



University of Hagen
Department of Mathematics and Computer Science
Chair of Enterprise-wide Software Systems



A Matheuristic for a Class of Scheduling Problems for Parallel Batch Processing Machines

Lars Mönch
LAAS-CNRS
June 2018

Introduction

- ▷ University of Hagen: largest distance learning university in Germany, Austria, and Switzerland
- ▷ 75,000 students

- ▷ Chair of Enterprise-wide Software Systems
 - <http://ess.fernuni-hagen.de>
 - Main research interests:
 - Information Systems for Manufacturing/Logistics
 - Production Planning/Scheduling
 - Simulation
 - Multi-Agent Systems
 - Research funding
 - current grants from
 - Infineon Technologies AG
 - European Commission/German government

Introduction (cont'd)

- ▷ Department of Mathematics and Computer Science



Motivation

- ▷ Electronics industry is one of the major industries
 - ▷ Manufacturing of integrated circuits is one of the key aspects of this industry
 - ▷ Scheduling problems attracted researchers and people from industry for the last three decades
 - ▷ Causes: high degree of automation even 30 years ago
 - automated real-time data collection
 - manual production control leads often to unexpected behavior of the system
 - ▷ Increasing automation pressure by automated material handling systems (AMHS) and new requirements on production control (Industry 4.0)
- => Scheduling approaches are both promising and necessary in the semiconductor domain
- ▷ Current state of the art in this industry: **sophisticated dispatching rule-based systems, MILP-based approaches for critical machine groups**



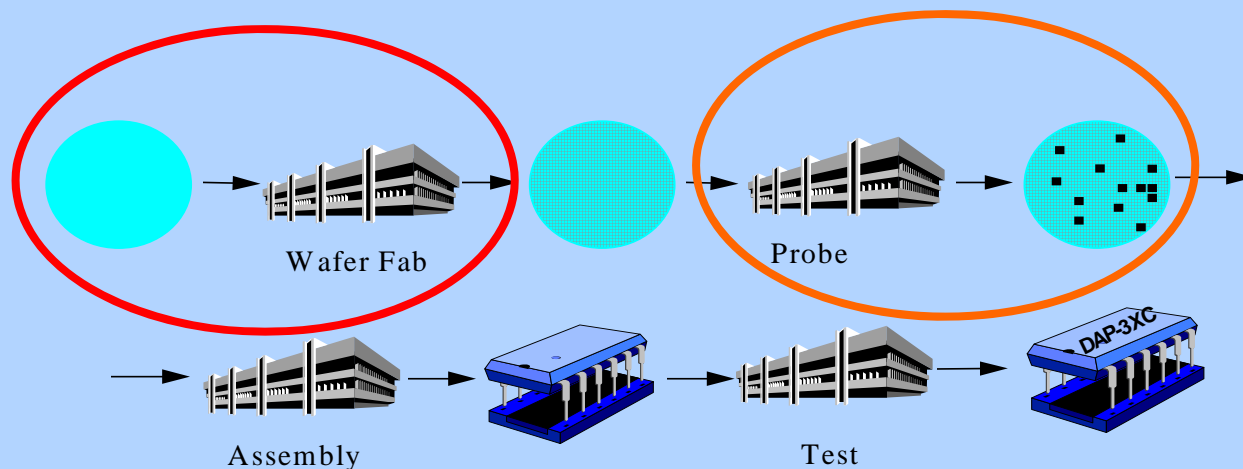
Outline

- ▷ **Process Description**
- ▷ **Scheduling Problems and Solution Techniques**
- ▷ **A Matheuristic Framework for Batch Machines**
- ▷ **Conclusions and Future Challenges**

Process Description

- ▷ semiconductor chip = highly miniaturized, integrated electronic circuit consisting of thousands of components
- ▷ raw wafers = thin discs made of silicon or gallium arsenide
- ▷ Up to thousand identical chips can be made on each wafer.
 1. The circuits are built up layer by layer in a **Wafer fab**.
 2. Then the wafers are sent to **Probe**, where electrical tests identify any die that is not good when packaged.

Front-end = Wafer fab + Probe

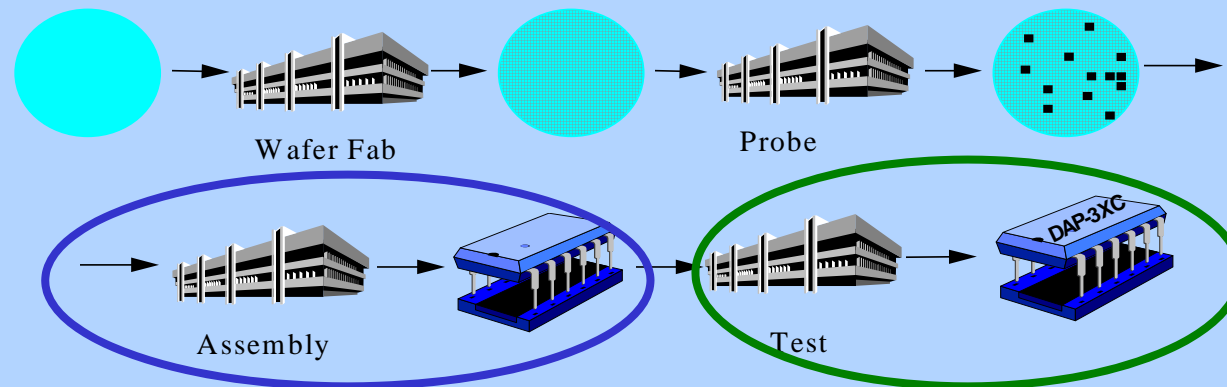


Process Description (cont'd)

3. Probed wafers are sent to **Assembly** where good dies are put into a package.
4. Packaged dies are sent to **Test** where they are tested before they are going to customers.

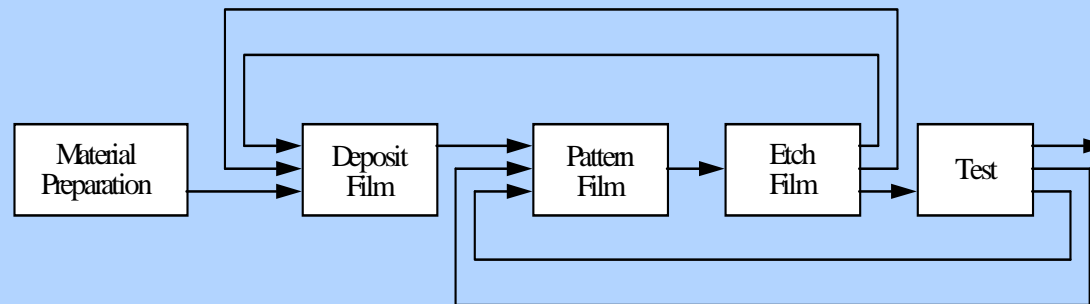
Back-end = Assembly + Test

- ▷ In this talk => We consider mainly the wafer fab part of the manufacturing process.



Process Description (cont'd)

- ▷ many process flows
- ▷ each process flow contains up to 800 operations
- ▷ several hundred (often very expensive) machines
- ▷ unrelated parallel machines
- ▷ re-entrant flows



- ▷ different processing times
- ▷ **long operations often involve batch processes (one third of all operations might be batch operations)**
- ▷ **non-linear flow because of mixture of batching and non-batching machines => long queues in front of the machines**
- ▷ (nested) time constraints for the processing of jobs to prevent native oxidation and contamination effects on the wafer surface

Process Description (cont'd)

- ▷ probabilistic occurrence of long machine failures leads to a large variability
- ▷ some machines like implanters require significant sequence-dependent setup times
- ▷ dynamic bottlenecks depending on the product mix
- ▷ some processing steps require auxiliary resources such as reticle in photolithography
- ▷ wafers are transported in **Front Opening Unified Pods (FOUP)** using an AMHS => manual handling of the wafers is not desirable



⇒ **manufacturing environment that is different in several ways from traditional flow and job shops**



Scheduling Problems & Solution Techniques

- ▷ **Batch** = group of jobs that have to be processed together
- ▷ **Completion time of a batch** = completion time of the last job of the batch

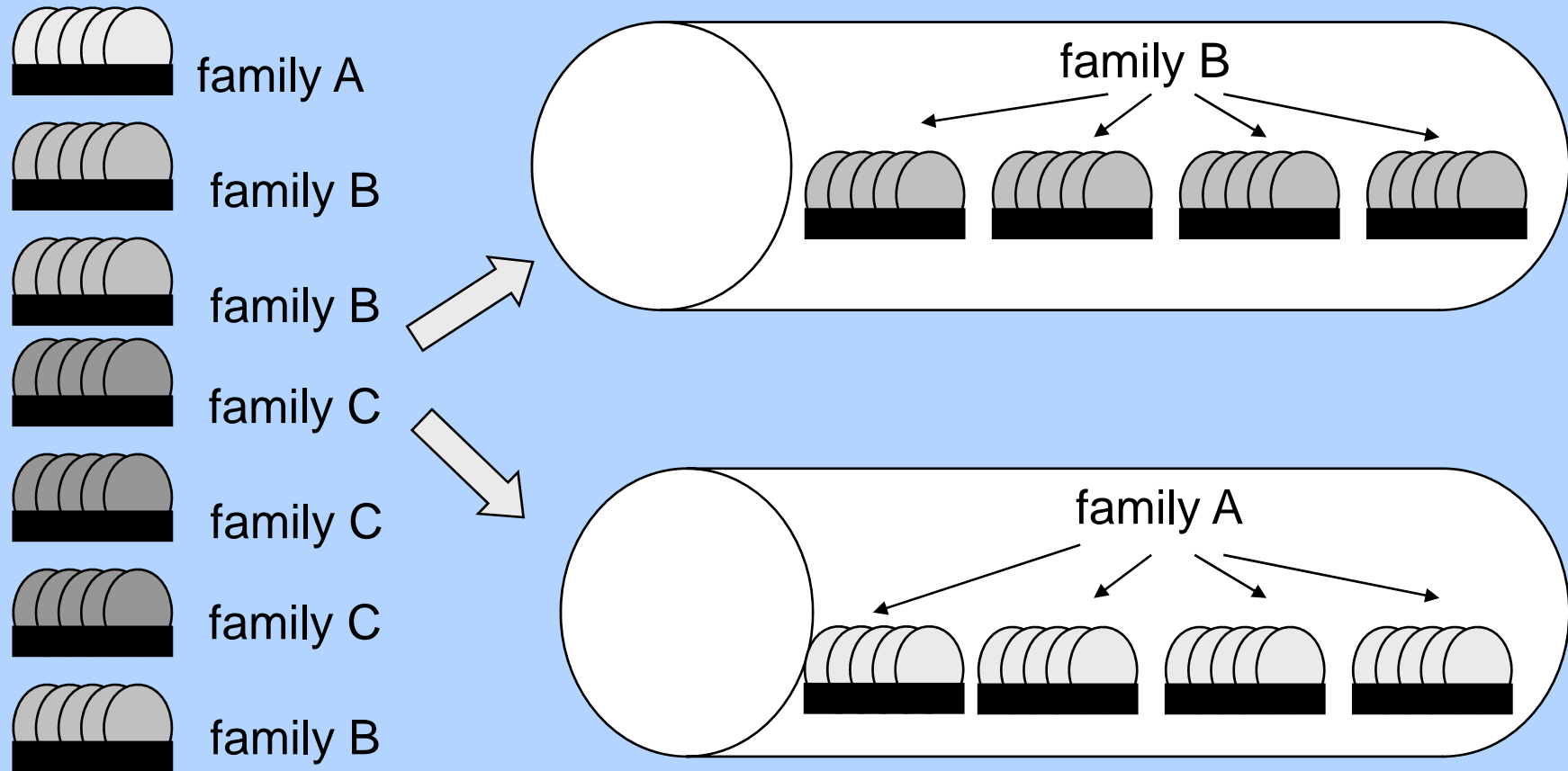
- ▷ Two types of batching problems (with batch availability):
 - **s-batching**: processing time of the batch is the **sum** of the processing time of the jobs
 - **p-batching**: processing time of the batch is the maximum processing time of the jobs that form the batch
- ▷ p-batching is very important in semiconductor manufacturing.

- ▷ Assumption of a fixed batch size B
 - burn-in ovens are used to heat-stress test chips
 - diffusion furnaces => incompatible job families, only jobs of one job family can be batched together due to the chemical nature of the process

- ▷ Sometimes secondary resources are typical for batching machines (load boards, reticles)

Scheduling Problems & Solution Techniques (cont'd)

Example: three incompatible families, $B=4$



Scheduling Problems & Solution Techniques (cont'd)

- ▷ Three decisions:
 - **batch formation**
 - assignment of batches to machines
 - sequencing of the batches

- ▷ Solution techniques:
 - dynamic programming
 - genetic algorithms
 - simulated annealing
 - taboo search
 - variable neighborhood search
 - ant colony optimization
 - **MILP/CP for subproblems**

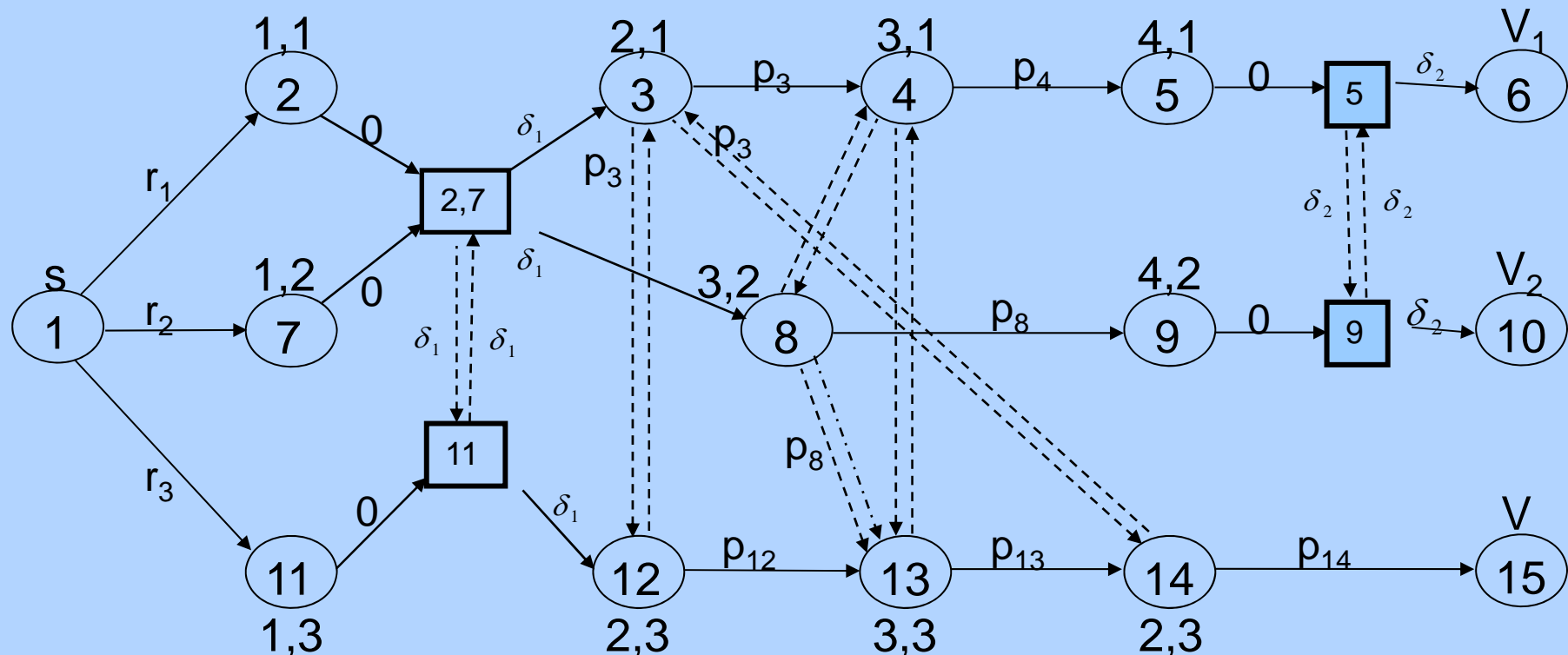
- ▷ Inclusion of ready times make batching problems much harder.
- ▷ $P_m \mid p\text{-batch, incompatible, } r_j \mid \sum w_j T_j$

Scheduling Problems & Solution Techniques (cont'd)

- ▷ Wafer fabs can be modeled as complex job shops (Ovacik & Uzsoy, 1997):
 - Unrelated parallel machines with sequence-dependent setup times
 - Parallel batch machines, re-entrant flows
 - Ready times of the jobs
- ▷ $FJm | r_j, p - \text{batch, incompatible}, s_j, \text{recrc} | \sum w_j T_j$
- ▷ $FJm | r_j, p - \text{batch, incompatible}, s_j, \text{recrc} | L_{\max}$
- ▷ **Disjunctive graph formulations** to solve these problems => shifting bottleneck heuristic or neighborhood search
- ▷ Large-scale job shops: several hundred machines, several hundred jobs, diverse production mix
- ▷ Interaction of job scheduling and AMHS
- ▷ Attempts to use constraint programming

Scheduling Problems & Solution Techniques (cont'd)

- ▷ Appropriate sub problem formulation and solution techniques for the sub problems
- ▷ Measures for critical machine group
- ▷ Implementation of the solution of the most critical sub problem
- ▷ Simulation-based performance assessment because of rolling horizon setting and re-scheduling

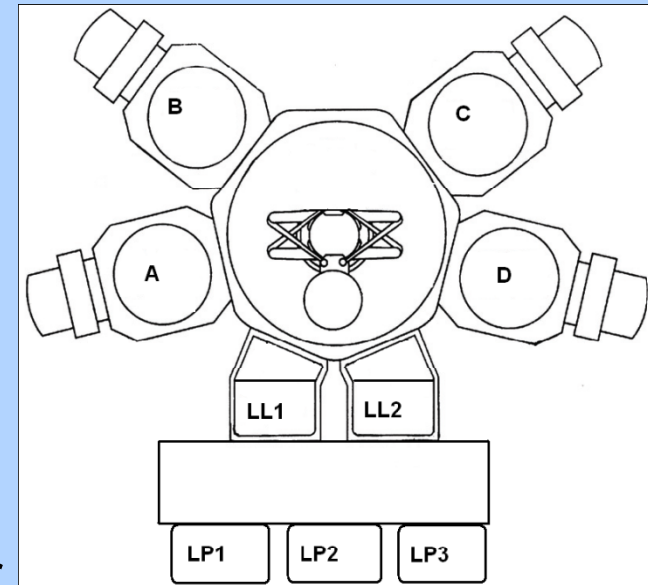


Scheduling Problems & Solution Techniques (cont'd)

▷ **Cluster tools** combine single wafer processing with wafer handling robots in one closed environment.

▷ **Example**

- Three load ports LP1,LP2,LP3 where lots are deposited (three lots)
- Wafers are transported to the two load locks LL1, LL2 (two wafers)
- Wafers are transferred to one of the four processing chambers A,B,C,D by a two-armed robot

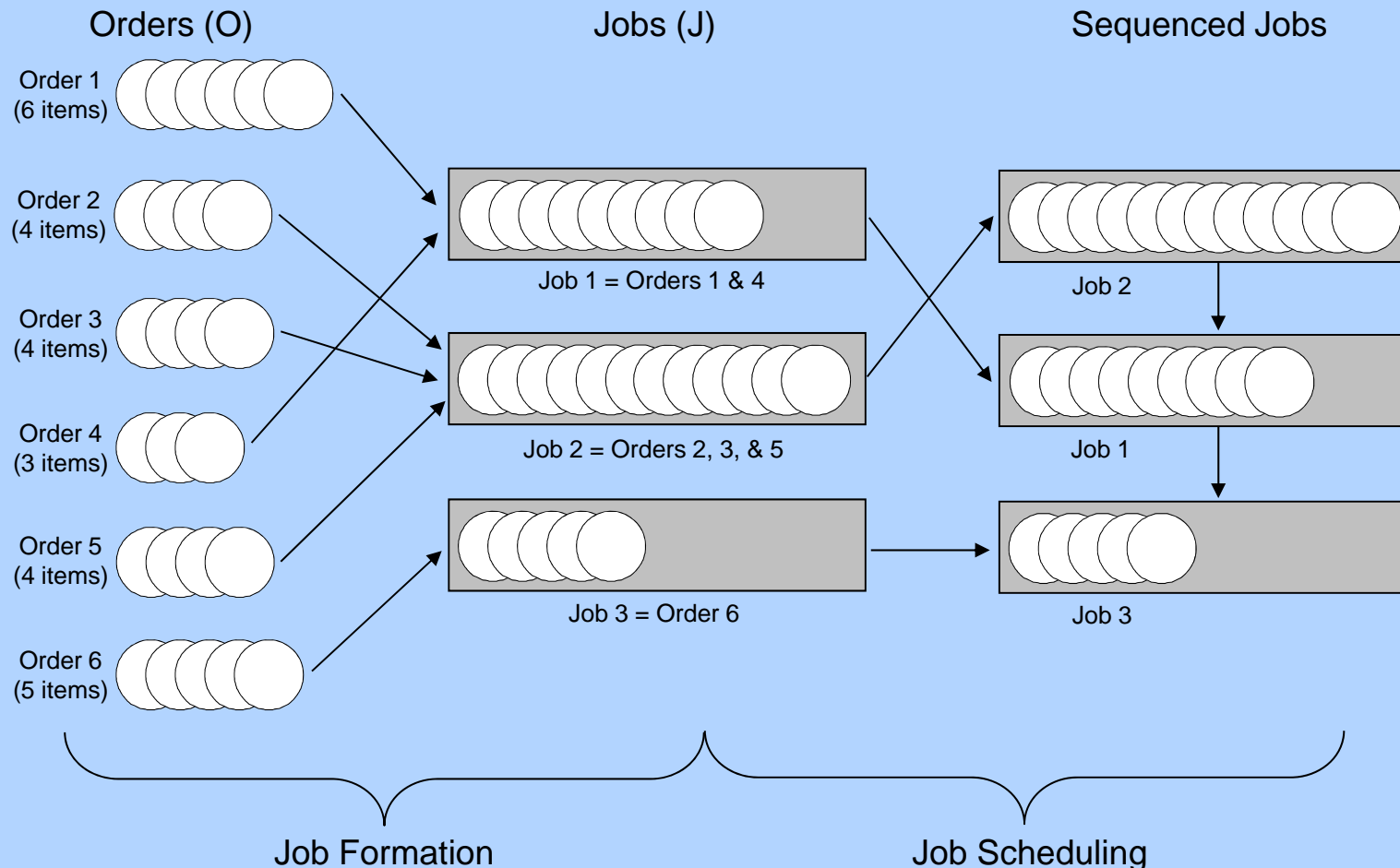


▷ **Scheduling of cluster tools** is challenging because the cycle time of wafers in a cluster tool depends on the used wafer recipes, cluster tool control and architecture, wafer waiting times, and sequencing

▷ **Methods:** discrete event simulation, neural networks are used to estimate the cycle time, beam search or metaheuristics for lot assignment and sequencing (for external scheduling), combination of Petri nets and integer programming (for internal scheduling)

Scheduling Problems & Solution Techniques (cont'd)

▶ multiple orders per job problems



▶ single machine, parallel machine problems + specific flow shops are studied using dispatching rules, metaheuristics, column generation

Scheduling Problems & Solution Techniques (cont'd)

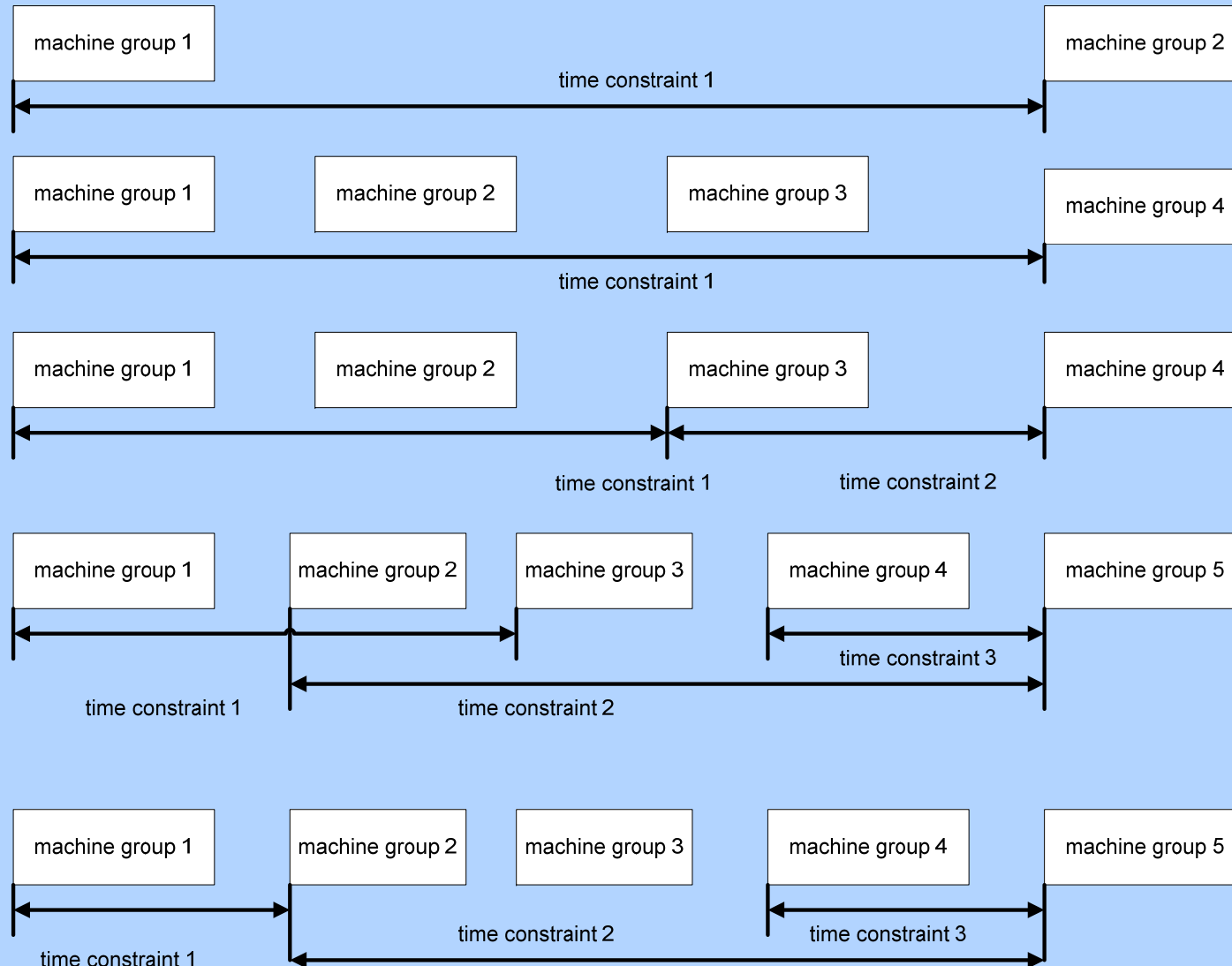
- ▷ **Definition:** We consider job J_i from product j . A time constraint holds for the two process steps O_{jr} and O_{js} if we have for the corresponding start times:

$$s_{is} \leq s_{ir} + \sum_{k=r}^{s-1} p_{jk} + t_{jrs}.$$

- ▷ Scheduling literature:
- only the very special case for $r = s - 1$ for positive time lag
 - negative time lag: modeling of transportation times
- ▷ Different classes of time constraints in semiconductor manufacturing due to possible contamination of the wafer surface, for instance between wet etch and furnace operations.
- ▷ Nested time constraints in contrast to the scheduling literature
- ▷ Solution techniques:
- Based on a job-based decomposition with MIP/CP subproblem
 - Embedded into a list-scheduling or GA-based approach to propose job sequences

Scheduling Problems & Solution Techniques (cont'd)

▷ classes of time constraints found in wafer fabs



A Matheuristic Framework for Batch Machines

- ▷ **Scheduling problem:**

$$P_m / p - \text{batch, incompatible} / \sum_{j=1}^n f_j(c_j) \quad (1)$$

- ▷ f regular sum objective function
- ▷ Incompatible families: all jobs of the same family have a common processing time

- ▷ **Property:**

There exists an optimal schedule for each instance of problem (1) that does not contain partially full batches except possibly the last batch of each family to be processed in the schedule.

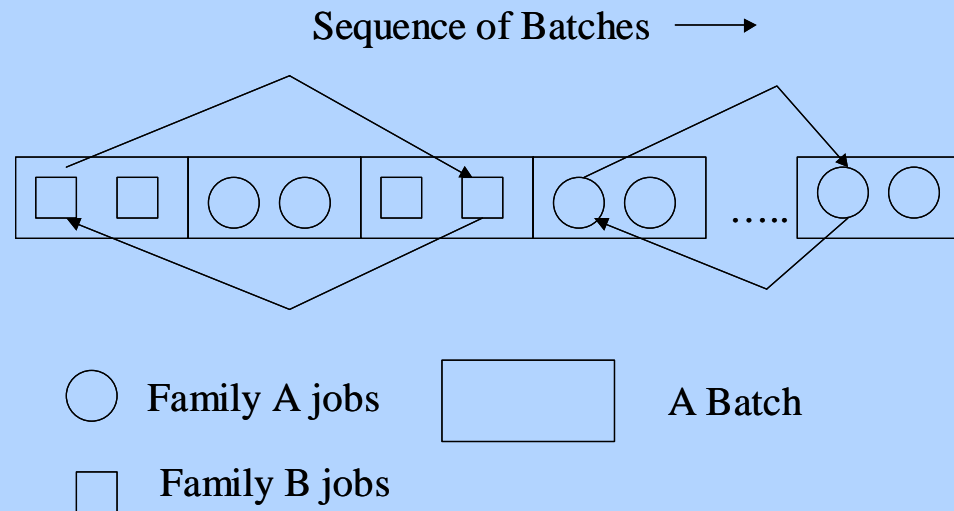
- ▷ The number of batches for each family can be precomputed.

A Matheuristic Framework for Batch Machines (cont'd)

1. How to form batches?
 2. How to assign the batches to one of the parallel machines?
 3. How to sequence the batches on each of the parallel machines?
- ▷ Swap is used in Balasubramanian et al. (2004), Almeder and Mönch (2011), Lausch and Mönch (2016) for correcting the content of batches
- ▷ **Swap-Iter:**
1. Start with the first job of the batch with the smallest start time.
 2. Try to interchange the current job with each job that is found in a batch of the same family that starts later.
 3. If an objective function value reduction occurs because of the job swap, exchange the two jobs and start over with the first job of the batch.

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ Works for single and parallel machines, only starting times of the batches of a given family are of interest



- ▷ **Observation:** If starting time of batches are known, the cost (objective function value) for assigning a job to any batch can be calculated.

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ Swap can be modeled as the following assignment-type problem for each incompatible family f :

$$\min \sum_{b=1}^{b_f} \sum_{j=1}^{n_f} f_j(C_b) x_{jb}$$

subject to

$$\sum_{b=1}^{b_f} x_{jb} = 1, \quad j = 1, \dots, n_f$$

$$\sum_{j=1}^{n_f} x_{jb} = B_b, \quad b = 1, \dots, b_f$$

$$x_{jb} \in \{0, 1\}, \quad j = 1, \dots, n_f, b = 1, \dots, b_f$$

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ **Observation 1:** Since the coefficient matrix is unimodular, the IP is easy to solve by the Simplex algorithm.

- ▷ **Observation 2:** IP can be reformulated as a minimum cost flow (MCF) problem =>
more efficient solution methods compared to the conventional Simplex algorithms are available in solver software, i.e., the network simplex algorithm or the out-of-kilter algorithm

=> **Swap-MCF**

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ It is enough to assign and sequence “empty” batches
- ▷ Computing the optimal content of the batches is “easy” task

- ▷ **Matheuristic:** biased random-key genetic algorithm (BRKGA) to assign and sequence batches
- ▷ Representation: $RK = [rk_1, rk_2, \dots, r_b]$
- ▷ Random keys
 - integer part: determines machine
 - fractional part: determines sequence

- ▷ Uniform crossover
- ▷ Immigration strategy for mutation

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ Applying Swap-MCF for each chromosome time-consuming
- ▷ Reference heuristic I based on list scheduling
- ▷ Heuristic H to compute the content of batches for each chromosome, followed by a workload balancing procedure

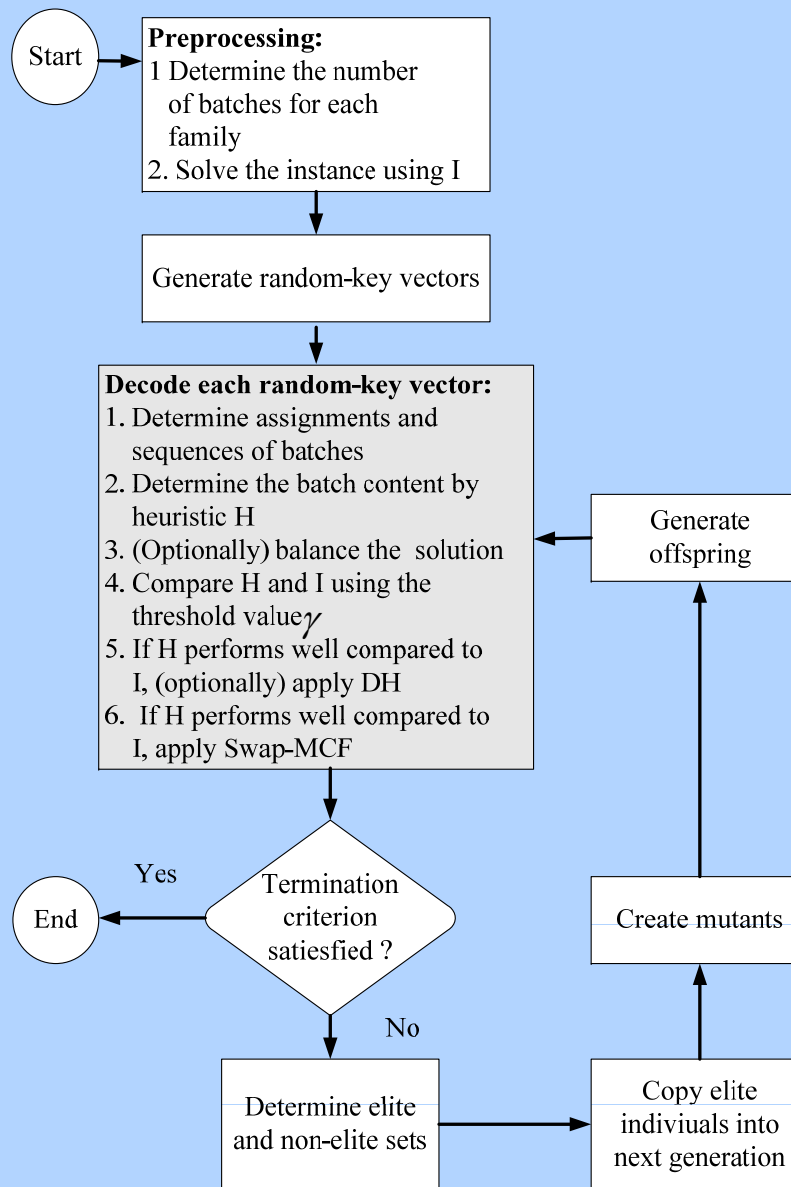
- ▷ Consider threshold value $\gamma > 0$:

$$\sum_{j=1}^n f_j(C_j(H)) \leq \gamma \sum_{j=1}^n f_j(C_j(I)) \quad (2)$$

- ▷ If (2) holds \Rightarrow improve the schedule by
 - (optionally) a decomposition heuristic (DH) for each single machine
 - Swap-MCF

A Matheuristic Framework for Batch Machines (cont'd)

▷ Overall algorithm:



A Matheuristic Framework for Batch Machines (cont'd)

▷ Application of the framework to:

$$1 / p\text{-batch, incompatible} / \sum_{j=1}^n w_j U_j \quad (3)$$

$$P_m / p\text{-batch, incompatible} / \sum_{j=1}^n w_j T_j \quad (4)$$

▷ (3): Dauzere-Peres & Mönch (2013):

- specific MILP formulations
- RKGA based on job sequences

▷ (4): Almeder & Mönch (2011)/Lausch & Mönch (2016):

- ACO
- VNS
- LNS

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ (3)
 - I: list scheduling based on weighted modified due date (WMDD) rule
 - H: modification of this approach for given batch set
- ▷ (4)
 - I: list scheduling based on apparent tardiness cost (ATC) rule
 - H: modification of this approach for given batch set
- ▷ Implementation issues:
 - brkga-API framework, C++ programming language
 - Ip_solve 5.5

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ 144 small-sized and 144 large-sized instances from Dauzere-Peres & Mönch (2013)
- ▷ Up to 144 jobs, up to 6 families, up to 90% tardy jobs
- ▷ Computational results (60 sec computing time per instance):
 - 72 small-sized unweighted instances: optimal solutions known from Dauzere-Peres & Mönch (2013) are found
 - 72 small-sized weighted instances: only 0.97% worst compared to the best performing MILP approach from Dauzere-Peres & Mönch (2013)
 - 72 large-sized unweighted instances: 0.31%
 - 72 large-sized weighted instances: 1.82%

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ 288 problem instances from Almeder & Mönch (2011)/ Lausch and Mönch (2016)
- ▷ Up to 300 jobs, up to 12 families, up to 60% tardy jobs
- ▷ different schedules:
 - **low-quality** solutions: ATC-BATC rule with randomly chosen look-ahead parameter
 - **medium-quality** solutions: ATC-BATC rule with appropriate look-ahead parameter
 - **fairly high-quality** solutions: ATC-BATC rule with appropriate look-ahead parameter and decomposition heuristic (DH) to re-sequence the already formed batches

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ Average deviation from results obtained by VNS/LNS is 1.33%

Factor Level	Deviation (in %)
Number of families F	
3	0.56
6	1.30
12	2.13
Maximum Batch Size B	
4	1.61
8	1.05
Number of machines m	
4	0.99
5	1.18
6	1.47
Number of Jobs n	
180	0.72
240	1.46
300	1.82
R, T combinations	
(0.5, 0.3)	1.90
(0.5, 0.6)	0.72
(2.5, 0.3)	0.91
(2.5, 0.6)	1.80
Overall	1.33

A Matheuristic Framework for Batch Machines (cont'd)

- ▷ Solution quality of Swap-Iter: percentage of optimal solutions

Low-quality Schedules	Medium-quality Schedules	High-quality Schedules
16.94%	36.39%	36.18%

- ▷ TWT ratios are 98% for low-quality and 99% for high-quality schedules
- ▷ Swap-MCF requires only 70% computing time of Swap-Iter

Conclusions and Future Challenges

- ▷ Wafer fab environments differ in many ways from traditional flow shop and job shop environment.
- ▷ Scheduling problems in semiconductor manufacturing are challenging with respect to process restrictions and problem sizes.

- ▷ **Some progress over the last three decades but many old challenges are still lasting.**
 - Common benchmarks are desirable like for traditional job shops
 - More theoretical insights are necessary related to hierarchical and non-hierarchical decomposition schemes
 - Consistency between global wafer fab scheduling and detailed area scheduling of full integration
 - Transfer of scheduling solutions from academia to the shop floor is challenging
 - Appropriate software representations of scheduling algorithms are required that can deal with distributed and missing data

Conclusions and Future Challenges (cont'd)

- ▷ **New challenges from automation pressure and global supply networks in the semiconductor industry**
 - Better understanding of the relationship between production planning and scheduling decisions for wafer fabs is desirable
 - Use of modern metaheuristics for scheduling wafer fabs globally
 - Better interaction of AMHS decisions and job scheduling decision
 - Storing and transporting lots must be done consistently with their production schedule
 - Interaction of scheduling decisions and Advanced Process Control (APC)
 - New constraints in scheduling to improve quality and yield

Discussion

