Proceedings of the 2025 Winter Simulation Conference
E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.

MODELING AND SOLVING COMPLEX JOB-SHOP SCHEDULING PROBLEMS FOR RELIABILITY LABORATORIES

Jessica Hautz[1], Andreas Klemmt[2], and Lars Mönch[3]

[1]Kompetenzzentrum Automobil- und Industrieelektronik GmbH, Villach, AUSTRIA
[2]Infineon Technologies Dresden GmbH, Dresden, GERMANY
[3]Dept. of Mathematics and Computer Science, University of Hagen, Hagen, GERMANY

ABSTRACT

We consider job-shop scheduling problems with stress test machines. Several jobs can be processed at the same time on such a machine if the sum of their sizes does not exceed its capacity. Only jobs with operations of the same incompatible family can be processed at the same time on a machine. The machine can be interrupted to start a new or to unload a completed job. A conditioning time is required to reach again the temperature for the stress test. The machine is unavailable during the conditioning process. Operations that cannot be completed before a conditioning activity have to continue with processing after the machine is available again. The makespan is to be minimized. A constraint programming formulation and a variable neighborhood search scheme based on an appropriate disjunctive graph model are designed. Computational experiments based on randomly generated problem instances demonstrate that the algorithms perform well.

1 INTRODUCTION

Semiconductor manufacturing deals with producing integrated circuits (ICs). The production requires the wafer fabrication, sort, assembly, and test stages. Chips are produced in the wafer fabrication stage on wafers, small discs made from silicon or gallium arsenide. Afterwards the wafers are diced, and defective chips are sorted out at the second stage. Only high-quality chips are assembled and packaged in assembly facilities, whereas tests are performed on the final devices in test facilities (Mönch et al. 2013). Before the production based on customer orders can start, the products must be designed and qualified for production in reliability laboratories where a series of measurement and stress operations under different temperature and stress conditions is performed. The qualification usually takes several weeks.

Semiconductor reliability laboratories can be modeled as complex job shops with some unusual facets. They contain machines that offer the same functionality, i.e., we have parallel machines, we also refer to them as machine groups. Reentrant flows are common in such facilities, i.e., the same machine group is visited several times by a single job. The machines for measurement steps, so-called testers, can be modeled as s-batching machines, whereas machines for stress tests can be seen as machines with p-batching and job availability. A p-batch is a group of jobs that are processed at the same time on a batch processing machine (Fowler and Mönch 2022), whereas a s-batch is a group of jobs that are processed in a consecutive manner on a single machine. Significant setup times occur when adjacent s-batches belong to different families. The stress test machines can be interrupted to start a new job or to unload a completed job. A conditioning time is required to reach again the necessary temperature for the interrupted stress test on a machine. The machine is unavailable during the conditioning process. Jobs with operations that cannot be completed before a conditioning activity have to continue with processing after the machine is available again.

While we have studied scheduling problems for single and parallel stress test machines (Hautz et al 2024; Hautz et al. 2025), we are not aware of any paper where stress test machines are considered in job-shop scheduling problems. In the present paper, we initiate the modeling of stress test machines in disjunctive graph representations for complex job shops. Moreover, we report the results of constraint programming (CP) and a variable neighborhood search (VNS) scheme for the scheduling problem at hand.

The paper is organized as follows. In the next section, we describe the scheduling problem, and discuss and analyze related work. The disjunctive graph model is discussed in Section 3. The different solution approaches, namely a CP approach and a VNS scheme are presented in Section 4. Results of computational experiments are reported in Section 5. Finally, conclusions and future research directions are provided in Section 6.

2 PROBLEM SETTING

2.1 Scheduling Problem

We consider n jobs $J_1, \dots, J_n$ with operations $o_{jk}, k = 1, \dots, n_j$ for job $J_j, j = 1 \dots, n$. Each job J_j, has a ready time $r_j \geq 0$. The operations have to be performed in the prescribed order $o_{j1} \rightarrow o_{j2} \rightarrow \cdots \rightarrow o_{jn_j}$. The machines which can be used to perform an operation are known and belong to the process flow, the route, associated with a job. Each operation o_{jk} has a processing time p_{jk} which is independent of the machine where the operation is executed.

In the present paper, we distinguish stress test machines from tester machines where measurement steps are executed. We focus on modeling the stress test machines. Although the tester machines are s-batch machines, for the sake of simplicity and due to space limitations, we consider them as regular, i.e. non-batching machines. Each stress test operation o_{jk} has a size s_{jk} measured in number of stress boards to carry chips. Moreover, operation o_{jk} belongs to a family f_{jk}. Several operations that can be processed on a machine m and belong to the same family can be processed together in a batch on m until the sum of the job sizes does not exceed the maximum batch size B_m. We assume job availability, i.e., a job in a batch must be removed from the machine if the operation of the job is completed. The processing of all unfinished jobs of the current batch is stopped for the removing activity. This results in a reduced stress temperature. To reach again the correct stress temperature, a condition time $cond_m$ is required for m (El-Kareh and Hutter 2020). Job j can be added to an already processed batch at time t if the following three conditions are fulfilled:

1. It holds $r_{jk} \geq t$, i.e., the operation o_{jk} is ready for processing at time t.
2. The operations associated with the jobs of the current batch and o_{jk} belong to the same family.
3. Its size s_{jk} of o_{jk} fits into the batch.

A conditioning time $cond_m$ is again required after the job is added to an already processed batch on a machine m (El-Kareh and Hutter 2020). The conditioning time is enlarged when operations of other jobs are finished during the time span of a conditioning activity or other jobs are added to the batch within this time span. The remaining jobs of the batch continue with processing after a conditioning activity is completed. Therefore, resumable operations of jobs are assumed. The makespan C_{max} is considered. Using the three-field notation from deterministic machine scheduling the problem at hand can be stated as follows:

$$FJ|r_j, prec, recrc, p-batch, incompatible, s_j, B_m, cond_m, r-a\,,|C_{max}, \tag{1}$$

where FJ, $prec$, and $recrc$ refer to a flexible job shop, precedence constraints, and reentrant process flows, respectively, $p-batch, incompatible$ indicates p-batch processing with incompatible families, and $cond_m, r-a$ describe the machine-specific conditioning time and the resumable operations of the jobs, respectively. It is shown by Hautz et al. (2025) that $1|\,p-batch, incompatible, s_j, B, cond_m, r-a|C_{max}$, a special case of (1), is already NP-hard. Hence, we have to look for efficient heuristics to tackle large-sized problem instances in appropriate computing time.

2.2 Related Work

Flexible job shop scheduling approaches are surveyed by Dauzere-Peres et al. (2024). However, job shops with the characteristics of stress test machines are not described in this paper. Next, we discuss papers related to disjunctive graph modeling for job shops. Knopp et al. (2017) introduce the batch-oblivious approach for complex job shops with p-batching, i.e., batches are treated by modifying edge weights rather than introducing additional batching nodes in the disjunctive graph. Rocholl and Mönch (2025) extend this concept to flexible flow shops with s-batching machines. In the present paper, we extend the batch-oblivious approach towards modeling stress test machines. These machines are p-batching machines, but the machines can or must be interrupted for including new jobs and removing jobs with already completed operations.

Because of the limited machine availability of stress test machines, work for job-shop scheduling with machine unavailability periods is discussed next. Mauguière et al. (2005) study job-shop scheduling problems for the C_{max} measure. Generalized unavailability periods for machines are assumed. Branch & bound algorithms are proposed for the different types of unavailable periods. Azem et al. (2012) consider a job-shop scheduling problem with C_{max} measure where operations can be interrupted by machine unavailability periods movable within prescribed time windows. Tamssaouet et al. (2018) study a job-shop scheduling problem with machine availability constraints. The disjunctive graph model is extended to allow for machine unavailability. Simulated annealing and tabu search are applied. A job-shop scheduling problem with unavailable periods is studied by Lin and Ying (2020). The jobs are non-preemptive. A simulated annealing scheme is proposed. However, in these papers p-batching is not considered and the interruption regime for jobs due to the unavailability periods is often different to the one assumed in the present paper. To the best of our knowledge, only the papers Hautz et al. (2024), (2025) propose exact and heuristic scheduling algorithms for stress test machines. However, only single- and parallel-machine situations are investigated. In the present paper, we will extend the machine environment to flexible job shops.

3 DISJUNCTIVE GRAPH MODEL

3.1 Modeling Stress Test Machines

A disjunctive graph $G = (V, E)$ with a set of nodes V and a set of edges or arcs E is considered. The nodes of the graph represent the operations of problem (1), the artificial start and end operations 0 and $*$, as well as artificial end nodes o_{j,n_j+1} for each job j. Nodes associated with operations of the same job are connected with directed arcs (conjunctive arcs) according to the given precedence constraints, while operations from different jobs with additional precedence constraints are also connected by conjunctive arcs. Disjunctive arcs are used between nodes that can be executed on the same machine to model sequencing decisions. To derive a conjunctive graph, a machine has to be assigned to each operation, and the disjunctive arcs have to be directed accordingly such that the resulting graph is acyclic. In a conjunctive graph of problem (1), the weight of all edges $(0, o_{j1}) \in E$, connecting the artificial start node 0 with the first operation $o_{j,1}$ of job j, are set to the ready date r_j. For all edges $(0, o_m)$ connecting the artificial start node with the initial operation o_m scheduled on machine m, the edge weight is set to zero. For each node $v \in V$ in a conjunctive graph, we denote its route successor by $r(v) \in V \setminus \{0\}$ and its machine successor by $m(v) \in V \setminus \{0\}$. Analogously, its route predecessor is denoted by $r^{-1}(v) \in V \setminus \{*\}$ and its machine predecessor by $m^{-1}(v) \in V \setminus \{*\}$. Furthermore, the machine chosen for node v is denoted as m_v, and the sets of route successors and predecessors associated with additional precedence constraints, i.e. from different jobs, are denoted as $SR(v)$ and $PR(v)$, respectively.

We extend the batch-oblivious approach for disjunctive graphs of Knopp et al. (2017) to model stress test machines. In contrast to Knopp et al. (2017), the batches within problem (1) do not have the same start and completion times, as unequal processing times for operations belonging to the same job family are possible and job availability is assumed. Moreover, conditioning periods and job preemptions have to be tackled. Given a conjunctive graph, the start time S_v of an operation $v \in V \setminus \{0,*\}$ can be derived by computing the longest path from 0 to v, which can be recursively computed as

$$S_v = \max\{S_{r^{-1}(v)} + l_{r^{-1}(v),v}, S_{m^{-1}(v)} + l_{m^{-1}(v),v}, \max_{w \in PR(v)}(S_w + l_{wv})\}, \tag{2}$$

where l_{uv} is the weight of the sequence arc between u and v. Within the batch-oblivious approach, the weight of the sequence arc between v and $m(v)$ is set to $l_{v,m(v)} = 0$ if v and $m(v)$ belong to the same batch. To model the stress test machines of problem (1), we do not have to guarantee $S_v = S_{m(v)}$ if $l_{v,m(v)} = 0$. Further, the weight of sequence arcs is set to 0 for all adjacent stress operations that belong to the same family, i.e. to the same batch. Hence, no batching decisions have to be made since the order of operations defines the batches for stress test machines. A sequence of $k \geq 1$ adjacent operations that belong to the same family is called a batching chain $v, m(v), \dots, m^k(v)$. The weight of the sequence arc of the last operation in the batching chain has to be set to the difference of the maximum completion time of all operations in the batching chain and the start time of the first operation of the batch. Since conditioning periods have to be considered, (2) only serves as an earliest start date E_v of the first operation of the batch. For the remaining operations in the batching chain, the machine predecessor solely contributes with its start date to the earliest start date of the respective node, i.e.

$$E_{m^q(v)} = \max\{S_{r^{-1}(m^q(v))} + l_{r^{-1}(m^q(v)),m^q(v)}, S_{m^{q-1}(v)}, \max_{\mathrm{w}\in\mathrm{PR}(\mathrm{m}^{\mathrm{q}}(\mathrm{v}))}(S_w + l_{w,m^q(v)})\} \tag{3}$$

for $q = 1, \dots, k$. The start and completion times of the operations in the batching chain then are computed with the Algorithm 2 for start time computation of stress test machines proposed by Hautz et al. (2024), with job order $\pi = (v, m(v), \dots, m^k(v))$, and earliest start dates $r = (E_v, \dots, E_{m^k(v)})$. Subsequently, the start and completion times of the operations in the batching chain are set, and the edge weights on the routing edges are updated to $l_{v,r} = C_v - S_v, \dots, l_{m^k(v),r^k} = C_{m^k(v)} - S_{m^k(v)}, r \in SR(v) \cup \{r(v)\}, r^k \in SR(m^k(v)) \cup \{r(m^k(v))\}$, to ensure job availability. In addition, the sequencing arc of the last operation in the batching chain is set to $l_{m^k(v),m(m^k(v))} = \max_{\mathrm{q}\in\{1,\dots,\mathrm{k}\}}(C_{m^q(v)}) - S_v$. Thus, we require the invariant

$$\left(l_{v,m(v)} = 0\right) \vee \left(l_{v,m(v)} = p_v + cond_{m_v} \wedge l_{m^{-1}(v),v} > 0\right) \vee \left(l_{v,m(v)} = \max_{\mathrm{j}\in\mathrm{B_v}}\left(C_j\right) - S_w \wedge l_{m^{-1}(v),v} = 0\right) \tag{4}$$

for all nodes $v \in V$ on stress machine routes, whereas B_v denotes the batch containing v and w denotes the first node in the batching chain containing v. In addition, the route arcs have to be updated in order to model the job availability, and the actual processing time needs to be computed for stress operations since it depends on the preemptions that occur within the batch. Consequently, the invariant

$$\left(\left(l_{m^{-1}(v),v} > 0 \wedge l_{v,m(v)} > 0\right) \vee l_{v,r} = C_v - S_v\right) \wedge \left(\left(l_{m^{-1}(v),v} = 0 \vee l_{v,m(v)} = 0\right) \vee l_{v,r} = p_v + cond_{m_v}\right) \tag{5}$$

is required for all $v \in V, r \in SR(v) \cup \{r(v)\}$ on stress test machine routes. If a stress operation is not in a batch, then conditioning has to be applied just once at the start of the operation. In this case, $l_{v,m(v)} = l_{v,r} = p_v + cond_{m_v}$ holds, which fulfills (4) and (5). Note that for $v \in V$ on tester routes, no batching is considered as the testers are modelled as regular machines, and the conditioning times of all tester machines are set to 0. Thus, $l_{v,m(v)} = l_{v,r} = p_v > 0$ holds for all $v \in V$ on tester routes, which also fulfills (4) and (5).

Figure 1 shows an example with three jobs and one batch consisting of four operations. For the sake of visibility, only the disjunctive arcs of the stress test machine containing the batch are shown, the artificial end nodes were omitted. The weights of the route and sequencing arcs fulfill (4) and (5). The solution obtained by Algorithm 2 from Hautz et al. (2024) for the batch is also depicted in the Gantt chart in the figure.

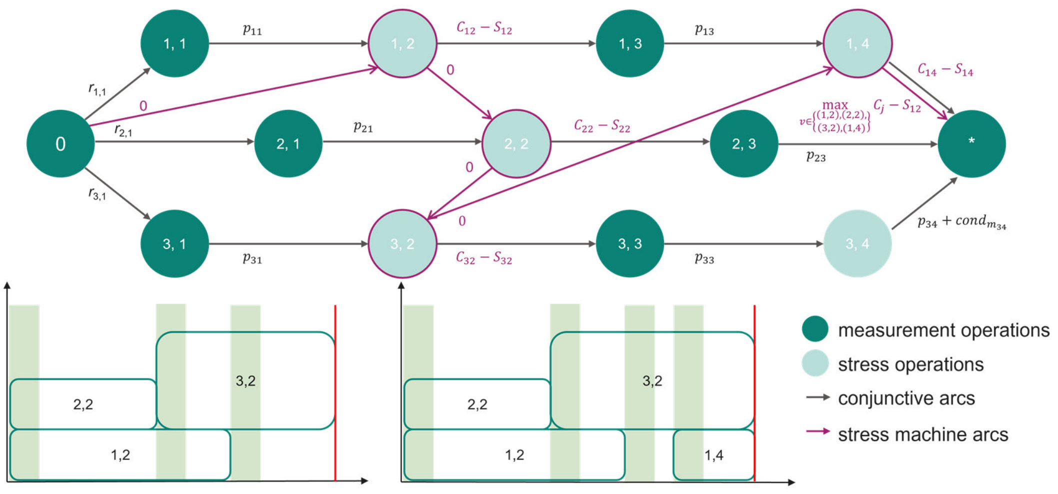

Figure 1: Example of a disjunctive graph for problem (1) consisting of jobs. The Gantt chart shows the solution obtained by Algorithm 2 from Hautz et al. (2024) for the formed batch.

3.1 Initial Solution

The presence of additional precedence constraints between two non-identical jobs could lead to cycle formation when searching for an initial solution. To address this problem, a list scheduling approach (LS) is proposed. The additional precedence constraints between operations of different jobs are modeled by adding conjunctive arcs to the graph. Let $G = (V, E_c \cup E_d)$, where E_c denotes the set of conjunctive arcs from the job sequences and additional precedence constraints, and E_d is the set of disjunctive arcs representing sequencing decisions. To avoid cycle formation, a careful machine assignment and iterative scheduling process is proposed. In the first step, operations are sorted according to their ready times and for each operation, machines are evaluated based on their current workloads, prioritizing machines that already contain operations of the same job family to enhance batch formation. Ties are broken by selecting machines with the smallest workload. The workload of the machines is computed as the sum of the ratio of processing times and job sizes of operations assigned to that machine. Once the machine assignment is completed, the operations are scheduled iteratively in a second step. A topological sorting σ of the graph G is used to determine the sequencing order, ensuring that all precedence constraints are respected. The calculation of the topological sorting can be done by a depth first search (DFS) for acyclic graphs. Initially, only the arcs in E_c are considered for calculating σ. After sequencing the first machine, additional disjunctive edges are introduced to the graph. These edges could influence the unscheduled machines. Hence, σ is updated after each iteration incorporating the current set E_d before scheduling the next machine.

3.2 Start Time Computation

Other than in the approach for p-batching in Knopp et al. (2017), the invariants (4) and (5) do not allow to compute start times within a single pass through the set of nodes sorted in topological order because all operations in the set B_v could possibly affect the actual processing time of v, i.e. the processing time of v plus the sum of conditioning times during preemptions. Furthermore, the possibility of reentrant process flows could make it impossible to evaluate a batching chain together in one iteration, as depicted in Figure 1. At least one recomputation is necessary since the earliest start time of o_{14} can only be evaluated after the start time of o_{13} was computed, that is again depending on o_{12}. Hence, the batch containing o_{12}, o_{22}, and

o_{32} has to be evaluated first, subsequently the start time of o_{13} and finally, the batch containing o_{12}, o_{22}, o_{32} and o_{14} is reevaluated. In addition, all nodes that were evaluated in between the batch nodes have to be reevaluated, since the incorporation of o_{14} could possibly change the actual processing time of all nodes in the batch. In the present example, o_{14} changes the completion time of o_{32} and hence, o_{33} is also affected.

Algorithms 1 and 2 consider the general case. Algorithm 1 computes a topological order σ of the $N = \sum_{j=1}^{n} n_j$ nodes and iterates over them, while storing all nodes that were already visited in the set $V_{visited}$. Measurement and stress operations are distinguished. For measurement operations, the start time is computed by computing the maximum completion time of machine and route predecessors, as stated in (2). Similarly, the earliest start time of stress operations are computed, with the difference that the completion time of the previous batch is computed if the node belongs to another job family than its machine predecessor. Otherwise, just the start time of the previous node of the batch is considered for the machine contribution, as described in (3). To compute the start time of stress operations, Algorithm 2 is recursively used to adjust the dynamic changes in processing times when preemptions occur. When a stress operation is reached in Algorithm 1, all nodes that were visited up to this iteration are given to Algorithm 2. If the current node is the first node of a batching chain (or possibly the only one), then the batch is just evaluated once and no adaptions have to be made. Otherwise, all nodes belonging to the batch B_v are collected and the earliest start time of v is computed. If $E_v \geq C_{batch}$, then no operations of the batch are preempted by the current node and hence, no adaptions have to be made. Otherwise, at least one operation of the batch is preempted, which could affect the start and completion times of all operations that were computed in between the first and the last operation of B_v. Hence, all those operations, except the nodes that are part of the batch, have to be visited again, whereas stress operations are recursively revisited since multiple batches could be affected.

```
procedure Algorithm1($G = (V, E)$)
    $V_{visited} \leftarrow \emptyset$
    for $j = 1, \ldots, N$ do
      $v = \sigma(j)$
      if $v$ is a measurement operation do
        $S_v = \max\{S_{r^{-1}(v)} + l_{r^{-1}(v),v}, S_{m^{-1}(v)} + l_{m^{-1}(v),v}, \max_{w \in PR(v)}(S_w + l_{wv})\}$
      else
        $B_{visited} \leftarrow \{v\}$, $S_v =$ Algorithm2($v, V_{visited}, B_{visited}$)
      end if
      $V_{visited} \leftarrow V_{visited} \cup \{v\}$
    end for
end procedure

procedure Algorithm2($v, V_{visited}, B_{visited}$)
    $B_v \leftarrow \{v\}$                    // the batch containing $v$
    $C_{batch} = -1$                     // the completion time of $B_v$ without $v$
    $w = m^{-1}(v)$
    if $w \neq 0$ then
```

```
        **while** $f_w == f_v$ **do**
            $B_v \leftarrow B_v \cup \{w\}$
            $B_{visited} \leftarrow B_{visited} \cup \{w\}$
              **if** $C_w > C_{batch}$ **then**
                  $C_{batch} = C_w$
              **end if**
              **if** $m^{-1}(w) \neq 0$ **then**
                  $w = m^{-1}(w)$
            **end if**
        **end while**
    **end if**
    $reverse(B_v)$
        **if** $m^{-1}(v) \neq 0$ and $f_{m^{-1}(v)} \neq f_v$ **then** //(*)
            $w = \mathrm{argmax}_{u \in B_{m^{-1}(v)}}\{C_u\}$
            $E_v = \max\{S_{r^{-1}(v)} + l_{r^{-1}(v),v}, S_w + l_{w,v}, \max_{u \in PR(v)}(S_u + l_{uv})\}$,
        **else**
            $E_v = \max\{S_{r^{-1}(v)} + l_{r^{-1}(v),v}, S_{m^{-1}(v)}, \max_{u \in PR(v)}(S_u + l_{uv})\}$,
        **end if** //(**)
    evaluate the batch $B_v$ with Algorithm 2 from Hautz et al. (2024) and set weights
    **if** $E_v < C_{batch}$ **then**
        **for** $w \in V_{visited}, pos(first(B_v)) < pos(w) < pos(last(B_v))$ **do**
            **if** $v$ is a measurement operation **do**
              $S_v = \max\{S_{r^{-1}(v)} + l_{r^{-1}(v),v}, S_{m^{-1}(v)} + l_{m^{-1}(v),v}, \max_{w \in PR(v)}(S_w + l_{w,v})\}$
            **else**
              $B_{visited} \leftarrow \{v\}$, $S_v =$ Algorithm2$(v, V_{visited}, B_{visited})$
            **end if**
            evaluate $E_v$ like in (*)-(**)
            evaluate batch with Algorithm 2 from Hautz et al. (2024) and set weights
        **end for**
    **end if**
    **return** $S_v$
```

end procedure.

4 SOLUTION APPROACHES

4.1 CP Approach

We present an extended version of the CP model established for the single-machine case in Hautz et al. (2024). For detailed explanations of the used constraint constructs, especially for synchronization, we refer to this paper. The model is based on the following decision variables, state- and cumulative-functions are used:

J_{jo}:	interval variable representing job $j \in J$ at operation $o \in O$, no size is specified
OJ_{jok}:	optional interval variable representing job $j \in J$ at operation $o \in O$ and machine $k \in M$, no size is specified
CS_{jok}, CE_{jok}:	optional interval variable for conditioning periods of job j (start and end) at stress operation $o \in O^S$and stress test machine $k \in M^S$, size is $cond_k$
C_{sk}:	optional interval variable representing the conditioning periods s on stress test machine $k \in M^S$, no size is specified
O_{jo}:	integer variable representing the overlaps in the conditioning period of job $j \in J$ at stress operation $o \in O^S$
C_{max}:	integer variable representing the makespan
S_k:	sequence variable representing job operation sequence on tester $k \in M^T$
$cumul\ function\ C_k$:	$= \sum_{j=1}^{n} \sum_{o=oj1,\ldots,o_{j,n_j}} (pulse(CS_{jok}, 1) + pulse(CE_{jok}, 1))\,, k \in M^S$
$cumul\ function\ S_k$:	$= \sum_{s=1}^{S} pulse(C_{sk}, 1)\,, k \in M^S$
$cumul\ function\ R_k$:	$= \sum_{j=1}^{n} \sum_{o=oj1,\ldots,o_{j,n_j}} pulse(OJ_{jok}, s_j)\,, k \in M^S$
$state\ function\ F_k$:	state function indicating the active job family on stress test machine $k \in M^S$.

The $CS_{jok}, CE_{jik}, O_{jo}, C_k, S_k$, and R_k quantities are only defined for stress operations and stress test machines, while S_k only models tester machines. The efficient differentiation of the index sets is necessary to limit the problem size. This is carried out via tuple modeling inside the commercial CP solver used in the experiments. O^S and O^T describe the sets of stress and measurement operations, and M^S and M^T the sets of stress test and tester machines, respectively. P is the set of precedence constraints. The objective is given by $\min C_{max}$, and the constraints are:

$$startAtStart(OJ_{jok}, CS_{jok}), startAtEnd(CE_{jok}, J_{jok}), \quad j \in J, o \in O^S \tag{6}$$

$$presenceOf(OJ_{jok}) = presenceOf(CS_{jok}) = presenceOf(CE_{jok}), \ j \in J, o \in O^S \tag{7}$$

$$alwaysIn(S_k, CS_{jk}, 1,1), \ alwaysIn(S\_k, CE_{jok}, 1,1), \ j \in J, o \in O^S, k \in M^S, \tag{8}$$

$$alwaysIn(C_k, C_{sk}, 1, B_k), \quad s \in \{1, \ldots, S\}, k \in M^S \tag{9}$$

$$O_{jo} = \sum_{s=1}^{S} overlapLength(OJ_{jok}, C_{sk}), \quad j \in J, o \in O^S, k \in M^S, \tag{10}$$

$$sizeOf(J_{jo}) = p_{jo} + O_{jo}, \ j \in J, o \in O^S, \tag{11}$$

$$presenceOf(C_{s2,k}) => presenceOf(C_{s1,k}), \tag{12}$$

$$endBeforeStart(C_{s1,k}, C_{s2,k}), \ s1, s2 = 1, \ldots, S, s1 + 1 = s2, k \in M^S. \tag{13}$$

$$alwaysEqual(F_k, OJ_{jk}, f_j, 0,0), j \in N, \ R_k \le B_k, \quad k \in M^S \tag{14}$$

$$noOverlap(J_{io}, all_k(OJ_{iok})), alternative(J_{io}, all_k(OJ_{iok})), j \in J, o \in O, k \in M, \quad (15)$$
$$endBeforeStart(J_{jo}, J_{jo+1}),\ j \in J, endBeforeStart(J_u, J_v),\ (u, v) \in P, \quad (16)$$
$$endOf\left(J_{jo_{j,n_j}}\right) \leq C_{max}, j \in J. \quad (17)$$

Constraints (6) – (14) model the stress test machines as described in in Hautz et al. (2024) in detail, the presented model extends them by the operations sequence o and parallel machine capability k dimension. To treat measurement machines, the conventional and more powerful global noOverlap constraints is used directly in (15). An implicit modeling of capacities via cumulative functions over synchronized jobs is not needed in this situation. The alternative constraint set in (15) models the selection of one machine out of set of alternatives in each operation. This is done in the same way for stress and measurement operations. Operation sequence constraints ($o \to o + 1$) and job-operation precedence constraints ($u \to v$) are modeled via (16). Finally, (17) bounds the C_{max} value.

4.2 VNS Approach

VNS is a local search-based metaheuristic that explores dynamically changing neighborhood structures to escape local optima and improve solution quality (Mladenovic and Hansen 1997). We propose a VNS scheme building on the foundation of basic VNS (Hansen and Mladenovic 2001) based on the established disjunctive graph formulation. The neighborhood function is based on a swap move, where two successive critical operations on the same machine are reversed. Critical operations are on the critical path, which is the sequence of operations that determine the makespan of the schedule, i.e. the longest path between the start operation 0 and the end operation $*$. This move is among the most fundamental neighborhood definitions for job-shop scheduling problems and is motivated by two properties shown by Van Laarhoven et al. (1992). Firstly, solutions obtained by such moves are always feasible. Secondly, the neighborhood is connected, ensuring that an optimal solution can be reached from any initial solution via a finite sequence of swap moves. However, since problem (1) is a flexible job shop, reaching the optimum is not guaranteed with a fixed initial machine assignment. With the integration of additional moves that allow machine reassignment like the integrated move of Dauzere-Peres and Paulli (1997), it is likely that this approach can be improved to a large extent.

Reversing a critical arc on an acyclic graph, called edge reversal, yields an acyclic graph (Van Laarhoven et al. 1992). We use similar arguments to show that reversing an arc in our disjunctive graph does not introduce cycles if the swap between two nodes that are included in an additional precedence constraint is prohibited. Let u and v be two nodes of different jobs with an additional precedence constraint $u \to v$. Then, all disjunctive arcs between the predecessors of u and v, between the predecessors of u and the successors of v, between u and v, and between u and the successors of v have to follow the direction of the precedence constraint. Otherwise, a cycle would be created since there is a path from u to v and therefore, there is a path from all predecessors of u to v and all successors of v. However, those arcs mentioned above can never be part of the critical path except for the arc between u and v. Assume that an edge between a predecessor p of u and v or a successor s of v is part of the critical path. Then, since there is an additional precedence constraint between u and v, the path $(p, \ldots, u, v, \ldots, s)$ is a longer path than $(p, v, \ldots, s)$, which is a contradiction. Analogously, an edge between u and a successor s of v can never be part of the critical path. It is shown by Dauzère-Pérès et. al. (1998), that an integrated move is applicable for multi-resource job shop scheduling problems with nonlinear routings and resource flexibility. Furthermore, Knopp et al. (2017) demonstrate that the integrated move is also applicable for p-batching. Since the neighborhood structure of edge reversals is included in the neighborhood structure of integrated moves (Dauzère-Pérès and Paulli 1997), the applicability of edge reversals is guaranteed. The conditions in their theorems prohibit the case where two critical operations connected by an additional precedence constraint are reversed.

We propose a VNS scheme that utilizes the edge reversal move to define the neighborhood structures required for the shaking phase, whereas each neighborhood structure applies the move l times up to $l = 10$, i.e., we consider the ten neighborhood structures N_l defined in this way. For small to medium-sized instances, the specific choice of l does not significantly affect the results, but for larger instances, a higher value for l promised better results in preliminary experiments. In the local search phase, the critical paths to randomly chosen artificial end nodes are computed and edge reversals are applied to those paths, continuing as long as the objective function value can be improved.

5 COMPUTATIONAL EXPERIMENTS

5.1 Design of Experiments and Implementation Issues

We assess the performance of the VNS scheme and CP model by randomly generated instances motivated by settings in reliability laboratories. We consider small-, medium-, and large-sized instances with a different number of operations, number of machines, and number of incompatible job families. As a reference, we compare the results to the solutions obtained by the LS approach. In the following, $DU[a,b]$ refers to a discrete uniform distribution over the set of integers $\{a,\dots,b\}$. The number of jobs for the small instances is set to $n \in \{5,10\}$ with $n_j \sim DU[1,10]$. We consider $m \in \{3,5\}$ machines, whereas we distinguish between the amount of stress test machines m_s and of testers m_t, and $f_{max} \in \{2,3\}$ job families. For the medium-sized instances, we set $n \in \{24,36,48\}, n_j \sim DU[1,15], m \in \{8,12\}, f_{max} \in \{10,15\}$, and for the large-sized instances, we consider $j = 96,\ n_j \sim DU[1,20], m \in \{15,20\}$, and $f_{max} = 15$. Within all instances, the processing times for stress and measurement operations are distributed according to $DU[1,100]$ and $DU[1,20]$, respectively. The ready times follow $r_j \sim DU[1, \lceil 50N/B \rceil]$, and the job sizes are distributed according to $s_j \sim DU[1,13]$. The maximum batch size for stress test machines is taken from $\{21,42\}$ and the conditioning from $\{1,2,3\}$, whereas each value has the same probability. We use a maximum computing time per instance for the CP and the VNS scheme and minimize the makespan.

The VNS scheme is coded using the C++ programming language, for the disjunctive graph model the Boost library is applied. IBM ILOG CPLEX Optimization Studio version 22.1 CP Optimizer with OPL Language and CP default settings is applied. The computational experiments are carried out on a workstation with Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz.

5.2 Results

The obtained C_{max} values are shown in Table 1. The time limit for the CP and VNS approach is set to 60, 120, and 300s per instance. The LS is performed without any time limit and serves as a reference solution. The known optimal C_{max} values that are obtained by the CP are marked bold. For small-sized instances, both the CP and VNS perform very well, providing near-optimal or optimal solutions within 60s. The performance of the approaches is nearly identical in this category. For medium-sized instances, the CP approach slightly outperforms the VNS when given 300s of computing time per instance. Furthermore, the CP approach shows a clear advantage for shorter time limits, where it consistently provides better solutions than the VNS. The VNS benefits from extended computing times, as its performance improves significantly between the 120 and 300s marks, whereas the CP approach reaches high-quality solutions after 120s, with minimal improvements in comparison to 300s. For large-sized instances, the CP model struggles to find a feasible solution within the given computing time. While the VNS approach is always able to produce feasible solutions, it also shows slower improvements over time for those instances. This suggests that evaluating neighbors becomes increasingly time-consuming for larger instances. The majority of computing time is spent evaluating neighbors, indicating that more efficient approaches for evaluating solutions should be explored.

6 CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

Complex job-shop scheduling problems motivated by settings in reliability laboratories were studied. The batch-oblivious approach for disjunctive graphs was extended towards allowing the modeling of stress test machines found in reliability laboratories. A VNS scheme was designed based on the disjunctive graph model. It was compared with a CP formulation. The CP approach is able to outperform the VNS scheme for medium-sized instances, but it failed to compute a feasible solution for large-sized problem instance given a limited amount of computing time.

Table 1: Computational results for the C_{max} measure.

Benchmark				**LS**	**CP**	**VNS**	**CP**	**VNS**	**CP**	**VNS**
n	m_s	m_t	f_{max}		**(60s)**	**(60s)**	**(120s)**	**(120s)**	**(300s)**	**(300s)**
5	1	2	2	569	452	465	451	451	451	451
5	1	2	3	821	669	676	**667**	**667**	**667**	**667**
5	2	3	2	563	508	508	508	508	508	508
5	2	3	3	439	**374**	**374**	**374**	**374**	**374**	**374**
10	2	3	2	764	544	588	544	549	542	549
10	2	3	3	686	545	627	545	608	545	545
10	3	2	2	795	538	550	538	543	538	523
10	3	2	3	634	330	445	330	409	329	384
24	4	4	10	1819	1056	1433	1054	1228	1052	1083
24	4	4	15	1770	1050	1517	1010	1256	1004	1055
24	7	5	10	1382	616	944	616	832	616	650
24	7	5	15	1438	661	960	654	953	653	710
36	4	4	10	3060	-	2646	2147	2123	1211	1324
36	4	4	15	3024	1325	2750	1234	2208	1232	1331
36	7	5	10	1940	1257	1723	898	1542	892	960
36	7	5	15	1989	892	1517	897	1384	888	951
48	4	4	10	3438	1615	2967	1467	2416	1459	1615
48	4	4	15	3400	-	2814	1774	2569	1770	1820
48	7	5	10	3010	1183	2678	1018	2257	1005	1183
48	7	5	15	2754	-	2337	1166	2181	1162	1271
96	7	8	15	7565	-	7337	-	6962	-	6833
96	10	5	15	5820	-	5804	-	5679	-	5307
96	10	10	15	5637	-	5319	-	5253	-	4613
96	14	6	15	6574	-	6473	-	6323	-	6045

There are several directions for future research. First of all, we believe that moves similar to the integrated move of Dauzere-Peres and Paulli (1997) will considerably improve the performs of the VNS scheme. The CP approach can be improved by starting from an initial solution and designing decomposition methods. Moreover, the real-world scheduling problem found in reliability laboratories is more difficult, including s-batching machines, maximal time lags, and on-time delivery-related performance measures. In future research, we will tackle these problems. In addition to static settings, the proposed algorithms should be applied in a rolling horizon setting similar to Mönch and Zimmermann (2011) taking into account process uncertainty.

ACKNOWLEDGMENTS

This work was funded by the Austrian Research Promotion Agency (FFG, Project No. FO999900268).

REFERENCES

Azem, S., R. Aggoune, and S. Dauzère-Pérès. 2012. "Heuristics for Job Shop Scheduling with Limited Machine Availability". *IFAC Proceedings Volumes* 45(6):1395–1400.

Dauzère-Pérès, S., J. Ding, L. Shen, and K. Tamssaouet. 2024. "The Flexible Job Shop Scheduling Problem: A Review". *European Journal of Operational Research* 314:409–432.

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* 70:282–306.

Dauzère-Pérès, S., W. Roux, and J. B. Lasserre. 1998. "Multi-resource Shop Scheduling with Resource Flexibility". *European Journal of Operational Research* 107(2): 289–305.

El-Kareh, B., and L. N. Hutter. 2020. *Silicon Analog Components: Device Design, Process Integration, Characterization, and Reliability*. 2nd ed., Cham: Springer.

Fowler, J. W., and L. Mönch. 2022. "A Survey of Scheduling with Parallel Batch (p-Batch) Processing". *European Journal of Operational Research* 298(1):1–24.

Hansen, P., and N. Mladenovic. 2001. "Variable Neighborhood Search: Principles and Applications". *European Journal of Operational Research* 130: 449–467.

Hautz, J., A. Klemmt, and L. Mönch. 2024. "Scheduling Jobs on a Single Stress Test Machine in a Reliability Laboratory". In *2024 Winter Simulation Conference (WSC)*, 1797-1808 https://doi.org/10.1109/WSC63780.2024.10838938.

Hautz, J., A. Klemmt, and L. Mönch. 2025. "Exact and Heuristic Approaches for Scheduling Parallel Stress Test Machines in Semiconductor Reliability Laboratories". *Flexible Services and Manufacturing Journal*, in print.

Knopp, S., S. Dauzère-Pérès, and C. Yugma. 2017. "A Batch-oblivious Approach for Complex Job-shop Scheduling Problems". *European Journal of Operational Research* 263(1):50–61.

Lin, S.-W., and K.-C. Ying. 2020. "Minimising Makespan in Job-shops with Deterministic Machine Availability Constraints". *International Journal of Production Research* 59(1):4403–4415.

Mauguière, P., J. C. Billaut, and J. L. Bouquard. 2005. "New Single Machine and Job-Shop Scheduling Problems with Availability Constraints". *Journal of Scheduling* 8:211–231.

Mladenovic, N., and P. Hansen. 1997. "Variable Neighborhood Search". *Computers & Operations Research* 24:1097–1100.

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.

Mönch, L., J. W. Fowler, and S. J. Mason. 2013. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*. New York: Springer.

Mönch, L., and J. Zimmermann. 2011. "A Computational Study of a Shifting Bottleneck Heuristic for Multi-Product Complex Job Shops". *Production Planning & Control* 22(1):25–40.

Rocholl, J., and L. Mönch. 2025. "Metaheuristics for Solving Flexible Flow-shop Scheduling Problems with s-Batching Machines". *International Transactions in Operational Research* 32(1):38–68.

Tamssaouet, K., S. Dauzère-Pérès, and C. Yugma. 2018. "Metaheuristics for the Job-shop Scheduling Problem with Machine Availability Constraints". *Computers & Industrial Engineering* 125:1–8.

Van Laarhoven, P. J., E. H. Aarts, and J. K. Lenstra. 1992. "Job Shop Scheduling by Simulated Annealing". *Operations Research* 40(1):113–125.

AUTHOR BIOGRAPHIES

JESSICA HAUTZ is a PhD student at the University of Hagen. She received her master's degree in Mathematics in 2022 from the University of Klagenfurt. She works as a PhD researcher at KAI GmbH in the data science team. Her research interests are combinatorial optimization, mathematical programming, and scheduling. Her email adress is Jessica.Hautz@k-ai.at.

ANDREAS KLEMMT is a Lead Principal Engineer at Infineon Technologies. He received his master's degree in Mathematics in 2005 and Ph.D. in Electrical Engineering in 2011 from Dresden University of Technology. He works as vertical integration solution architect in the Global Factory Integration Department of Infineon. His research interests are scheduling, mathematical programming, capacity planning, production control, and simulation. His email adress is Andreas.Klemmt@infineon.com.

LARS MÖNCH is full professor of Computer Science at the Department of Mathematics and Computer Science, University of Hagen where he heads the Chair of Enterprise-wide Software Systems. He holds M.S. and Ph.D. degrees in Mathematics from the University of Göttingen, Germany. After his Ph.D., he obtained a habilitation degree in Information Systems from Technical University of Ilmenau, Germany. His research and teaching interests are in information systems for production and logistics, simulation, scheduling, and production planning. His email address is Lars.Moench@fernuni-hagen.de. His website is https://www.fernuni-hagen.de/ess/team/lars.moench.shtml.