Proceedings of the 2014 Winter Simulation Conference
A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, eds.

FLEXIBLE JOB-SHOP SCHEDULING WITH EXTENDED ROUTE FLEXIBILITY FOR SEMICONDUCTOR MANUFACTURING

Sebastian Knopp
Stéphane Dauzère-Pérès
Claude Yugma

Department of Manufacturing Sciences and Logistics
Ecole Nationale des Mines de Saint-Etienne - CMP
F-13541 Gardanne, FRANCE

ABSTRACT

Scheduling decisions have an important impact on the overall performance of a semiconductor manufacturing facility (fab). To account for machines that consist of several interdependent components, we generalize the flexible job-shop scheduling problem. We introduce the concept of route graphs to describe resource dependencies. Beside specifying feasible routes, route graphs can, for example, prescribe two different operations in the route of a job to use the very same resource. To solve the problem, we introduce an adapted disjunctive graph representation and propose a heuristic method that iteratively inserts jobs to construct an initial solution. This solution is then improved using a simulated annealing meta-heuristic. Several numerical experiments are performed. First, improved results for a real-world instance justify the increased complexity of our model. Second, a comparison to results of dedicated methods for the flexible job-shop scheduling problem shows that our approach obtains good results.

1 INTRODUCTION

Scheduling of wafer processing operations in the diffusion area of a semiconductor manufacturing facility (fab) has an important impact on the overall performance of the fab. Consequently, we want to optimize those decisions while taking real-world constraints into account. For each job in a given set, a linear route of operations must be performed. The operations in the diffusion area are cleaning or furnace operations that need specific machines. To obtain practicable schedules, we consider furnaces in detail and include their components as resources in our model. This leads to a generalized version of the flexible job-shop problem that imposes additional constraints for resource utilization. Currently, our approach does not consider the batching capabilities of furnaces. This would be an important extension to make the method fully applicable in practice.

Furnaces in this area consist of tubes, boats and a load port. A *tube* is the place where processes are conducted. A *boat* is a movable carrier for wafers and necessary to run a process inside a tube. Boats are also utilized to load, unload and cool wafers. The *load port* is the place where wafers are loaded and unloaded. Commonly, such machines consist of two tubes, four boats (two per tube), and one load port. To process a set of wafers, a boat is used as follows: First, wafers are loaded from its carrier to the boat at the load port. Then, the boat is moved into the tube where the process is conducted. Afterwards, the boat is removed from the tube and has to cool down before its wafers can be unloaded at the load port. Potentially, the boat has to wait in case the tube or the load port is occupied. Some operations, e.g. loading wafers, require more than one resource at the same time. The load port, tubes and boats are components of a furnace. Figure 1 provides a schematic illustration of such a machine. We decompose one processing step of a furnace into separate operations using its internal resources as described before. Note that processing times of operations may depend on the specific internal resource that is used.

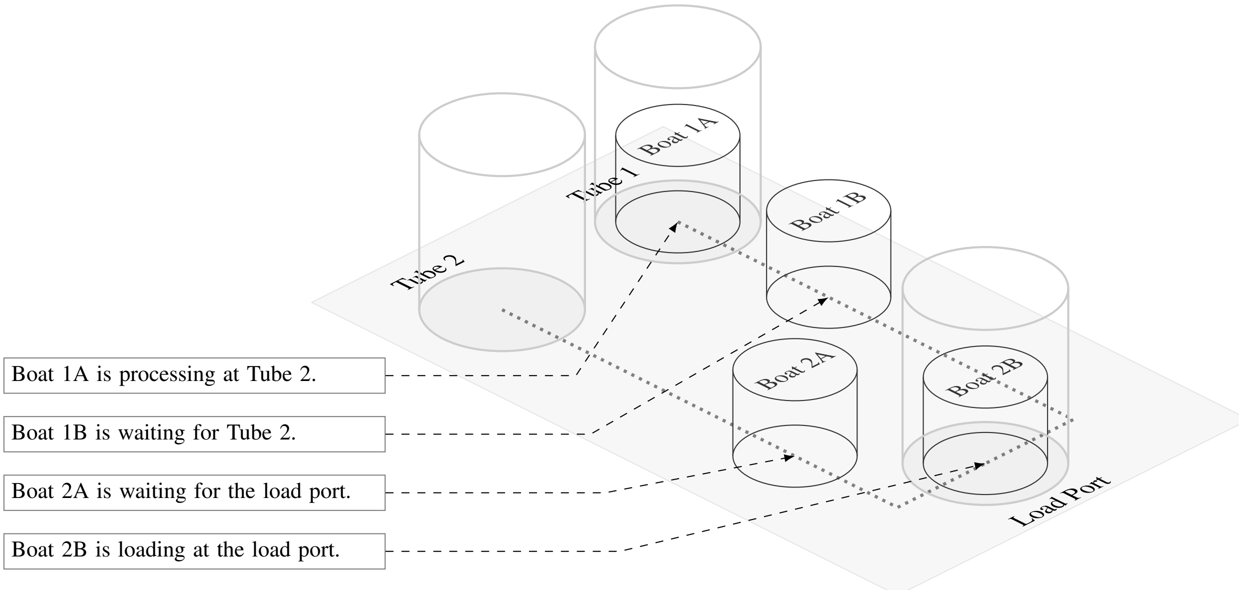

Figure 1: A schematic representation of a furnace in the diffusion area.

The properties described in the previous paragraph impose dependencies between resources used by consecutive operations. Consider the following two characteristic examples. Firstly, if a loading operation uses the load port of a machine, the corresponding unloading operation must use the load port of the very same machine. Secondly, after a specific boat is loaded, it cannot be used elsewhere before being unloaded. So, a resource can be blocked even if no operation is currently using it.

We base our approach on the classical job-shop scheduling problem. There, a given set of jobs must be scheduled using given resources. For each job, an individual route of operations must be processed. Exactly one resource is statically assigned to each operation. This is extended by the flexible job-shop scheduling problem that allows to choose the resource used to process an operation from its set of allocated resources.

To cope with the given constraints of furnaces, we formulate an extension of the flexible job-shop scheduling problem. The formal description is given in section 3.2. In this extension, resource dependencies are taken into account by defining multiple routes with fixed resource assignments: For each job, we statically assign resources to operations and resource flexibility is obtained by allowing different routes. Allowed routes are specified by *route graphs* which are introduced before the actual problem definition.

The introduced model is not dependent on the objective to be optimized. In the context of this study we concentrate on minimizing the makespan. However, other objectives could be used as well which are more relevant in practice. We review the literature related to the described problem in section 2. In section 3, we provide the formal problem definition and describe its application to our real-world problem. In section 4, we present our solution approach. First, a suitable disjunctive graph representation is given that builds a basis for the following algorithms. Then, we present a construction heuristic to compute initial solutions which is an iterative sampling method based on job insertions. We define a neighborhood that reorders resources and exchanges operations. Such moves must take the given resource dependencies into account. The neighborhood is then used for a simulated annealing algorithm. In section 5, we present and discuss numerical results of our implementation. There, we evaluate the benefit of our modeling and compare our results to literature on the flexible job-shop problem. Finally, a conclusion is given in section 6.

2 RELATED WORK

A tremendous amount of research on scheduling was conducted in the last decades. A classification of scheduling problems is given by Graham et al. (1977). A more recent version can be found in the book of Pinedo (2012). The benefits of scheduling methods for the performance of semiconductor manufacturing facilities are known for years, consider for example Wein (1988). A recent overview of challenges and opportunities for the use of scheduling methods in the area of semiconductor manufacturing is provided by Mönch et al. (2011).

In this study, we treat specific constraints that have their origin in the diffusion area of semiconductor manufacturing facilities. We concentrate on the modeling of resource dependencies. Other important properties of that area, such as time constraints or batching, are not considered here. Still, ideas from this work could be combined with known approaches for those properties. Mönch et al. (2005) introduce a method for a complex scheduling problem with batching that uses dispatching heuristics in combination with a genetic algorithm. Yugma et al. (2012) consider additional properties including time constraints and propose an approach based on simulated annealing. The latter approach relies on disjunctive graphs, a widely used representation for scheduling problems introduced by Roy and Sussmann (1964). Since we consider machine internal components, the problem at hand is also related to cluster tool scheduling problems; Lee (2008) provides an overview of related work. Ding, Yi, and Zhang (2006) model specific behavior of machine components by introducing event graphs. Geiger, Kempf, and Uzsoy (1997) and Ham (2012) describe the scheduling of a wet-etch station. However, we are not aware of an approach that is applicable for the problem at hand.

The basic problem related to the one at hand is the classical job-shop scheduling problem which is already NP-hard (Garey, Johnson, and Sethi 1976). An overview of solution methods is given by Vaessens, Aarts, and Lenstra (1996). A classical solution approach is the shifting bottleneck heuristic of Adams, Balas, and Zawack (1988) which was improved in Dauzère-Pérès and Lasserre (1993). Simulated annealing was first used to tackle the job-shop scheduling problem by Van Laarhoven, Aarts, and Lenstra (1992). They use a connected neighborhood induced by swapping critical arcs. Vaessens (1995) discusses deterministic and probabilistic local search approaches for the job-shop problem. Nowicki and Smutnicki (1996) propose a taboo search method for which they introduce a neighborhood definition which employs blocks of jobs. Mati, Dauzère-Pérès, and Lahlou (2011) propose an algorithm for regular criteria that utilizes an efficient method to evaluate moves.

The flexible job-shop problem was first studied by Brucker and Schlie (1990). An extensive study of the problem including lower bounds for several test instances is given in Jurisch (1992). Hurink, Jurisch, and Thole (1994) present a tabu search method for the problem. A disjunctive graph model that allows to perform reassignments and reorderings in a uniform way was introduced in Dauzère-Pérès and Paulli (1997). Mastrolilli and Gambardella (2000) improve the results of Dauzère-Pérès and Paulli (1997) using a comparable taboo search approach. Gao, Sun, and Gen (2008) propose a hybrid genetic algorithm. Recently, the problem was successfully tackled using constraint programming based approaches (Pacino and Van Hentenryck 2011; Schutt, Feydy, and Stuckey 2013). A summary of experimental results for the flexible job-shop scheduling problem is given in Behnke and Geiger (2012).

The closest approach to our route graph model we are aware of is that of Kis (2003). It describes processing alternatives as a directed graph. His approach considers also nonlinear routes given as partial orderings of operations (called *and-subgraphs*), which are not a requisite for the problem at hand. Our approach is distinguished in particular from the one of Kis by the consideration of resource acquisitions.

A generalization of precedence constraints by Möhring, Skutella, and Stork (2004) allows a job to depend on further jobs which can be chosen from a set of alternatives. An extension of the flexible job-shop problem for multiple resources per operation was introduced by Dauzère-Pérès, Roux, and Lasserre (1998). It is further generalized in Dauzère-Pérès and Pavageau (2003) by occupying the resources of a specific operation for varying periods of time.

3 PROBLEM DESCRIPTION

In this section, we provide a formal description of our problem. Beforehand, we introduce some notation and define the concept of route graphs. This concept is subsequently used to specify valid routes for a job. Since we statically assign resources to operations, resource flexibility is achieved by allowing different routes. Finally, we indicate how the formal model can be applied to our real-world problem.

3.1 Preliminaries

Consider a graph $G = (V, E)$ with a set of nodes V and a set of edges E. For a node $v \in V$, we denote the set of incoming edges as $in(v) \subset E$ and the set of outgoing edges as $out(v) \subset E$. A *path* from $v_1 \in V$ to $v_k \in V$ is defined as a sequence of nodes $(v_1, v_2, \ldots, v_k)$ with $(v_i, v_{i+1}) \in E$ for all $1 \leq i < k$. Next, we introduce the term *two-terminal series parallel graph* according to the definition of Eppstein (1992) and repeat his definition here for completeness. Then, we use that definition to define the term route graph.

Definition 1 (Eppstein 1992) A directed graph G is *two-terminal series parallel*, with terminals s and t, if it can be produced by a sequence of the following operations:

1. *Initialization*: Create a new graph, consisting of a single edge directed from s to t.
2. *Parallel composition*: Given two two-terminal series parallel graphs X and Y, with terminals s_X, t_X, s_Y, and t_Y, form a new graph $G = P(X,Y)$ by identifying $s = s_X = s_Y$ and $t = t_X = t_Y$.
3. *Serial composition*: Given two two-terminal series parallel graphs X and Y, with terminals s_X, t_X, s_Y, and t_Y, form a new graph $G = S(X,Y)$ by identifying $s = s_X$, $t_X = s_Y$, and $t = t_Y$.

Definition 2 A two-terminal series parallel graph $G = (V, E)$ with terminals s and t is called *route graph* $R = (V, E, s, t)$ if, for all $v \in V \setminus \{s, t\}$, one of the following properties is true:

1. $|in(v)| = |out(v)| = 1$
2. All paths from s to t include v; i.e., they have the form $(s, \ldots, v, \ldots, t)$. Such a node v is called a *route separator* if $|in(v)| > 1$ and/or $|out(v)| > 1$.

A node of a route graph is called *operation*. A path from s to t in a route graph is called a *route*. The nodes of a route define a sequence of operations. In addition to the definition above, the start node s and the terminal node t are called *route separators* as well.

3.2 Problem Definition

We consider an extension of the flexible job-shop scheduling problem. Our problem can be described as follows: We are given a set of *jobs* $\mathcal{J}$ that have to be processed using a set of *resources* $\mathcal{M}$. For each job $J_i \in \mathcal{J}$, we are given a route graph $R_i = (O_i, E_i, s_i, t_i)$ whose routes specify feasible sequences of operations. For each job, one of those sequences of operations has to be performed. Each operation $o_{i,j} \in O_i$ has a processing time $p_{i,j} \in \mathbb{N}_0$ and a fixed set of resources $M_{i,j} \subset \mathcal{M}$ that are needed to process $o_{i,j}$. A resource can be utilized for only one operation at a time and the processing of an operation cannot be interrupted (no preemption allowed).

A schedule is characterized by a selection of routes and the start times for the operations of these routes. For each job J_i, a *route selection* $S_i \subset O_i$ describes a route $(s_i = o_{i,1}, \ldots, o_{i,|S_i|} = t_i)$ in the route graph of J_i. The *start times* $t_{i,j} \in \mathbb{Z}$ of selected *operations* $o_{i,j} \in S_i$ must consider processing times and preserve the precedence of operations. So, *completion times* $C_{i,j} = t_{i,j} + p_{i,j}$ of operations $o_{i,j}$ have to fulfill $C_{i,j} \leq t_{i,j+1}$ for all $i \in \{1, \ldots, |\mathcal{J}|\}$, $j \in \{1, \ldots, |S_i| - 1\}$. Since resource utilization is exclusive, we require for two scheduled operations $o_{i,j} \neq o_{k,l}$ with $M_{i,j} \cap M_{k,l} \neq \emptyset$ that either $t_{i,j} \geq C_{k,l}$ or $t_{k,l} \geq C_{i,j}$ holds. Our objective is to minimize the makespan. It is defined as the maximum completion time over all jobs, i.e.

$C_{\max} = \max\left\{C_{i,|S_i|} \;\middle|\; 1 \leq i \leq |\mathcal{J}|\right\}$. We refer to the problem defined in this section as the basic version of the *job-shop scheduling problem with extended route flexibility*.

3.2.1 Resource Acquisition

Next, we extend the basic version of the problem by an additional constraint. Consider two operations that are part of the same route and require a common resource. In some cases, we want to exclusively acquire the resource between such two operations; we prohibit other operations to use the resource in between. More formally, we are given a set of resource acquisitions as a subset $A_{i,j} \subset M_{i,j}$ for each operation $o_{i,j}$. For all given acquisitions $m \in A_{i,j}$, their release must be uniquely defined; there must exist an operation $o_{i,k}$ with $m \in M_{i,k}$ such that there is a path $P_m = (o_{i,j}, \ldots, o_{i,k})$ that does not contain a route separator. This path must be minimal in the sense that, for all operations $o_{i,h} \in P_m$ with $h \neq j$, $h \neq k$, we must have $m \notin M_{i,h}$. Note that a resource can be immediately reacquired: we allow $m \in A_{i,k}$. The resource acquisition constraint now imposes that for an acquisition of a resource $m \in A_{i,j}$ at an operation $o_{i,j}$ with a corresponding release operation $o_{i,k}$, there must not be any other operation that uses m in the time between $t_{i,j}$ and $C_{i,k}$. So, for all operations $o_{x,y}$ ($\neq o_{i,j}$, $\neq o_{i,k}$) with $m \in M_{x,y}$, either $t_{x,y} \geq C_{i,k}$ or $C_{x,y} \leq t_{i,j}$ must hold.

Figure 2 presents an example of a route graph including resource acquisitions. That route graph allows six different routes between s and t: three alternatives in the first section times two alternatives in the second section. The nodes represent operations and are labeled with the resources that are required to process them. Resource acquisitions are indicated by a superscript “A” and a dashed line (which is not an edge of the route graph) to the release operation.

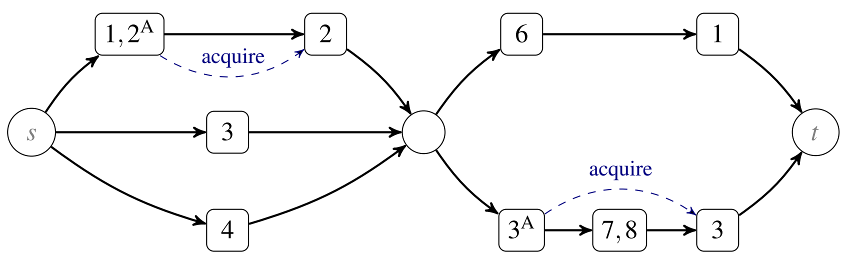

Figure 2: This example illustrates a route graph describing six different routes.

3.3 Application

The formal model given above can be applied in a straightforward way to the real-world problem described in section 1. In the real-world problem, we are given a linear route of processing steps for each job. Each step must be assigned to a machine which can be chosen from a set of suitable machines.

To apply the formal model, each component of a furnace is represented as a resource. Each processing step is decomposed into a set of paths of operations. Each of these paths reflects a sequence of operations on the components of a furnace. Now, each decomposition of a processing step into a set of paths of operations yields a partial route graph: The paths of operations are inserted between two separator nodes. Finally, we concatenate for each job the partial route graphs of its processing steps. This procedure creates route graphs applicable to our formal model.

4 SOLUTION METHOD

This section presents two heuristic methods to solve the job-shop scheduling problem with extended route flexibility. Both are based on a disjunctive graph representation, presented in section 4.1, which takes the extended route flexibility into account. Section 4.2 explains a construction algorithm based on job insertions. Section 4.3 presents a simulated annealing approach which is based on a neighborhood induced by reorderings of resources and exchanges of paths of operations between separator nodes.

4.1 Disjunctive Graph Representation

Disjunctive graphs represent combinatorial properties of schedules: Nodes represent operations and edges represent precedence relations. For the problem at hand, we adapt ideas given in Dauzère-Pérès and Paulli (1997). We use the notation introduced in section 3.2. The nodes V of our disjunctive graph are the artificial start node α, the artificial end node Ω, and all nodes of the route graphs of all jobs: $V = \{\alpha, \Omega\} \cup \bigcup_{1 \leq i \leq |\mathcal{J}|} O_i$.

Edges represent precedence relations of routes and precedence relations of resources. For each route selection S_i with $1 \leq i \leq |\mathcal{J}|$, all edges $(o_{i,k}, o_{i,l}) \in E_i \cap (S_i \times S_i)$ are added to the disjunctive graph. They represent route precedences and correspond to conjunctive edges in the classical representation. For each resource $m \in M$, its scheduled operations are given by $R_m = \bigcup_{1 \leq i \leq |\mathcal{J}|} \{o_{i,j} \in S_i \mid m \in M_{i,j}\}$. For feasible schedules, R_m must be totally ordered; this leads to $|R_m| - 1$ edges that are added to represent resource precedences. Those edges correspond to disjunctive edges in the classical representation. Finally, we connect the artificial nodes α and Ω. We add $|M| + |\mathcal{J}|$ edges from α: One to each starting operation of a route, and one to each operation that uses a resource at first. Analogously, we introduce $|M| + |\mathcal{J}|$ edges to connect final operations of routes and resources usages with Ω. An additional edge (α, Ω) directly connects the artificial nodes to facilitate the handling of partially computed schedules. Hence, our graph consists of $|M| + |\mathcal{J}| + 1$ disjoint paths from α to Ω. The in-degree (and out-degree) of each scheduled node is given by the number of its resources plus one for the route. Unscheduled operations, i.e. those nodes that are not part of a route selection, remain a part of the graph but are disconnected.

Next, we include the resource acquisition constraint as follows. For each acquisition of a resource $m \in A_{i,j}$ at an operation $o_{i,j}$ with a corresponding release operation $o_{i,k}$, the following property must hold: For all operations $o_{x,y}$ ($\neq o_{i,j}$, $\neq o_{i,k}$) with $m \in M_{x,y}$ there must not be any path in the disjunctive graph that has the form $(o_{i,j}, \ldots, o_{x,y}, \ldots, o_{i,k})$. Consequently, $o_{i,k}$ must directly follow $o_{i,j}$ in the total ordering of R_m if both operations are scheduled. This prescribes the presence of a fixed edge from $o_{i,j}$ to $o_{i,k}$ related to the usage of the resource m.

Earliest and latest start times of scheduled operations are determined using a standard method. They are computed in a time linear to the number of scheduled nodes ($\mathcal{O}(|\bigcup_{1 \leq i \leq |\mathcal{J}|} S_i|)$), by determining a topological ordering and traversing nodes in that order. The makespan is given by the length of the longest path from α to Ω. Infeasible schedules are identified by cycles in the disjunctive graph.

4.2 Construction Heuristic

In this section, we describe a construction heuristic that uses ideas from the insertion method of Werner and Winkler (1995) for the classical job-shop scheduling problem and the construction heuristic described in Yugma et al. (2012). The idea of this method is to successively insert the operations of jobs and probe for each operation its best insertion position. We sort the jobs by their shortest possible route duration, i.e. the shortest path in the route graph with edge weights given by processing durations. We insert jobs in that order, starting with the longest route duration. We successively insert operations: For each job, we iterate the sections between separator nodes—starting at the start node of the route graph. We determine the best insertion by probing all possible insertion positions for all paths between the current separator nodes. An insertion position is described as a sequence of nodes to be inserted together with insertion positions for involved resources (given by edges of the disjunctive graph). The best insertion is executed. Then, we continue with the section between the following pair of separator nodes. To evaluate the best insertion, we rate corresponding partially computed schedules by determining their makespan. Often, different insertion positions yield the same makespan. In those cases, we break ties by considering the total completion time of the partial schedule. We take care of resource acquisition constraints by forbidding all insertions that would violate the corresponding constraint.

4.3 Simulated Annealing

In this section, we describe a meta-heuristic based on simulated annealing to improve solutions found by the construction heuristic. We introduce a neighborhood structure based on two kinds of moves. The first move changes selected routes by modifying the route selection between two successive route separators of the route graph (*route move*). The second move reorders the usage of resources (*resource move*).

First, we introduce the term *route move* that specifies the modification of a route selection between two route separators. It is defined by a path of operations between two route separators that should be removed, and a different path of operations between the same route separators that should be inserted. In addition, the move specifies the sequencing position of all resources required by the operations to be inserted. We check the feasibility of route moves by executing the move and checking if the disjunctive graph contains cycles. The efficiency of these checks is improved by using properties given in Dauzère-Pérès, Roux, and Lasserre (1998) as necessary conditions for the feasibility of a move.

Second, we introduce the term *resource move* that specifies the re-positioning of one resource utilization. It is given by an operation, and a resource to be moved used at this operation. In addition, the target position is specified by a second operation using the same resource. The resource is then moved from its previous position to a position behind that resource. To move a resource to the first position, the artificial start node is given as target operation. The feasibility of a resource move can efficiently be determined by considering only adjacent nodes in the disjunctive graph. For this, we use properties of the graph proven in Dauzère-Pérès, Roux, and Lasserre (1998).

In each step of the simulated annealing algorithm, we need to select some schedule from the neighborhood to be evaluated next. To obtain such a neighbor, we execute a move that is determined as follows. We randomly select a node $v \in V \setminus \{\alpha, \Omega\}$. If v is scheduled, we generate all feasible resource moves that re-position a resource used by that operation and randomly pick one of those moves. If v is not scheduled, we generate all feasible route moves that insert v into the schedule. Again, we randomly pick one of those moves. In this way, the balancing between resource moves and route moves is automatically done. It depends on the flexibility to choose other routes that is proportional to the number of unscheduled nodes. Again, we take care of resource acquisitions by forbidding moves that would violate the corresponding constraint.

We maintain a temperature T that is initialized by a fraction β of the objective value of the initial solution. After each step that evaluates a candidate schedule with an objective value f_{n+1}, we immediately accept all schedules that improve the objective value f_n of the current solution. Otherwise, we consider the difference $\Delta = f_{n+1} - f_n$ between the current and the candidate schedule. We accept the new schedule with a probability of $p = e^{\frac{-\Delta}{T}}$. We use a geometrical cooling schedule: After each step, the temperature is multiplied by a factor $\alpha < 1$. After some initial attempts, we settled on values of $\beta = 2.5\%$ and $\alpha = 0.9995$ for all our numerical experiments.

5 EXPERIMENTAL RESULTS

The objectives of this section are twofold. First, we justify the additional complexity introduced by our modeling, by comparing on a real-world instance results obtained with our model to those obtained with a less complex model. Second, we show that our implementation performs well on test instances of Hurink, Jurisch, and Thole (1994) for the flexible job-shop scheduling problem.

The degree of details in our modeling induces a considerable complexity. We illustrate the benefits of the more complex modeling using an instance from a real-world fab of STMicroelectronics. We compare two schedules: The first one is obtained using the presented job-shop model with extended route flexibility, the second is computed using a simpler model based on a flexible job-shop with multiple resources per operation. In the simpler model, a consistent resource usage is guaranteed neither for cooling operations including boats, nor for loading and unloading operations at load ports. Therefore, we neglect those properties in that model. Consequently, the computed start times of operations may differ from actual

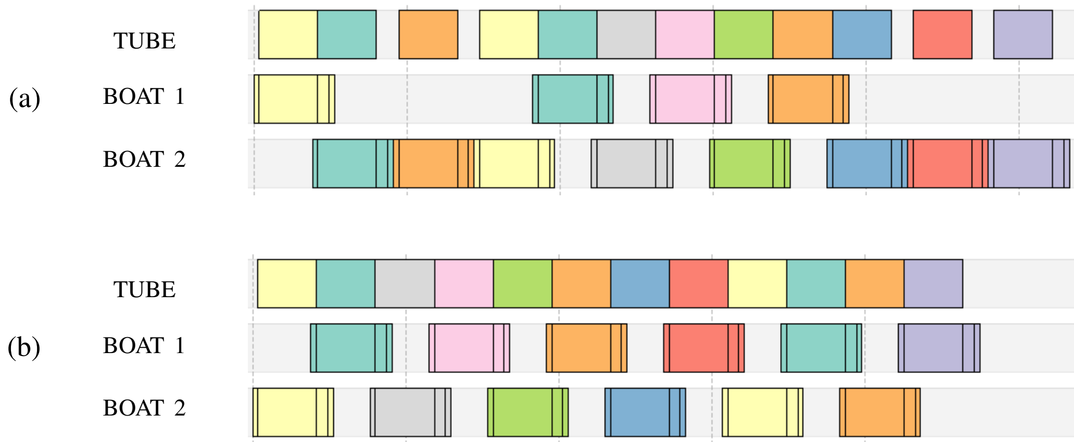

Figure 3: A comparison of schedules obtained by different models.

ones. We simulate the real performance of the sequence determined by the simpler model, by using the extended model to compute a schedule with the same sequence of operations—including loading, cooling and unloading. Figure 3 compares the schedules obtained by the two approaches. This extract of a schedule visualizes the jobs scheduled on a machine with one tube and two boats. Schedule (a) is obtained by the simpler model, Schedule (b) by the model with extended route flexibility. The makespan of the schedule corresponding to the simpler model is 74 hours. This is significantly larger than the makespan of 66 hours obtained with the extended model. This difference stems from idle periods in the usage of the tube in the simpler model. Those idle periods are caused by the inefficient allocation of loading, cooling, and unloading operations to boats. This is because the simpler model neglects those operations. Note that the alternating usage of boats is most efficient. However, we do not see how this observation could lead to a simpler model. One reason is that a job could be processable only one of two boats. This example illustrates the relevance of resource dependency constraints for scheduling operations in the diffusion area of semiconductor manufacturing.

We deal with a new problem for which no test instances are available. Since we consider a generalization of the flexible job-shop scheduling problem, we use related instances to evaluate our method. Behnke and Geiger (2012) provide a helpful collection of results for such a comparison. Experiments show that our approach obtains results comparable to dedicated methods. We implemented the presented algorithm in C++11 and compiled our software using GCC 4.8.0. Experiments were run on a Intel Xeon W3530 2.80 GHz machine. Table 1 provides results for the instances of Hurink, Jurisch, and Thole (1994). These instances consist of three sets of test problems (*edata*, *rdata*, and *vdata*), each incorporating a different degree of resource flexibility. Column C_{start} reports the makespan of solutions found by our construction heuristic. For all instances, the time needed to run the construction heuristic is far less than one second. Column C_{best} provides the makespan of solutions obtained by improving the initial solution using simulated annealing. We limited the time per run to 5 minutes per instance. For most instances, we find the best result far before the expiration of that time limit (several seconds). For some instances, we performed additional numerical experiments with a computation time of 6 hours. No further improvements were observed in this setting. Column C_{publ} states the best makespans provided in literature; they are labeled with a letter indicating which publication found the result first. For the *edata*, *rdata*, and *vdata* instances, we observe an average gap to the best known results of 5.5%, 1.7%, and 0.6%, respectively. This shows that our approach performs well on these instances, although large gaps are observed on some instances (*edata la36*: 20.3%, *rdata la25*: 6.1%, *vdata la05*: 3.3%). Additional work is on-going to improve these results.

Table 1: Results for flexible job-shop instances of Hurink, Jurisch, and Thole (1994). Abbreviations: **B**: Behnke and Geiger (2012), **D**: Dauzère-Pérès and Paulli (1997), **J**: Jurisch (1992), **M**: Mastrolilli and Gambardella (2000), **P**: Pacino and Van Hentenryck (2011), **S**: Schutt, Feydy, and Stuckey (2013).

	edata			rdata			vdata		
	C_{start}	C_{best}	C_{publ}	C_{start}	C_{best}	C_{publ}	C_{start}	C_{best}	C_{publ}
mt06	56	55	55 J	54	47	47 J	47	47	47 J
mt10	997	959	**871** J	831	727	**686** D	932	655	655 J
mt20	1407	1185	**1088** J	1320	1026	**1022** M	1063	1025	**1022** J
la01	840	609	609 J	662	587	**570** S	700	579	**570** J
la02	761	656	**655** J	667	538	**529** S	614	540	**529** J
la03	730	568	**550** J	619	493	**477** S	526	487	**477** M
la04	694	581	**568** J	656	508	**502** J	596	506	**502** J
la05	659	503	503 J	580	462	**457** J	557	472	**457** M
la06	1102	833	833 J	857	806	**799** M	1004	805	**799** J
la07	887	778	**762** J	818	754	**749** S	834	753	**749** M
la08	929	867	**845** J	989	769	**765** M	971	769	**765** M
la09	1093	884	**878** J	1017	857	**853** M	945	859	**853** D
la10	1019	866	866 J	860	808	**804** M	915	807	**804** J
la11	1220	1106	**1103** J	1244	1073	**1071** J	1206	1075	**1071** J
la12	1215	960	960 J	1035	939	**936** D	992	939	**936** J
la13	1156	1053	1053 J	1178	1042	**1038** J	1106	1042	**1038** J
la14	1345	1123	1123 J	1276	1073	**1070** J	1155	1072	**1070** D
la15	1234	1125	**1111** J	1285	1092	**1089** S	1167	1090	**1089** D
la16	1285	947	**892** J	933	717	717 J	772	717	717 J
la17	807	735	**707** J	872	646	646 J	646	646	646 J
la18	1088	878	**842** J	826	669	**666** J	783	663	663 J
la19	1058	823	**796** J	848	703	**700** M	728	617	617 J
la20	1033	983	**857** J	1164	756	756 J	991	756	756 J
la21	1412	1092	**1009** P	1149	865	**835** M	1031	813	**804** B
la22	1178	1012	**880** P	1103	796	**760** M	946	752	**736** B
la23	1214	1063	**950** M	1193	892	**842** M	1200	830	**815** M
la24	1314	1022	**908** P	1162	835	**808** M	1079	797	**775** B
la25	1189	1050	**936** P	1052	839	**791** M	1071	768	**756** M
la26	1468	1241	**1107** P	1424	1083	**1061** M	1250	1062	**1054** M
la27	1594	1260	**1181** P	1435	1105	**1091** M	1425	1091	**1084** B
la28	1766	1183	**1142** P	1554	1094	**1080** M	1363	1075	**1070** M
la29	1516	1259	**1111** P	1390	1003	**998** M	1229	998	**994** M
la30	1627	1310	**1195** P	1531	1140	**1078** M	1288	1077	**1069** M
la31	2056	1631	**1538** S	1988	1523	**1521** M	1826	1522	**1520** M
la32	2521	1717	**1698** D	2080	1668	**1659** M	1940	1659	**1658** J
la33	2086	1580	**1547** J	1837	1503	**1499** M	1687	1500	**1497** M
la34	2117	1664	**1599** M	2001	1539	**1536** M	1926	1536	**1535** M
la35	2147	1736	1736 J	1930	1557	**1550** M	1895	1552	**1549** M
la36	1782	1395	**1160** P	1415	1061	**1030** D	1232	948	948 J
la37	1588	1454	**1397** J	1579	1089	**1077** M	1269	986	986 J
la38	1601	1266	**1141** S	1247	1016	**962** M	1184	943	943 J
la39	1731	1343	**1184** J	1430	1040	**1018** B	1045	922	922 J
la40	1469	1205	**1144** P	1268	995	**970** M	1140	955	955 J

6 CONCLUSION

In this paper, we introduced a job-shop scheduling problem with extended route flexibility to take properties and components of machines in semiconductor manufacturing facilities into account. We use the concept of extended route flexibility to force different operations to use the very same resource. This feature is extended by resource acquisitions that can exclusively reserve a resource between two operations. We are not aware of other approaches that provide a route flexibility that allows modeling resource acquisitions. Our approach obtains better results on industrial instances than simpler modeling approaches. An interesting direction for future research is the application of the approach to areas beyond semiconductor manufacturing. Resource acquisition constraints appear, for example, in the scheduling of railway maintenance operations (Ramond, de Almeida, and Dauzère-Pérès 2006).

We presented a simulated annealing meta-heuristic based on a neighborhood which is induced by two kinds of moves. We expect that our results can be improved by incorporating properties that are already exploited by established methods for the flexible job-shop scheduling problem. Also, the utilization of other meta-heuristics such as taboo search, variable neighborhood search or genetic algorithms seems very promising. Still, we showed that our generalized method can compete with dedicated methods for the flexible job-shop problem.

In this study, we concentrate on optimizing the makespan. However, many other objectives are more important semiconductor manufacturing. So, it would be useful to consider them as well or to take multiple criteria into account. A very important property of machines used in the diffusion area is their batching capability: They can process multiple lots of wafers at the same time. Additionally, temporal constraints such as maximum time lags play an important role in this area. We aim at extending our approach to develop a scheduling method that is fully applicable in practice.

ACKNOWLEDGMENTS

This work is supported by the ENIAC European Project INTEGRATE. We would like to thank all contributors from STMicroelectronics who supported our work with valuable information and discussions. Also, we would like to thank the anonymous reviewers for their suggestions and comments.

REFERENCES

Adams, J., E. Balas, and D. Zawack. 1988. "The shifting bottleneck procedure for job shop scheduling". *Management Science* 34 (3): 391–401.

Behnke, D., and M. J. Geiger. 2012. "Test Instances for the Flexible Job Shop Scheduling Problem with Work Centers". Technical report, Helmut-Schmidt-Universität, Universität der Bundeswehr Hamburg.

Brucker, P., and R. Schlie. 1990. "Job-shop scheduling with multi-purpose machines". *Computing* 45 (4): 369–375.

Dauzère-Pérès, S., and J.-B. Lasserre. 1993. "A modified shifting bottleneck procedure for job-shop scheduling". *International Journal of Production Research* 31 (4): 923–932.

Dauzère-Pérès, S., and J. Paulli. 1997. "An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem using tabu search". *Annals of Operations Research* 70:281–306.

Dauzère-Pérès, S., and C. Pavageau. 2003. "Extensions of an integrated approach for multi-resource shop scheduling". *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 33 (2): 207–213.

Dauzère-Pérès, S., W. Roux, and J. Lasserre. 1998. "Multi-resource shop scheduling with resource flexibility". *European Journal of Operational Research* 107 (2): 289–305.

Ding, S., J. Yi, and M. T. Zhang. 2006. "Multicluster tools scheduling: An integrated event graph and network model approach". *IEEE Transactions on Semiconductor Manufacturing* 19 (3): 339–351.

Eppstein, D. 1992. “Parallel recognition of series-parallel graphs”. *Information and Computation* 98 (1): 41–55.

Gao, J., L. Sun, and M. Gen. 2008. “A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems”. *Computers & Operations Research* 35 (9): 2892–2907.

Garey, M. R., D. S. Johnson, and R. Sethi. 1976. “The complexity of flowshop and jobshop scheduling”. *Mathematics of Operations Research* 1 (2): 117–129.

Geiger, C. D., K. G. Kempf, and R. Uzsoy. 1997. “A tabu search approach to scheduling an automated wet etch station”. *Journal of Manufacturing Systems* 16 (2): 102–116.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. Rinnooy Kan. 1977. “Optimization and approximation in deterministic sequencing and scheduling: a survey”. *Annals of Discrete Mathematics* 5:287–326.

Ham, M. 2012. “Integer programming-based real-time dispatching (i-RTD) heuristic for wet-etch station at wafer fabrication”. *International Journal of Production Research* 50 (10): 2809–2822.

Hurink, J., B. Jurisch, and M. Thole. 1994. “Tabu search for the job-shop scheduling problem with multi-purpose machines”. *OR Spectrum* 15 (4): 205–215.

Jurisch, B. 1992. *Scheduling jobs in shops with multi-purpose machines*. Ph. D. thesis, Fachbereich Mathematik/Informatik, Universität Osnabrück.

Kis, T. 2003. “Job-shop scheduling with processing alternatives”. *European Journal of Operational Research* 151 (2): 307–332.

Lee, T.-E. 2008. “A review of scheduling theory and methods for semiconductor manufacturing cluster tools”. In *Proceedings of the 2008 Winter Simulation Conference*, edited by S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, and J. W. Fowler, 2127–2135. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mastrolilli, M., and L. M. Gambardella. 2000. “Effective neighbourhood functions for the flexible job shop problem”. *Journal of Scheduling* 3 (1): 3–20.

Mati, Y., S. Dauzère-Pérès, and C. Lahlou. 2011. “A general approach for optimizing regular criteria in the job-shop scheduling problem”. *European Journal of Operational Research* 212 (1): 33–42.

Möhring, R. H., M. Skutella, and F. Stork. 2004. “Scheduling with AND/OR precedence constraints”. *SIAM Journal on Computing* 33 (2): 393–415.

Mönch, L., H. Balasubramanian, J. W. Fowler, and M. E. Pfund. 2005. “Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times”. *Computers & Operations Research* 32 (11): 2731 – 2750.

Mönch, L., J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, and O. Rose. 2011. “A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations”. *Journal of Scheduling* 14 (6): 583–599.

Nowicki, E., and C. Smutnicki. 1996. “A fast taboo search algorithm for the job shop problem”. *Management Science* 42 (6): 797–813.

Pacino, D., and P. Van Hentenryck. 2011. “Large neighborhood search and adaptive randomized decompositions for flexible jobshop scheduling”. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Volume 3, 1997–2002. AAAI Press.

Pinedo, M. 2012. *Scheduling: theory, algorithms, and systems*. Springer.

Ramond, F., D. de Almeida, and S. Dauzère-Pérès. 2006. “Enhanced Operation Scheduling within Railcar Maintenance Centers”. In *Proceedings of the 7th World Congress on Railway Research*, Paper 485.

Roy, B., and B. Sussmann. 1964. “Les problemes d’ordonnancement avec contraintes disjonctives”. *Note DS* 9.

Schutt, A., T. Feydy, and P. J. Stuckey. 2013. “Scheduling Optional Tasks with Explanation”. In *Principles and Practice of Constraint Programming*, edited by P. Stuckey, 628–644. Springer.

Vaessens, R. J., E. H. Aarts, and J. K. Lenstra. 1996. “Job shop scheduling by local search”. *INFORMS Journal on Computing* 8 (3): 302–317.

Vaessens, R. J. M. 1995. *Generalized Job Shop Scheduling: Complexity and Local Search*. Ph. D. thesis, Eindhoven University of Technology.

Van Laarhoven, P. J., E. H. Aarts, and J. K. Lenstra. 1992. "Job shop scheduling by simulated annealing". *Operations Research* 40 (1): 113–125.

Wein, L. M. 1988. "Scheduling semiconductor wafer fabrication". *IEEE Transactions on Semiconductor Manufacturing* 1 (3): 115–130.

Werner, F., and A. Winkler. 1995. "Insertion techniques for the heuristic solution of the job shop problem". *Discrete Applied Mathematics* 58 (2): 191 – 211. Workshop on Discrete Algoritms.

Yugma, C., S. Dauzère-Pérès, C. Artigues, A. Derreumaux, and O. Sibille. 2012. "A batching and scheduling algorithm for the diffusion area in semiconductor manufacturing". *International Journal of Production Research* 50 (8): 2118–2132.

AUTHOR BIOGRAPHIES

SEBASTIAN KNOPP is a Ph.D. Student at the Center of Microelectronics in Provence of the EMSE where he works on scheduling problems in semiconductor manufacturing. He received a Diploma degree in computer science from the University of Karlsruhe, Germany, in 2006. From 2006 to 2013, he worked as a Software Developer at PTV Group, Germany, on the implementation and design of algorithms in the field of logistics and digital maps. He was concerned with algorithms for routing in large road networks, map matching, and fault tolerant geocoding, as well as the scheduling of driving time and rests periods as a component for real world vehicle routing problems. His email address is sebastian.knopp@emse.fr.

STÉPHANE DAUZÈRE-PÉRÈS is a Professor at the Center of Microelectronics in Provence (CMP) of the EMSE. He received the Ph.D. degree from the Paul Sabatier University in Toulouse, France, in 1992; and the H.D.R. from the Pierre and Marie Curie University, Paris, France, in 1998. He was a Postdoctoral Fellow at the Massachusetts Institute of Technology, U.S.A., in 1992 and 1993, and Research Scientist at Erasmus University Rotterdam, The Netherlands, in 1994. He has been Associate Professor and Professor from 1994 to 2004 at the Ecole des Mines de Nantes in France where he headed the team Production and Logistic Systems (about 20 members) between 1999 and 2004. He was invited Professor at the Norwegian School of Economics and Business Administration, Bergen, Norway, in 1999. Since March 2004, he is Professor at the Ecole des Mines de Saint-Etienne, where he headed the research department Manufacturing Sciences and Logistics (SFL, about 20 members) from 2004 to 2013 and the CMP from 2013 to 2014. His research interests broadly include modeling and optimization of operations at various decision levels (from real-time to strategic) in manufacturing and logistics, with a special emphasis on semiconductor manufacturing. He has published 49 papers in international journals and contributed to more than 120 communications in conferences. Stéphane Dauzère-Pérès has coordinated multiple academic and industrial research projects, and also five conferences. His email address is stephane.dauzere-peres@emse.fr.

CLAUDE YUGMA is an Associate Professor at the Center of Microelectronics in Provence of the EMSE. He received the Ph.D. degree from the Institut National Polytechnique of Grenoble, France, in 2003; He was a Postdoctoral Researcher at the Ecole Nationale Supérieure de Génie Industriel, Grenoble, from 2003 to 2004 and from 2005 to 2006 at the Provence Microelectronics Center. His email address is claude.yugma@emse.fr.