3 seconds, two devices in parallel may take 4 seconds to test, and three would take 4.5 seconds. It is important to distinguish between multi-head testing (multiplexing) and parallel-testing of several devices on the same head. In parallel-testing on a single-head tester the devices must be identical, the parallel-tested devices must be loaded and unloaded together, and the lot processing time is known with certainty. In multi-head testing each head can test a different device type, each head is independent of the other heads in terms of loading and unloading its devices, and the device processing time (flow time through the tester) depends on the devices processed on the other heads. From a scheduling perspective, parallel-testing of a lot of devices can be viewed as testing a smaller lot of devices, with longer test time per device. We therefore assume in this paper that test time and lot size data are pre-adjusted to the parallel-testing case, if applicable.

Lot changeover times in semiconductor testing may be significant (several hours) and sequence-dependent. We assume that the changeover time matrix is symmetric (the changeover time from lot A to lot B is equal to the changeover time from lot B to lot A) and that it satisfies the triangle inequality (the sum of changeover times from A to B and from B to C is greater than or equal to the changeover time from A to C). A comprehensive analysis of the changeover operation is presented in [1].

Each lot is assumed to have a unique associated value which reflects factors such as due date, tardiness, urgency of processing, expected revenue, and resource consumption. Value determination methods are beyond the scope of this work; we refer the interested reader to [19], [20], [21] for approaches that represent the above considerations. In this paper we assume that the scheduling objective is oftentimes to maximize the total value of the facility throughput.

We assume here that throughout the time horizon to be scheduled (shift, day, week, or month) lot values are fixed. In [1] and [18] we allow lot values to increase from one shift to the next, representing the increasing urgency in processing the lot. We also assume in this paper that a lot is not considered processed and its value is not realized until all of its units are completed (due to the impracticality of partial shipping).

However, if all lots have identical values, or if the most important objective of the test facility is to complete processing arriving lots as soon as possible and WIP (work in progress) accumulation is to be
avoided (due to reasons of cycle time, inventory costs and production smoothing), a semiconductor test facility may choose to minimize the total time to complete the available WIP (minimum makespan).

Based on the above analysis, the models for the single test operation, single-head tester(s) scheduling problems presented here are classified according to the following characteristics:

1. Objective function - maximum cumulative value vs. minimum makespan;
2. Number of testers - single tester vs. multiple identical testers;
3. Sequence-dependency of the changeover times.

In many of the cases the scheduling problems can be modeled as well-known NP-complete combinatorial optimization problems. The advantage of recognizing that a problem belongs to this class stems from the fact that the popularity of research on these well-known problems has sometimes led to the availability of a variety of solution techniques.

In order to simplify the presentation, we omit the consideration of initial conditions and tester maintenance from the formulations throughout this paper. We assume that all lots are available for processing at the beginning of the time period. The reader is referred to [1] and [18] for a solution methodology for the case where initial conditions and tester maintenance are part of the formulation.

In Section 4.1 we present single and multiple tester models assuming sequence-independent changeover time, while Section 4.2 focuses on formulations for sequence-dependent changeover time.

Within each section, models are classified by objective function. For the sequence-independent, multiple tester case and for the sequence-dependent, single tester case two objective functions are considered: 1) maximum cumulative value, and 2) minimum makespan. For the sequence-independent, single tester case the objective of minimum makespan is meaningless, and therefore not considered. To the best of the authors' knowledge, the only solution methodology suggested for the sequence-dependent, multiple tester case with a maximum value objective function can be found in [1] and [18], where the problem is generalized to multiple test operations and multiple-head testers.

4. SCHEDULING MODELS FOR SINGLE-OPERATION LOTS ON SINGLE-HEAD TESTERS
4.1 Scheduling Strategies for Sequence-Independent Changeover Times
4.1.1 The Sequence-Independent Changeover Time, Value Maximization, Single Tester Scheduling Problem

**Formulation**

Given a set $I$ of lots, each lot $i \in I$ consisting of $N_i$ identical units, for each lot $i \in I$ a changeover time $C_i$ that should be performed before the processing of lot $i$ can begin, a value $V_i$ per unit, a test time $t_i$ per unit, and a handling time $h_i$ per unit.

Find a set of lots that maximizes $V$, the total cumulative value of units processed during the given time horizon $T$, i.e., find a set of binary variables $x_i$ that satisfies the following:

$$\text{max } V = \sum_{i \in I} x_i N_i V_i$$

such that

$$\sum_{i \in I} [C_i + N_i (t_i + h_i)] = \sum_{i \in I} x_i L_i \leq T$$

where $V_i = N_i \cdot V_i$ is the lot value, $L_i = C_i + N_i (t_i + h_i)$ is the lot lead time, and $x_i = 1$ if lot $i$ is selected for processing, and 0 otherwise (the decision variables).

Since the lot values remain constant throughout the shift, all sequences of the selected set of lots will result in identical value.

**Analysis**

This problem is equivalent to the Knapsack problem, which is NP-complete ([22], [23]). The size of each item in the knapsack problem corresponds to the lot lead time $L_i$, and the value of each item corresponds to the lot value $V_i$. The knapsack problem can be solved in pseudo-polynomial time by dynamic programming ([24]). Examples of solution techniques can be found in [25] and in [26].

4.1.2 The Sequence-Independent Changeover Time, Value Maximization, Multiple Tester Scheduling Problem

**Formulation**

Given a set $I$ of lots, each lot $i \in I$ consisting of $N_i$ identical units, for each lot $i \in I$ a changeover time $C_i$ that should be performed before the processing of lot $i$ can begin, a lot-value $V_i$, a test time $t_i$ per unit, and a handling time $h_i$ per unit. Given an integer number $M > 1$ of test systems.

Find an assignment of lots to testers that maximizes $V$, the total cumulative value of units processed during the given time horizon $T$, i.e., find a set of binary variables $x_{im}$ that satisfies the following:
\[ \max V = \sum_{m=1}^{M} \sum_{i \in I} x_{im} V_i \]

such that

\[ \sum_{i \in I} x_{im} [C_i + N_i (t_i + h_i)] = \sum_{i \in I} x_{im} L_i \leq T \]

for \( m = 1, \ldots, M \),

\[ \sum_{i \in I} x_{im} \leq 1 \text{ for all } i \in I \]

and \( x_{im} = 1 \) if lot \( i \) is selected for processing on tester \( m \), and 0 otherwise.

As in Section 4.1.1, since the lot values remain constant throughout the shift, any sequence of the assigned set of lots on the corresponding tester will result in the same value.

**Analysis**

The sequence-independent, multiple tester scheduling problem can be viewed as an extension of the Bin Packing problem, which is also NP-complete ([24]). Each of the \( M \) testers is viewed as a bin. The size of each item, \( L_i \), and the value of each item, \( V_i \), are as defined in Section 5.2.1. However, in the original Bin Packing problem no value is attached to the items and the objective is to maximize the number of items allocated to the bins. The addition of item values makes the problem harder to solve. If all lot values are identical (e.g., \( V_i = 1 \) for all \( i \)) the problem can be reduced to the original Bin Packing problem since in this case maximizing the total value of processed lots is equivalent to maximizing the number of processed lots. The original bin packing problem can be solved in pseudo-polynomial time for any fixed \( M \). Since \( T \) is fixed in our case, the bin packing problem is solvable in polynomial time by exhaustive search.

### 4.1.3 The Sequence-Independent Changeover Time, Makespan Minimization, Multiple Tester Scheduling Problem

**Formulation**

Given a set \( I \) of lots, each lot \( i \in I \) consisting of \( N_i \) identical units, for each lot \( i \in I \) a test time \( t_i \) per unit, and a handling time \( h_i \) per unit. Given an integer number \( M > 1 \) of test systems.

Find an assignment of lots to testers that minimizes the makespan \( Z \), the longest processing time among the testers, i.e., find a set of binary variables \( x_{im} \) that satisfies the following:

\[ \min Z = \max \{ \sum_{m=1}^{M} \sum_{i \in I} x_{im} L_i \} \]

such that

\[ \sum_{i \in I} x_{im} = 1 \text{ for all } i \in I \]

\[ L_i = C_i + N_i (t_i + h_i) \]

is the lot lead time, \( Z \) is the processing makespan, and \( x_{im} = 1 \) if lot \( i \) is selected for processing on tester \( m \), and 0 otherwise.

As in Section 4.1.1 and 4.1.2, the processing duration of each tester is not sensitive to the sequence of the lots assigned to the tester.

**Analysis**

The sequence-independent, makespan minimization, multiple tester scheduling problem can be modeled as a Multiprocessor Scheduling problem, which is also NP-complete ([23]), where each tester is viewed as a processor. The multiprocessor scheduling problem can be solved in pseudo-polynomial time for any fixed \( M \).

### 4.2 Scheduling Strategies for Sequence-Dependent Changeover Times

#### 4.2.1 The Sequence-Dependent Changeover Time, Makespan Minimization, Single Tester Scheduling Problem

**Formulation**

Set \( I \) of lots, each lot \( i \in I \) consisting of \( N_i \) identical units, for each lot \( i \in I \) a test time \( t_i \) per unit, and a handling time \( h_i \) per unit. For each pair of lots \( i \) and \( j \) a non-negative changeover time \( C(i,j) \) that takes effect after the completion of processing of lot \( i \) and before the processing of lot \( j \) can begin.

Find a tester schedule that minimizes \( Z \), the processing makespan, i.e., find a sequence of binary variables \( x_{i,s} \), \( i \in I \), \( s = 1, 2, \ldots, S \), \( S = |I| \), that satisfies the following:

\[ \min Z = \sum_{s=1}^{S} \sum_{i \in I} x_{i,s} L_i + \sum_{s=1}^{S-1} \sum_{i \in I} \sum_{j \in I} x_{i,s} x_{j,s+1} C(i,j) \]

such that

\[ \sum_{s=1}^{S} x_{i,s} = 1 \text{ for all } i \in I \]

\[ \sum_{i \in I} x_{i,s} = 1 \text{ for } 1 \leq s \leq S \]

where \( x_{i,s} = 1 \) if lot \( i \) is the \( s \)th lot in the processing sequence, and 0 otherwise.

**Analysis**

The objective function consists of two components: the processing time component and the
changeover time component. Since the processing times of the lots are constant and independent of their location in the sequence, the processing time component can be eliminated from the objective function, resulting an equivalent formulation, as follows:

\[
\min Z = \sum_{s=1}^{S-1} \sum_{i \in I} \sum_{j \in I} x_{i,s} x_{j,s+1} C(i-j)
\]

such that

\[
\sum_{s=1}^{S} x_{i,s} = 1 \text{ for all } i \in I
\]

\[
\sum_{i \in I} x_{i,s} = 1 \text{ for } 1 \leq s \leq S
\]

\[
x_{i,s} \in \{0,1\}
\]

Thus, the problem is equivalent to the minimization of the total changeover time. The sequence-dependent, makespan minimizing, single tester scheduling problem resembles the well known traveling salesman problem (TSP), which is NP-complete ([23]). A city visited in the traveling salesman tour corresponds to a lot processed by the tester. The order of cities in the traveling salesman's tour corresponds to the processing sequence, and the distance between cities corresponds to the changeover time between consecutively processed lots.

In the original TSP the salesman has a "home city", which is the first and last on his tour. In the tester scheduling problem the processing schedule corresponds to a path (as opposed to a tour) of the traveling salesman, i.e., any of the lots can be selected to be first in the schedule, and this lot will never be returned to (processed again). The modification of the TSP from a tour-TSP to a path-TSP is simple and does not increase the complexity of the problem. It requires the addition of a single fictitious lot to the problem instance (this lot serves as the home city), with zero changeover time between this lot and all the other lots.

Solution techniques for the TSP include techniques which find optimal solutions and heuristic techniques ([27]). Optimal solution techniques can be further divided into (1) techniques which combine cutting planes and branch and bound methods (e.g., [28]), and (2) dynamic programming techniques (e.g., [29]). Heuristic techniques include (1) construction algorithms (e.g., nearest neighbor rule), and (2) improvement algorithms (e.g., edge exchange procedures [30], [31]).

### 4.2.2 The Sequence-dependent Changeover Time, Value Maximization, Single Tester Scheduling Problem

#### Formulation

Set \( I \) of lots, each lot \( l \in I \) consisting of \( N_l \) identical units, for each lot \( l \in I \) a test time \( t_l \) per unit, and a handling time \( h_l \) per unit. For each pair of lots \( i \) and \( j \) a non-negative changeover time \( C(i-j) \) that takes effect after the completion of processing of lot \( i \) and before the processing of lot \( j \) can begin.

Find a set of lots that maximizes \( V \), the total cumulative value of units processed during the given time horizon \( T \), i.e., find a sequence of binary variables \( x_{i,s}, i \in I, s = 1, 2, \ldots, S, S = |I| \), where \( x_{i,s} = 1 \) if lot \( i \) is the \( s \)th lot in the processing sequence and 0 otherwise, that satisfies the following:

\[
\max V = \sum_{i \in I} x_{i,s} N_i v_i = \sum_{i \in I} x_{i,s} V_i
\]

such that

\[
\sum_{s=1}^{S} x_{i,s} = 1 \text{ for all } i \in I
\]

\[
\sum_{i \in I} x_{i,s} = 1 \text{ for } 1 \leq s \leq S
\]

\[
x_{i,s} \in \{0,1\}
\]

**Analysis**

Among the class of Traveling Salesman Subset-tour Problems (TSSP) which include problems such as The Prize Collecting TSP, The Time Constrained TSP and The Orienteering Problem, the sequence-dependent changeover time, value maximization, single tester shift scheduling problem is most similar to the Orienteering Problem. The Orienteering Problem can be described as follows: on a set of nodes, each with an associated profit, and a set of arcs, each with an associated length, find the path beginning at a specified origin and terminating at a specified destination that maximizes the total profit from the nodes on the path subject to: 1) a constraint on the length of the path, and 2) the condition that no node is visited more than once. [32] shows that the orienteering problem is NP-hard.

The assumptions of the orienteering problem include: 1) symmetry of the distance between each pair of nodes, and 2) arc lengths satisfy the triangle inequality. A few modifications should be made in order to formulate the sequence-dependent changeover time, value maximization, single tester shift scheduling problem as an orienteering problem. The first modification involves adding a dummy node as a destination node (following the same principles as adding the fictitious lot of Section 4.2.1). Note that...
typically initial conditions would correspond to the origin node, thus circumventing the need for a fictitious origin node. The distance between nodes (lots) in our problem should be the sum of the changeover time between the pair of lots \((i,j)\) and the processing time of lot \(j\), so that the constraint on the length of the path will enforce the makespan to be shorter than the duration of the time horizon.

Solution techniques for the orienteering problem include Branch and Bound algorithms ([33]), and several heuristic methods (e.g., [32], [34], [35], [36], [37]).

4.2.3 The Sequence-dependent Changeover Time, Makespan Minimization, Multiple Tester Scheduling Problem

The sequence-dependent changeover time, makespan minimization, multiple tester shift scheduling problem can be viewed as determining a collection of optimal paths of traveling salesmen, such that the longest path is minimized. This problem can be modeled as the \(k\) traveling salesmen problem (k-TSP), in which \(k\) salesmen should divide the set of cities to be visited among them, such that each city is visited by a single salesman and the longest salesman's path (schedule makespan) is minimized (this objective function was suggested in [38]).

For the k-TSP, the distance between cities (lots) must include the processing time of the lots, since otherwise the workload allocation among the testers is likely to be very unbalanced. There are several ways to define the distance between the cities. The straightforward way is to define the distance from lot \(i\) to lot \(j\) as the sum of the changeover time from lot \(i\) to lot \(j\) and the processing time of lot \(j\), i.e.,

\[
D(i-j) = C(i-j) + N_j(t_j + h_j)
\]

However, if the changeover time matrix is symmetric then it may be advantageous to define the distance between lot \(i\) and lot \(j\) as the sum of twice the changeover time between lots \(i\) and \(j\) and the processing time of both lots \(i\) and \(j\), i.e.,

\[
D(i-j) = 2N_i(t_i + h_i) + 2N_j(t_j + h_j) + 2C(i-j)
\]

The reason for defining the distance this way is that although each changeover and processing time is counted twice, the resulting distance matrix would be symmetric. This characteristic may be of importance for certain solution techniques (e.g., the optimal solution technique used in [28] to solve the 318-city TSP requires that the TSP distance matrix should be symmetric). The k-TSP can, therefore, be formulated as follows:

\[
\begin{align*}
\min Z &= \max_k \left\{ \sum_{s=1}^{S_k} x_{i,s,k} x_{j,s+1,k} D(i-j) \right\} \\
&\quad \text{such that} \\
&\quad K = S_k \\
&\quad \sum_{s=1}^{S_k} x_{i,s,k} = 1 \text{ for all } i \in I \\
&\quad \sum_{s=1}^{S_k} x_{i,s,k} \leq 1 \text{ for all } i \leq k \leq K \\
&\quad x_{i,s,k} = 1 \text{ if lot } i \text{ is the } s^{th} \text{ lot processed by tester } k, s = 1,...,S_k, \text{ and 0 otherwise.}
\end{align*}
\]

There are several algorithms for the k-TSP that can find near-optimal solutions even for a large number of cities. The best algorithm known starts with a standard traveling salesman tour generated by one of the algorithms for the TSP, and then partitions the tour into \(k\) mutually exclusive tours ([39]). The worst-case performance of this heuristic procedure is superior to other heuristics, such as the greedy incremental approach that grows all \(k\) tours in parallel using nearest insertion techniques ([38]). When symmetry and triangle inequality conditions are satisfied solution techniques with good performance guarantees are available ([27]).

5. CONCLUSIONS

A taxonomy of semiconductor device test scheduling problems is provided in this paper. It can be used to identify directions for research, as well as to assist production planners in understanding the scheduling problems they are facing. The various complexities render every variant of the test(s) scheduling problem NP-complete. The paper demonstrates, however, that in many cases the scheduling problems can be modeled as well-known NP-complete combinatorial optimization problems, and the popularity of research on these problems has led to availability of solution techniques.

6. REFERENCES


part II: shop floor control. *IIE Transactions on Scheduling and Logistics*, 26, 44-55.


