

Review

# Problems and Solution Methods of Machine Scheduling in Semiconductor Manufacturing Operations: A Survey

Jianxin Fang <sup>1,\*</sup>, Brenda Cheang <sup>2</sup> and Andrew Lim <sup>3</sup>

<sup>1</sup> International Business School, Xi'an Jiaotong-Liverpool University, Suzhou 215123, China

<sup>2</sup> Red Jasper Holdings, Westech Building, 237 Pandan Loop, Singapore 128424, Singapore

<sup>3</sup> School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 610032, China

\* Correspondence: jianxin.fang@xjtu.edu.cn

**Abstract:** Machine scheduling problems associated with semiconductor manufacturing operations (SMOs) are one of the major research topics in the scheduling literature. Lots of papers have dealt with different variants of SMOs' scheduling problems, which are generally difficult to tackle theoretically and computationally. In this paper, the single machine, parallel machines, flow shops, and job shops scheduling problems from SMOs have been reviewed, based on different processing constraints, e.g., batch processing, auxiliary resources, incompatible job families, and reentrant flow, etc., with the cycle time, flow time, and throughput-related performance measures. Given the vast and diverse nature of the current literature, it is urgently needed to make a systematic survey in order to identify the important research problems, research trends, and the progress of the related solution methods, as well as clarify future research perspectives. We hope the findings and observations could provide some insights to the researchers and practitioners in this domain.

**Keywords:** machine scheduling; semiconductor manufacturing; wafer production; survey



**Citation:** Fang, J.; Cheang, B.; Lim, A. Problems and Solution Methods of Machine Scheduling in Semiconductor Manufacturing Operations: A Survey. *Sustainability* **2023**, *15*, 13012. <https://doi.org/10.3390/su151713012>

Academic Editor: Mosè Gallo

Received: 17 July 2023

Revised: 22 August 2023

Accepted: 25 August 2023

Published: 29 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Semiconductor products such as microprocessors, memory chips, and microcontrollers are widely used both in personal appliances and industrial equipment. Integrated Circuits (ICs) are the main parts of semiconductor products, which are produced in semiconductor fabrication plants (also known as “fabs”) based on the miniaturization of the very-large-scale-integration (VLSI) technology [1]. According to the 2021 semiconductor industry association report, the 2020 global semiconductor sales value doubled that of 2000 and reached 440.4 billion US dollars (Report: 2021 State of the U.S. Semiconductor industry, <https://www.semiconductors.org/>, accessed on 1 July 2022). Though many industries suffered a major setback during the COVID-19 pandemic, the semiconductor industry continues its growth momentum to \$573 billion US dollars in 2022 according to the world semiconductor trade statistics forecast, thanks to the fast advancement of information and communication technologies such as Internet-of-Things, 5G, and AI.

Despite such a strong growth trend, the semiconductor industry is facing more competition nowadays. From the demand's perspective, the product mix is increasing with a shortening product life cycle, which leads to reduced delivery lead-times and stricter quality standards. From the supply's perspective, the throughput is limited and variable because production capacity is difficult to be expanded shortly and high-quality standards result in uncertain yields, which depend on the maturity of processing steps and products. Furthermore, the global distribution of facilities and increasing numbers of firms specializing in particular stages have let the management of the whole semiconductor supply chain extremely difficult. Many pieces of research have been conducted to study semiconductor supply chain topics such as network design, capacity

planning and production planning; see [2–4]. Since the pandemic, semiconductor manufacturing has been particularly under the spotlight as many industries including automobiles and smartphones are facing the chip shortage phenomenon. “How to expedite chip production and delivery” is a frequently asked question. Solutions such as capacity expansion and supply chain coordination have been discussed (Report: Chip shortage: how the semiconductor industry is dealing with this worldwide problem, <https://www.weforum.org/agenda/2022/02/semiconductor-chip-shortage-supply-chain/>, accessed on 1 July 2022). Among those solutions, one potential way is to use advanced production scheduling methodologies to increase machine utilization, reduce cycle time, and improve throughput. In addition, the rapid expansion of the industry also comes with an unexpected high water and energy consumptions with high greenhouse gas emissions [5,6]. These facts intensify the significance of sustainable development of the industry. Indeed, most corporations have proposed their sustainability strategies, e.g., water recycling and energy reduction. From the machine scheduling literature, it is well noted that many recent papers try to focus on the energy-related or pollution-related scheduling problems [7,8]. Machine scheduling indicates how to allocate machines over time to process jobs such that production requirements are satisfied, and one or more performance measures could be optimized [9]. Applying machine scheduling to semiconductor manufacturing operations (SMOs) is challenging. On one hand, most scheduling problems are NP-hard, thus requiring the development of efficient computational algorithms. On the other hand, scheduling methods are difficult to be implemented in practice, as manufacturing systems face unexpected events frequently.

Machine scheduling problems have been examined in different industries because their specific manufacturing processes lead to distinct formulations of relevant scheduling problems; see examples from furniture [10], solar cell [11], and potash [12]. In this paper, we will focus specifically on the semiconductor industry. The reasons are manifold. First, semiconductor manufacturing probably has the most complicated production processes, which provide rich resources for extracting interesting scheduling problems. Second, machines in SMOs are usually extremely expensive, and thus applying suitable scheduling methods to increase machine utilization is an important way to reduce operational costs and increase competitiveness. Third, SMOs are almost completely automated. Wafers are transported in the front opening unified pods (FOUPs) by the automated material handling systems (AMHS). Hence, scheduling algorithms can be much easier to be embedded into the manufacturing control system than other manually operated facilities. Moreover, practical pieces of evidence have shown that fabs have benefited a lot from effective scheduling strategies, although there is still potential for improvement [13].

Plenty of research regarding SMOs’ scheduling problems has appeared since the 1980s. However, SMOs’ scheduling problems remain difficult to analyze. Most of these challenges hinge upon its complicated manufacturing processes, which are summarized as follows:

- (1) Complex processing steps. It is common for a single process flow to consist of 300–900 processing steps over hundreds of machines and perhaps ten or more major process flows in wafer fabrication [14].
- (2) Reentrant flows. Wafers need to revisit some workstations multiple times to build the prescribed circuitry patterns, which complicates the production system greatly.
- (3) Diverse machine types. Wafers are generally transported in a lot of 25 units by FOUPs. Fabs have four typical machine types, i.e., batch processing machines (multiple lots per process), lot processing machines (single lot per process), discrete processing machines (single wafer per process), and cluster tools. A production line is composed of a mixed combination of these machine types, where improper coordination could lead to increased flow variability and excessive delays.
- (4) Distinct processing characteristics. Depending on the nature of machines and jobs, SMOs incorporate distinct processing constraints, e.g., batch processing, auxiliary resources, sequence-dependent setups, and multiple orders per job (MOJ).

- (5) Integral decisions. SMOs scheduling needs to be coordinated with other shop-floor control decisions, e.g., preventive maintenance (PM) and advanced process control (APC).

Several review papers have traced the development of SMOs' scheduling problems. For example, the production planning and shop floor control problems arising from wafer fabs are presented in [15,16], in which commonly used dispatching rules, order release, and deterministic scheduling methods are discussed. Problem configurations and solution methodologies concerning parallel batch machine scheduling from SMOs are presented in [17]. Job shop scheduling techniques from SMOs are summarized in [18], and deterministic SMOs scheduling problems mainly from the wafer fabrication process are elaborated in [19]. Those papers provide an excellent starting point for understanding relevant problems; however, they tend to focus only on certain aspects such as batch machine scheduling or job shop scheduling, based upon deterministic problem setting.

Our paper is most related to [19], which is used as the baseline for the current state of the art. This paper will review the most relevant papers in the past decade and try to figure out the latest developments for problems and solution techniques that are involved. Moreover, it will not only examine deterministic SMOs scheduling problems, but also stochastic and dynamic SMOs scheduling problems; it will examine SMOs scheduling problems from front-end operations as well as those from back-end operations; it will classify problems based on shop environment and processing characteristics and try to identify key problems in related operations.

Relevant papers are extracted from the Scopus and Web of Science databases based on two keyword groups: "machine scheduling & semiconductor manufacturing" and "machine scheduling & wafer fabrication" from 2011 to date. After removing unrelated topics such as transportation scheduling and production planning, more than 170 papers are left for further study. This paper also excludes the internal scheduling of cluster tools in this paper, as these problems usually utilize Petri net-based models to study tool configuration and schedulable issues, which are largely different from the scheduling problems that will be investigated here. For more information about cluster tool scheduling problems, readers could refer to a recent survey in [20]. Although extracting papers based on the above two keywords seem limited, it still generates sufficient papers that this survey could center around. Furthermore, we also add other relevant papers that are not identified during the above searching process in our survey. Since some scheduling problems are common in many industries, researchers may only explain problem descriptions with general background and focus more on the theoretical properties and algorithmic design. We will try to include these papers as well; however, it is impossible for us to summarize all scheduling papers with similar processing characteristics or job environment considering the extensive review scope, and most importantly it is not our main purpose.

The remainder of this paper is organized as follows. Section 2 outlines major SMOs and corresponding processing characteristics. Sections 3–6 elaborate and summarize single machine, parallel machines, flow shops, and job shops scheduling problems, respectively, based on deterministic and stochastic problems settings. Section 7 provides discussions of current research progress and future research perspectives.

## 2. Processing Characteristics, Classifications, and Notations

### 2.1. Processing Characteristics

In general, semiconductor manufacturing processes can be categorized into four stages, i.e., wafer fabrication, wafer probe, assembly, and final testing, where the first and last two stages are referred to as front-end and back-end operations, respectively. Among these stages, wafer fabrication is the most crucial stage because of its complex technology and high investment. The process starts with blank wafers made of silicon or gallium arsenide, and patterned layers are built up on wafers to produce ICs. Oxidation/diffusion, deposition, photolithography, etching, impurity doping, and planarization are the main operations of the wafer fabrication process. The schematic diagrams of semiconductor

manufacturing and wafer fabrication are shown in Figures A1 and A2 in Appendix A. Since SMOs' scheduling problems are our focus, the involved processing characteristics are emphasized below. For more detailed information about each processing step, readers could refer to [21,22].

- (1) *Batch processing.* Diffusion furnaces in the oxidation/diffusion and burn-in oven in the final testing operations are typical batch processing machines, where a batch of jobs can be processed simultaneously. However, jobs from different job families cannot be batched together in the diffusion furnace because the chemical elements of job families could be different, which may contaminate jobs from other families. This is usually referred to as *incompatible job families*. In batch machine scheduling problems, decisions include how to form batches and how to sequence batches. If jobs have *unequal release times*, the decision of whether to wait for impending jobs to build a fuller batch or to process current jobs in the batch right away should be determined. Mathirajan and Sivakumar [17] presented a literature meta-analysis for deterministic batch machine scheduling problems from SMOs, while Koo and Moon [23] provided a survey on real-time control strategies based on threshold and look-ahead policies for similar problems. More recently, Fowler and Mönch summarized the compatible and incompatible batch scheduling problems and the corresponding solution methodologies [24].
- (2) *Auxiliary resource.* Several SMOs rely on auxiliary resources to proceed. For instance, the photolithography step consists of coating, exposing, and developing operations to frame various regions in ICs. Patterns are transferred to wafer surfaces through reticles. Jobs can be processed only when both stepper machines and reticles are available. Since the patterns of different products and different layers of a product are not the same, and few reticles for each pattern are stocked, how to schedule jobs, machines, and reticles is a significant problem in this process. Similarly, test stations in the final testing stage require *multiple auxiliary resources* including tester, kit, and enabler assembly. Sometimes, jobs requiring the same auxiliary resources are processed serially in order to reduce machine setup time, which is also known as *serial-batching*, compared to the *parallel-batching* of batch processing machines.
- (3) *Sequence-dependent setup times.* Machine setup is a common procedure, which is defined as the time required to prepare necessary resources to perform a task. For example, impurity doping introduces controlled amounts of impurities into wafers in order to change their electronic properties, which could be different for various jobs. Hence, the dopant for each job or job family is not the same and thus needs to be changed frequently, in which case the sequence-dependent setup times should be considered. Setup times are sometimes job-family-based, which means jobs from *incompatible families* require longer setup times. A survey that specifically considers scheduling problems under setup times/costs can be found in [25].
- (4) *Reentrant flows.* Reentrant flows refer to jobs entering into production lines more than once, which is an intrinsic characteristic for many SMOs, especially for the photolithography and etching stations. Reentrant flows complicate production lines because jobs with different processing status compete for the same production resources. An inappropriate job scheduling with reentrant flows could lead to increased cycle times. Lin and Lee [26] summarized scheduling problems concerning reentrant flows and emphasized the commonly used solution methods.
- (5) *Advance process control (APC).* APC maintains machine process qualifications so as to prevent process excursions and increase tool utilization. Typical APC constraints include equipment health index, equipment qualification, preventive maintenance, and run-to-run control loop [27]. For instance, Obeid et al. [28] considered a *qualification-run* constraint, which refers to the action of conducting a test run if a machine has not processed certain job families for a pre-specified period that can either be measured in terms of time (time-based) or the number of jobs processed (count-based). Duc et al. [29] investigated *preventive maintenance* (PM) together with

machine scheduling in the etching operation. As chemical substances are accumulated over time during the process, suitable PM schedules, i.e., regular cleaning should be maintained to prevent yield loss. Sometimes, a maximum processing *time window* or *overlapping/nested time window* should be imposed between two or more consecutive operations. Otherwise, wafers could be contaminated. More information about binding scheduling decisions with APC systems for SMOs could be found in [27].

- (6) *Multiple order per jobs* (MOJ). As shown in [30], if each customer order is assigned to one FOUP in the 300 mm wafer fab, an extremely large number of FOUPs would need to be maintained, which can increase AMHS congestion. In practice, orders from multiple customers are combined into a single job in the FOUP in order to reduce the AMHS workload. This is commonly termed as MOJ scheduling, which are challenging as delivery performance is assessed at the order level while scheduling is made at the job level. Depending on machine types, MOJ scheduling can be further divided into *moj(item)*, *moj(lot)*, and *moj(batch)* for item-, lot-, and batch-processing machines. For *moj(item)*, job processing time is the sum of processing time of all wafers in all orders; for *moj(lot)*, job processing time is equal to the processing time of a single wafer; for *moj(batch)*, job processing time is equal to the longest processing time of the jobs containing different customer orders. The MOJ scheduling mainly involves two decisions, that is, how to group orders and how to schedule jobs.
- (7) Other characteristics. For instance, *machine dedication* indicates a machine may only be capable of processing a certain set of jobs. As specifications of new ICs evolve, old machines may not have the required processing capabilities. Hence, jobs should be scheduled to dedicated machines. In addition, *process precedence* constraint is a common constraint when studying scheduling problems involving multiple sequential operations.

## 2.2. Classifications and Notations

Similar to most scheduling literature, we adopt the three-field  $\alpha \mid \beta \mid \gamma$  notation to summarize SMOs' scheduling problems. The  $\alpha$ ,  $\beta$ , and  $\gamma$  fields represent the shop environment, processing characteristics, and performance measures, respectively.

Fabs are commonly regarded as complex flexible job shops because of the complex processing characteristics mentioned above. Each work area consisting of multiple work centers that are organized based on functions or logistics could be viewed as a flexible job shop or flow shop. Each work center has multiple parallel machines performing similar operations. Single machine refers to tool-based scheduling.

Performance measures of SMOs' scheduling problems can be mainly classified into three categories, i.e., cycle time-related measures, due date-related measures, and throughput-related measures. Total (weighted) job completion times and total (weighted) job flow times are typical cycle time-related measures. These two measures are the same if jobs have the same release time. Due date-related measures include total (weighted) tardiness, the total (weighted) number of tardy jobs, and maximum lateness. Throughput-related measures include makespan and work in progress (WIP). Other measures such as total setup times and electrical power cost have also been studied. Moreover, some papers consider multiple performance measures. Except for considering multi-measures separately, the multi-criteria analysis can be carried out in the following ways. Assuming  $Z_1, Z_2, \dots, Z_k$  are corresponding measures, by taking notations from [31], we let  $F_l(Z_1, Z_2, \dots, Z_k)$  indicate the objective is to optimize a linear combination of the  $k$  measures;  $\#(Z_1, Z_2, \dots, Z_k)$  refers to a Pareto optimization of the  $k$  measures;  $Lex(Z_1, Z_2, \dots, Z_k)$  refers to a hierarchical analysis by optimizing performance one after the other; and  $\in (Z_1/Z_2, \dots, Z_k)$  indicates the objective is to optimize  $Z_1$  while other measures act as constraints that are subject to a pre-defined upper or lower bound.

Table 1 summarizes the commonly used three-field notations, which may be not the same as in the original papers. Other notations will be introduced in place in due course. For instance, the  $1|r_j, p - batch, incompatible| \sum w_j C_j$  refers to a single batch ma-

chine scheduling problem that considers unequal job release times and incompatible job families with the objective of minimizing total weighted job completion times. Throughout this paper, notation  $j$ ,  $b$ ,  $o$ , and  $f$  refer to job, batch, order, and job family, respectively;  $N$  and  $M$  refer to the set of jobs and machines, and  $n$  and  $m$  relate to the number of jobs and machines. The tardiness of job  $j$  is defined as  $T_j = \max(C_j - d_j, 0)$ , where  $C_j$  and  $d_j$  are the completion time and due date of job  $j$ ; the earliness and tardiness of job  $j$  is defined as  $ET_j = |C_j - d_j|$ ; and the number of tardy jobs is defined as  $U_j = \sum \mathbb{I}_{\max(C_j - d_j, 0)}$ .

**Table 1.**  $\alpha$ ,  $\beta$ ,  $\gamma$  fields descriptions for relevant SMOs scheduling problems.

$\alpha$ (Shop Environment)		$\beta$ (Processing Characteristics)		$\gamma$ (Performance Measures)	
Notation	Description	Notation	Description	Notation	Description
1	single machine	p-batch	parallel batch	$\sum(w_j)C_j$	total (weighted) completion time
Pm	identical $m$ parallel machines	s-batch	serial batch	$\sum(w_j)F_j$	total (weighted) flow time
Qm	uniform $m$ parallel machines	incompatible	incompatible job families	$\sum(w_j)T_j$	total (weighted) tardiness
Rm	unrelated $m$ parallel machines	aux	auxiliary resources	$\sum(w_j)U_j$	total (weighted) number of tardy jobs
Fm	$m$ -machine flow shop	recrc	reentrant flows	$\sum(w_j)ET_j$	total (weighted) earliness and tardiness
HFc	$c$ -stage hybrid flow shop	prec	process precedence	$L_{\max}$	maximum lateness
HFFc	$c$ -stage hybrid flexible flow shop	$r_j$	release time of job $j$	$C_{\max}$	makespan
Jm	$m$ -machine job shop	$a_j$ , (B)	job size (batch capacity)	TSC	total setup time/cost
FJc	$c$ -stage flexible job shop	$s_{kl}$	sequence-dependent setup times from job $k$ to job $l$	throughput	total number of finished jobs
		$M_j$	machine dedication	EPC	electric power cost
		$tw_j$	time window	$\sum f_j(C_j)$	regular objectives

Machine scheduling is a typical kind of optimization problems, whose solution methods include exact analytical methods, heuristics, metaheuristics, and simulation methods. Table 2 summarizes the abbreviations of widely applied solution methods that have been used in the papers that are reviewed. In addition to deterministic scheduling problems, we will also investigate stochastic and dynamic SMOs scheduling problems and related solution methods.

The contents of the surveyed papers are evaluated rigorously based on our previous categorization. Detailed analysis is presented in the coming sections. For each shop environment, deterministic scheduling problems are discussed followed by stochastic and dynamic scheduling problems. Within each problem setting, results are presented based on processing characteristics and solution methods, and single measure is discussed followed by multiple measures. To facilitate the context flow, all papers reviewed in each section below are summarized in tables of Appendix B.

**Table 2.** Abbreviations of solving methods.

Analytical Methods		Metaheuristics		Rule-Based Methods	
MILP	Mixed Integer Linear Programming	VNS/ VND	Variable Neighborhood Search (Descent)	(B)ATC	(Batched) Apparent Tardiness Cost
MINLP	Mixed Integer Nonlinear Programming	(B)RKGA	(Biased) Random Key Genetic Algorithm	ATCSR	Apparent Tardiness Cost with Setups and Ready times
B&B	Branch and Bound	GA	Genetic Algorithm	FIFO	First In First Out
B&C	Branch and Cut	MA	Memetic Algorithm	FCFS	First Come First Serve
DP	Dynamic Programming	ACO	Ant Colony Optimization	ERT	Earliest Release Time
CG	Column Generation	SA	Simulated Annealing	LST	Latest Start Time
CP	Constraint Programming	TS	Tabu Search	SPT	Shortest Processing Time
		PSO	Particle Swarm Optimization	LPT	Longest Processing Time
		ALNS	Adaptive Large Neighborhood Search	EDD	Earliest Due Date
		GRASP	Greedy Randomized Adaptive Search Procedure		

### 3. Single Machine

In single-machine environment, multiple jobs are waiting to be processed. The key decision of single-machine scheduling problems is the sequencing of jobs, i.e., determining jobs' relevant start processing times.

#### 3.1. Deterministic Scheduling

Deterministic single-machine scheduling problems are presented below. We separate the discussion of batch machine scheduling from other processing characteristics, see Tables A1 and A2 in Appendix B, respectively.

##### 3.1.1. Batch Machine Scheduling Problems

As shown in Table A1, most batch machine scheduling problems arise from cleaning, oxidation/diffusion, and burn-in operations. Besides relevant problems investigated from SMOs, single batch machine scheduling problems have been studied extensively since the 1980s [32,33]. The simple versions of problems only consider identical/non-identical job sizes and job processing times. Sorting jobs in the longest processing time (LPT) order and fill the batches as full as possible in the descending LPT order minimizes  $C_{\max}$  if jobs have identical sizes and non-identical processing times [33]. However, if jobs have non-identical sizes and identical processing times, problems are NP-hard because they are equivalent to bin-packing problems. Recent research of single batch machine scheduling problems also considers non-equal job releasing times and incompatible job family constraints.

Single batch machine scheduling problems involve two decisions, i.e., how to form batches and how to sequence batches. For instance, the  $1|p - batch, a_j, B|C_{\max}$  problem can be mathematically formulated as: minimize  $\sum_b P_b$ , s.t.  $\{\sum_b X_{jb} = 1, \forall j; \sum_j a_j X_{jb} \leq B, \forall b; P_b \geq p_j X_{jb}, \forall j, b; X_{jb} \in \{0, 1\}, \forall j, b\}$ , where  $P_b$  and  $p_j$  refer to the processing time of batch  $b$  and job  $j$ . The decision variable  $X_{jb}$  indicates whether to assign job  $j$  to batch  $b$ , and the first two constraints determine the batch formation decision. Minimizing  $C_{\max}$  is relatively easier compared to other performance measures, because only batch formation decision is required while batch sequencing decision is determined implicitly. The below discussions are expanded from this basic problem. Studies differ mainly in examined performance measures and solution techniques.

Chen et al. [34] studied the  $1|p - batch, a_j, B|C_{\max}$ . They claimed that jobs with similar processing times should be batched together. Based on this observation, they introduced the waste ratio of batch (WRB) and proved to minimize  $C_{\max}$  is equivalent to minimizing WRB. Therefore, a constrained agglomerative clustering of batches (CACB) algorithm with

worst-case complexity of  $O(n^4)$  was proposed. Experiments testing up to 500 jobs showed that CACB outperforms the best-fit longest processing time heuristic (BFLPT) and a GA method from [35]. Lee and Lee [36] studied a similar problem by constructive-based and improvement-based MILP heuristics, which were developed based on the WRB from [34]. Initial batches were generated from successive knapsack formulation in the constructive-based MILP heuristic, while waste improvement was used in the improvement-based MILP heuristic. Both heuristics were then combined with some greedy rules, e.g., the longest batch processing time and minimum remaining capacity to improve their performances. Experiments showed the constructive-based heuristic with a rolling scheme provides better performance though it requires longer computational time. Moreover, Li and Zhang [37] studied a generalized  $1|p - batch, a_j, B|C_{max}$  considering two-dimensional job sizes. They developed a MILP model and two heuristics. The first heuristic is a BRKGA algorithm with initial solutions based on four job sequencing rules including LPT, SPT, largest job size (LS), and smallest processing time/rime/size ratio (SR) from [33], and a batch formation rule of best-fit first (BFF). The second heuristic is a hybrid bin loading (HBL) algorithm that transforms the original problem into a successive bin-loading problem. Experiments showed that both heuristics provide comparable results in moderate instances and outperform CPLEX in large instances compared to the MILP model.

Parsa et al. [38] investigated the  $1|p - batch, a_j, B|\sum C_j$  using a hybrid max-min ant system (MMAS), which includes an MMAS to generate job sequences, a DP algorithm to form initial batches, and a local search to improve solution quality. The algorithm has a computational complexity of  $O(t_{max}n_a n^2)$ , where  $t_{max}$  and  $n_a$  refer to a maximum number of iterations and the number of ants, respectively. Experiments showed the algorithm is able to generate better solutions than the GA from [35] and the heuristics from Jolai and Dupont [39].

When jobs have unequal release times, a machine could be deliberately left idle as waiting for impending jobs to build fuller batches may bring benefits. The  $1|p - batch, r_j, a_j, B|C_{max}$  was studied in [40–42]. Shao and Chen [40] applied an ACO algorithm to generate job sequences and a DP algorithm to form batches, in which the ACO uses information of job processing and release times to update pheromone trails. By extending the WRB definition, Xu et al. [41] defined the waste and idle space (WIS), which includes the waste space from non-identical job sizes or processing times and the idle space from unequal job release times. They proved minimizing  $C_{max}$  is equivalent to minimizing the WIS, based on which a first-fit WIS earliest release time (FFWIS-ERT) heuristic and an ACO algorithm are developed. Zhou et al. [42] observed that two jobs should be batched as far as possible if they have close processing and release times, and their total job size is less than the residual batch capacity. Based on this observation, they defined the job distance function and then developed the first-row start (FRS), minimum distance start (MDS), and updated distance (UD) heuristics. Experiments showed their methods outperform the ACO in [41]. In addition, Mathirajan et al. [43] studied the  $1|p - batch, r_j, d_j, a_j, B|\sum w_j T_j$  considering both unequal job release times and due dates. They developed a constructive greedy heuristic (CGH) and an SA algorithm, in which the CGH sequences jobs based on EDD, ERT, and LST rules, and the SA improves initial solutions from CGH based on a forward-backward iteration mechanism.

Some studies consider the incompatible job family constraint. Dauzère-Pérès and Mönch [44] studied the  $1|p - batch, incompatible, a_j, B|\sum (w_j) U_j$ . They proposed two MILP formulations. The first MILP defined the positional variable as batches where multiple jobs can be assigned to the same position, while the second MILP defined the positional variable as jobs where batches are formed by consecutive jobs from the same family. Both formulations are based on the observation that there exists an optimal schedule such that all on-time jobs are processed before tardy jobs. An RKGA was also developed. Experiments showed the first MILP provides better upper bounds while the second MILP provides better lower bounds, and the RKGA is able to find high-quality solutions for large instances. In addition, Cheng et al. [45] studied the  $1|p - batch, incompatible, a_j, B|C_{max}$

and the  $1|p - batch, incompatible, a_j, B|\sum C_j$ . They proved both problems are NP-hard and developed two constructive heuristics with time complexity of  $O(mn \cdot \log(n))$ .

Moreover, Lu et al. [46] specially considered the processing time deterioration effect and resource constraints in a single batch machine scheduling problem. They proved the problem to be NP-hard, and a VNS algorithm is designed to solve the problem.

### 3.1.2. Non-Batch Machine Scheduling Problems

Processing characteristics including MOJ, APC, PM, and interfering jobs have been investigated in single non-batch machine scheduling problems. MOJ has been examined under the  $moj(item)$  and  $moj(lot)$ , and mainly involve two decisions, i.e., how to group different orders into jobs and how to schedule these jobs. Mason and Chen [47] proved the  $1|moj(item)|\sum C_o$  and  $1|moj(lot)|\sum C_o$  are both NP-hard. They developed a MILP formulation and proved optimal schedules for each problem. For  $1|moj(lot)|\sum C_o$ , they showed that jobs are sequenced in non-increasing order of the number of orders in the job is optimal. For  $1|moj(item)|\sum C_o$ , they showed that the total number of jobs in the optimal schedule is the minimum between the number of orders and the number of FOUPs, and the smallest order size first (SOSF) rule is optimal when the number of orders is not greater than the number of FOUPs. Sobeyko and Mönch [48] studied the  $1|moj(item)|\sum w_o C_o$  and  $1|moj(lot)|\sum w_o C_o$ . They indicated that the main decision of both problems is job formation since job sequencing can be determined optimally by non-increasing  $\sum_{o \in j} w_o$  for  $moj(lot)$  and non-increasing  $\sum_{o \in j} w_o / \sum_{o \in j} a_o$  for  $moj(item)$ . They used a grouping genetic algorithm (GGA) to form jobs, which was shown to outperform an RKGA that first sequences orders and then forms jobs. Sarin et al. [49] studied the  $1|moj(item)|\sum C_o$  under both uncapacitated and capacitated carriers. Structural properties are provided for both problems, based on which a B&B method is developed. Experiments showed their formulation is better than [47]. Sobeyko and Mönch [50] proved the  $1|moj(item)|\sum w_o T_o$ ,  $1|moj(item)|\sum w_o U_o$ ,  $1|moj(lot)|\sum w_o T_o$ , and  $1|moj(lot)|\sum w_o U_o$  problems are NP-hard. They developed a GGA for each problem, whose main difference is the way of tackling reinsertion and mutation procedures. Experiments showed the GGA provides a better solution than the GA used in [51]. Rocholl and Mönch [52] studied the  $1|moj(lot), incompatible|\sum ET_j$  with common non-restrictive due dates. They proposed a GA-based method that first assigns orders to jobs and then assigns jobs to early and tardy sets.

Cai et al. [53] studied the APC with a count-based qualification-run in the  $1|s_{kl}, APC|\sum Lex(C_{max}, U_j)$ . In this problem, both job schedules and machine qualification-run schedules need to be determined. The problem is proved to be strongly NP-hard. In addition, the optimal schedule for the two-family case was derived, and the structure of optimal schedules for the multi-family case was analyzed. Based on these properties, they developed a constructive-based heuristic, which is efficient in solving up to 20 job families with 10–30 jobs in each family.

Similar to APC, when considering PM activities in machine scheduling, decisions on job schedules and PM schedules have to be determined. Pang et al. [54] studied the  $1|r_j, cleaning|\#(\sum w_j T_j, \sum C_j)$ , where machines should be stopped for cleaning before the maximum allowable contamination has accumulated. They showed this problem is NP-hard and developed a scatter SA algorithm. Chung et al. [55] studied  $1|cleaning|\sum C_j$ . They provided a MILP formulation and explored the properties of optimal schedules, based on which a constructive-based heuristic that solves up to 500 jobs was built.

Furthermore, Ramacher and Mönch [56] studied the  $1|\#(\sum w_j C_j, L_{max})$ ,  $1|\#(\sum C_j, L_{max})$ , and  $1|\#(\sum w_j C_j, \sum w_j C_j)$  with two interfering agents that have their own jobs and private objectives. They proposed a decentralized automated negotiation mechanism where the mediator proposes contracts based on VNS. Experiments showed solutions generated by this mechanism are closer to the Pareto frontier compared to a centralized approach with full information.

### 3.2. Stochastic and Dynamic Scheduling

Relatively few papers have tackled stochastic and dynamic single-machine scheduling problems. The  $1|p - batch, incompatible, r_j, a_j, B|\sum w_j T_j$  is studied in [57], in which a hybridized GGA with initial solutions from a BATC-II list scheduling approach based on a time window decomposition scheme is developed. Jia et al. [58] studied  $1|r_j, incompatible, recrc|\sum w_j T_j$ . They developed a job-family-oriented MILP formulation, which is then combined with a rolling horizon (RH) strategy for real-time scheduling. In addition, Huang et al. [59] applied the queueing control model to study production and PM joint scheduling problems. They indicated that the state-dependent  $c\mu$ -rule is optimal under the constant service ratio assumption. Jun and Lee [60] proposed the learning of dispatching rules, which consist of a decision tree approach that generates rules automatically from existing schedules and a feature-construction-based genetic programming to improve the rule.

From above single-machine scheduling problems, it can be inferred that many scheduling problems exist depending on different processing characteristics and performance measures. Batch machine, auxiliary resources, and sequence-dependent setup times are commonly investigated characteristics.

## 4. Parallel Machines

This section elaborates scheduling studies on parallel machines environments, in which a group of identical or non-identical machines are assigned to process similar operations. Different from single-machine environments, decisions on how to allocate jobs to machines and how to sequence jobs on each machine should be made.

### 4.1. Deterministic Scheduling

Similar to the single-machine environment, batch scheduling problems and non-batch scheduling problems are discussed separately; see Tables A3 and A4 in Appendix B, respectively. Furthermore, the specific machine environment such as identical or unrelated parallel machines is listed inside the parentheses after the process characteristics.

#### 4.1.1. Batch Machine Scheduling Problems

In parallel batch machine scheduling problems, decisions include how to form jobs into batches, how to allocate formed batches to machines, and how to sequence batches for each machine. Most parallel batch machine scheduling problems are strongly NP-hard, see [61]. Parallel batch machine scheduling problems have been studied under different processing constraints, e.g., unequal job release times, incompatible job families, and machine dedication.

The  $P_m|p - batch, r_j, a_j, B|C_{max}$  is studied in [62,63]. Chen et al. [62] designed a GA and an ACO algorithm to form batches and used an ERT-LPT rule to schedule batches. Furthermore, two lower bounds based on job splitting were developed. Results showed GA outperforms ACO in small instances while ACO dominates GA in large instances. In their method, batch allocation and sequencing decisions are tackled by a single ERT-LPT rule. Similar to [42], the job distance definition and proposed three distance matrix-based heuristics (DMBHs) is used to form batches [63]. Then, a modified greedy longest processing time (GRLPT) heuristic was used to schedule batched jobs to machines. Experiments showed their methods outperform the GA and ACO in [62]. Both [62,63] used rule-based heuristics to tackle batch allocation and sequencing decisions.

Similar problems have also been examined under unrelated parallel machines, where machines have either non-identical processing times or capacities. Arroyo and Leung [64] studied the  $R_m|p - batch, r_j, a_j, B|C_{max}$ , in which machines have non-identical job processing times. They proposed three constructive heuristics considering ERT, first fit (FF), and best fit (BF) rules. They showed that the heuristic that forms and schedules batches in a single phase outperforms the other two heuristics, and solutions deteriorate as job sizes increase or release times decrease. Hulett et al. [65] studied  $R_m|p - batch, a_j, B_m|\sum w_j T_j$  with

nonidentical machine capacities. They proposed a PSO algorithm that uses the smallest position value rule to construct batching and scheduling decisions.

Lin et al. [66] studied the  $R_m|p - batch, a_j, B|\#(C_{\max}, \sum w_j C_j, \sum w_j T_j)$ . They proposed two LP-based heuristics for  $\#(C_{\max}, \sum w_j C_j)$  and  $\#(\sum w_j C_j, \sum w_j T_j)$  and designed a GA that sorts populations based on the Pareto ranking of  $\#(C_{\max}, \sum w_j C_j, \sum w_j T_j)$ . Experiments showed the GA outperforms the LP-based heuristics in terms of solution quality and the number of non-dominated solutions. Jia et al. [67] studied the  $P_m|p - batch, r_j, a_j, B|\#(C_{\max}, EPC)$ , in which the EPC is computed as the sum of processing power and idle power multiplied by the electricity price function. They developed a Pareto-based ACO algorithm, whose computational complexity is  $O(T_{\max} N_a m n^2)$  where  $T_{\max}$  and  $N_a$  refer to the maximum number of iterations and number of ants.

The  $P_m|p - batch, a_j, B, incompatible|\sum w_j T_j$  is studied in [68,69]. Alemeder and Mönch [68] designed an ACO and a VNS with initial solutions from the ATC-BATC rule and decomposition heuristics. Lausch and Mönch [69] developed an ALNS. Experiments showed the ALNS outperforms the ACO while being comparable to the VNS in [68]. Mönch and Roob [70] studied a similar problem with regular sum objectives and indicated that batch formation decisions can be regarded as a transportation problem. Therefore, a minimum cost flow problem was formulated to tackle batch formation decisions, and a BRKGA was designed to tackle the batch assignment and sequencing decisions. Experiments showed this method is competitive with the VNS in [68] for the  $P_m|p - batch, a_j, B, incompatible|\sum w_j T_j$  and the MILP heuristic in [44] for the  $1|p - batch, incompatible, a_j, B|\sum (w_j) U_j$ .

Moreover, incompatible job families together with unequal job release times scheduling problems are examined in [71–73]. Chiang et al. [71] studied the  $P_m|p - batch, r_j, a_j, B, incompatible|\sum w_j T_j$ . They proposed an MA that encodes batch formation and batch sequence simultaneously in the chromosome and decodes machine allocation by assigning batches to machines at the earliest available time. Experiments showed that the MA outperforms GA methods through benchmark examples from [74]. Rocholl et al. [72] studied the  $P_m|p - batch, r_j, a_j, B, incompatible|\#(\sum w_j C_j, EPC)$ . They formulated a MILP for the general problem and a simplified MILP based on the properties of optimal schedules for a special problem with the same job size and release time. Rocholl et al. extended the above problem to the Pareto optimization of  $\sum w_j T_j$  and  $EPC$  [73]. Similar approaches were applied for the reduced problem, and three versions of grouping GA with distinct solution representations and decoding schemes were developed as well.

In addition, processing characteristics such as auxiliary resources [75], machine dedication [76], and process precedence [77] have also been studied. Such characteristics let related problems become more challenging. For instance, the  $R_m|p - batch, a_j, B, M_j, tw_j, incompatible|Lex(C_{\max}, \sum w_j C_j, \sum T_j)$  is studied in [76], where jobs can only be processed by a subset of machines. A VND algorithm based on six neighborhood structures was designed to tackle the problem.

#### 4.1.2. Non-Batch Machine Scheduling Problems

Many papers tackled auxiliary resources in parallel non-batch machine scheduling problems. Other processing characteristics including APC, batch arrivals, job precedence, and limited waiting time have also been considered (see Table A4).

We first discuss auxiliary resources scheduling problems. Since auxiliary resources are usually limited in fabs, decisions regarding resource allocation and job scheduling should be made simultaneously. One of the popular auxiliary resources in fabs is reticles that are used in photolithography operations. Klemmt et al. studied a parallel machine scheduling problem considering auxiliary resources [78]. They designed a multistage MILP approach, in which multiple objectives and multiple constraints were considered based upon hierarchical four-stage optimization models. Since reticles need to be recalibrated after a certain period of time, Yan et al. studied a similar problem by considering the reticle expiration objective [79]. They proposed a B&C method to obtain near-optimal solutions

and developed a two-phase heuristic that considers a subset of constraints in each phase so as to improve the computation efficiency. Furthermore, resource transportation time is considered in [80,81]. Bitar et al. [80] studied the  $R_m|aux, s_{lk}|(\sum w_j C_j, throughput)$ , in which only one reticle is stocked for each job type, and a unitary transport time is needed for each movement. They proposed an MA whose coding and decoding schemes are based upon some neighborhood structure properties. Ham [81] formulated a similar problem as a MILP and showed that the CP model combined with variable ordering heuristic is able to find optimal solutions for large instances, e.g., up to 200 jobs, 30 machines, and 200 reticles. Chung et al. [82] studied the  $P_2|aux|C_{max}$ . They proposed three LPT-based heuristics, which were proven to have a worst-case performance ratio of 3/2. Ham et al. [83] studied an integrated machine and vehicle scheduling problem in the photolithography area, in which a constraint programming method is proposed.

Given resource changeovers will incur long setup times, it is suggested that jobs requiring the same resource should be processed sequentially to reduce machine setup time. Deng et al. considered such a situation in the burn-in operations, where machine-resources-temperature combinations are preset before job scheduling so that long setup times can be avoided [84]. They modeled the problem as a MILP and designed a two-phase reactive GRASP. Experiments showed the MILP generates larger gaps than the GRASP when resources are highly limited. Wang et al. utilized an EDD with the least recipe changeover (EDDLC) rule for the  $R_m|aux, r_j, s_{lk}|F_l(C_{max}, U_j)$  [85]. Chen et al. studied the  $R_m|r_j, s_{lk}, tw_j|Lex(throughput, C_{max})$  considering sequence-dependent setup times and job expiration time [86]. A two-phase MILP model and a hybrid TS algorithm were applied, in which experiments showed the latter performs better on solution quality and computational time. Moser et al. studied  $R_m|s_{lk}, M_j|Lex(\sum T_j, C_{max})$  [87]. They modelled the problem as a MILP and proposed several variants of SA with different neighbourhood search procedures.

In addition, Zhang et al. studied the  $Q_m|aux, recrc, s_{lk}|throughput$  considering multiple resources coordination, i.e., a tester, a handler, and an enabler, in the final testing operation [88]. They proposed a state-action-reward-state-action (SARSA)-based reinforcement learning algorithm that incorporates five actions defining how to choose jobs and test resources. Experiments showed the SARSA method outperforms rule-driven industrial methods by nearly 70%. Multi-resource scheduling problems regarding wafer probe operation are studied in [89–91]. Ying and Lin [91] designed a hybrid artificial immune system (HAIS), which was shown to outperform the iterated greedy heuristic in [90] and the GA, ACO, and TS in [89]. Furthermore, Munoz et al. [92] examined a dual-resource constrained machine scheduling problem from the photolithography area, i.e.,  $P_m|r_j, s_{lk}, aux|\sum C_j$ . They proposed a hybrid model that includes a MILP with initial solutions from a constructive heuristic to reduce solution space. Similarly, Ref. [93] studied  $R_m|r_j, s_{kl}, aux|\sum w_j C_j$  from the photolithography operation. They proposed two MILP formulations and integrates VNS into GA as the solution improvement procedure. Ref. [94] studied a similar problem considering machine deterioration and preventive maintenance. They first theoretically identified the optimality property, based on which an exact scheduling algorithm is developed.

Some studies tried to incorporate APC decisions into parallel machine scheduling problems. For example, Refs. [95,96] studied the  $R_m|incompatible, APC, s_{lk}|F_l(\sum C_j, \sum machine disqualifications)$  considering time-based qualification-run APC. Obeid et al. showed the problem is strongly NP-hard. They developed two MILP models with job-based and family-based decision variables separately [95]. Moreover, a scheduling-centric heuristic (SCH) to minimize machine setup times and a qualification-centric heuristic (QCH) to minimize machine disqualifications were also developed. Experiments showed the MILP with a family-based decision variable is more effective, and there is a trade-off between the total completion time and the total number of machine disqualification times. In contrast to [95], Ref. [96] proposed a new MILP model so that machine disqualification at the end of the schedule could be considered. They developed a CP model, and

two improvement procedures for the SCH and QCH, which generates better solutions than [95]. Kao et al. considered the machine health factor (MHF), which indicates that job quality risk is modeled by the MHF function and machine processing capability [97]. They proposed two MILP formulations in the case of static and dynamic MHF separately. The MHF deteriorates based on scheduling decisions in the dynamic model. Experiments showed that incorporating MHF into scheduling decisions helps maintain a balance between productivity and quality. Pang et al. studied the  $R_m|r_j, d_j, PM|C_{\max}$  with periodic maintenance activities considering machines will deteriorate over time [98]. They showed the problem is NP-hard and developed an iterative feature-based method. Zhang and Chen considered both machine health conditions and PM as the APC tools in the related scheduling problem [99]. They developed a VNS heuristic that integrates with a B&B algorithm for large instances.

Chung et al. studied the  $P_m|r_{batch}|\sum C_j$  with batch arrivals [100]. They applied a B&B algorithm considering several dominance properties of partial schedules and designed an iterative heuristic based on the MILP model of  $P_m||\in(C_{\max}/\sum C_j)$  that solves each arrival per stage.

Wang et al. examined the  $R_m|r_j, s_{lk}, tw_j|(\sum w_j C_j / \sum exceeded\ time)$  with the limited waiting time [101]. They formulated the problem as an MINLP model and developed a GA based on multi-subpopulation parameters with a hybrid estimation of distribution (MSPHEDA). Rolim et al. studied the  $R_m|restrictive\ due\ date|\sum w_j ET_j$ , in which two MILP and two constructive heuristics are developed [102].

In addition, Rocholl and Mönch examined the  $P_m|moj(item), incompatible|\sum ET_j$  and  $P_m|moj(lot), incompatible|\sum ET_j$  with non-restrictive common due dates, which were proved as NP-hard [103]. They developed a MILP model and proved some properties of optimal schedules for each problem. A BRKGA decomposition method was also designed, which is shown to outperform simple listing scheduling and bin-packing heuristics.

#### 4.2. Stochastic and Dynamic Scheduling

Stochastic and dynamic parallel machine scheduling problems are shown in Table A5 of Appendix B. Batch machine scheduling problems and non-batch machine scheduling problems are discussed separately.

##### 4.2.1. Batch Machine Scheduling Problems

Since jobs arrive continuously in dynamic scheduling problems, mathematical programming methods cannot be applied directly because the system does not know how many jobs need to be scheduled. However, this issue could be tackled by limiting the time horizon of the scheduling problem and then solving it iteratively. This procedure is commonly referred as time-window decomposition or the rolling horizon (RH) method. Klemmt et al. [104] studied the  $R_m|r_j, p - batch, incompatible|C_{\max}, \sum w_j C_j, \sum w_j T_j$ . They developed a time window decomposition approach where each subproblem was solved by a related MILP model. Then, a simulation model was applied to evaluate solutions, which outperform the BATC rule. Kohn and Rose [105] combined the VNS with time window decomposition for the  $R_m|r_j, M_j, p - batch, incompatible|\sum T_j$ . They indicated that the length of decomposition intervals has a large effect on the scheduling quality. Usually, small intervals generate better scheduling results. However, the trade-off between solution quality and computational complexity should be balanced for real-time scheduling in fabs. Tajan et al. developed an online model predictive control (MPC) heuristic for the  $P_m|r_j, p - batch, incompatible|\sum C_j$  [106]. The MPC tackled the batch decision sequentially by limiting the number of job families and future job arrivals. In addition, Jia et al. studied the  $P_m|r_j, p - batch, incompatible, recrc|\sum w_j T_j$  [107]. They proposed batch-oriented and family-oriented scheduling algorithms, which are integrated into an RH strategy in a real-time scheduling simulation platform. Results showed the family-oriented algorithm outperforms the batch-oriented algorithm.

Rule-based methods have also been applied. For instance, Kim et al. [108] proposed three rule-based scheduling algorithms with or without look-ahead checks for the  $P_m|r_j, s_{lk}, p - batch|\sum T_j$ . Jia et al. studied the  $P_m|r_j, p - batch, incompatible, recrc|\sum w_j T_j$  based on a real-time closed-loop control heuristic, which uses a pull-pull-push-push strategy if the number of jobs of certain families is less than the maximum batch size, and a GA otherwise [109]. Chen et al. developed a learning-based adaptive dispatching method (LBADM) for the  $P_m|r_j, p - batch, incompatible|F_l(\sum w_j T_j, C_{max})$  [110]. The LBADM includes an adaptive ACO to obtain initial solutions, a dispatching rule that considers multiple indicators, and a parameter learning method that dynamically modifies the parameters of the dispatching rule.

#### 4.2.2. Non-Batch Machine Scheduling Problems

Hung et al. studied a photolithography rescheduling problem when new events make the original schedule unavailable [111]. The rescheduling starts with the best solution of original schedules as initial solutions that are then re-optimized by SA, GA, and TS.

Photolithography scheduling problems are studied in [112–114]. Ham and Cho proposed a two-stage model that includes a MILP formulation for the job-reticle-machine assignment in the first stage, and the real-time dispatching (RTD) for job sequencing in the second stage [112]. Different from fixed periodic RH methods, Zhang et al. used a variable time interval-based RH to study online scheduling for the  $R_m|r_j, M_j, aux|\sum C_j$  [113]. The variable time interval is determined by the cumulative processing time of arrival jobs. Within this RH framework, they proposed an improved imperialist competitive algorithm (ICA) and a local search method. They verified the superiority of their method through real fab applications. Kim et al. applied a Deep-Q-Network (DQN) with an action filter to solve the  $P_m|r_j, s_{lk}, M_j, aux|\sum T_j$  [114].

Chien and Lan integrated the DQN and a hybrid GA (HGA) to solve the  $R_m|r_j, s_{lk}, incompatible|C_{max}$  [115]. Cao et al. studied the  $R_m|r_j, p_j, s_{lk}|C_{max}$  considering stochastic processing time depends on gamma or log-normal distribution [116]. They designed a simulation optimization approach, in which the first stage uses a GA to generate initial solutions with great potential and the second stage uses an optimal computing budget allocation (OCBA)-based simulation to accurately evaluate the solutions. Lee et al. proposed a deep Q-learning (DQL) for a  $R_m|r_j, s_{lk}, M_j, PM|F_l(\sum s_{lk}, \sum T_j)$  [117].

In parallel machine scheduling problems, the commonly investigated processing characteristics and performance measures are similar to a single machine environment. However, depending on the machine differences, the machine dedication and PM-related constraints have also been studied. In addition, since the decisions are more involved in parallel machine settings than in single machine, researchers also focus more on designing efficient computational algorithms instead of theoretically characterizing optimal schedules.

### 5. Flow Shops

An m-stage flow shop comprises m machines ordered in series, where each job has to be processed by each machine in the same order. Only one machine exists at each stage in m-stage flow shops while there is more than one machine in at least one stage in hybrid flow shops as an effect of increasing stage processing capacity. In addition, some stages can be skipped by some jobs in (hybrid) flexible flow shops. In both deterministic and stochastic flow shop scheduling problems, discussions of flow shop scheduling problems will be presented followed by hybrid (flexible) flow shops. Within each class, non-reentrant flow shops will be discussed before reentrant flow shops, and flow shops with a specific number of stages will be discussed before those with a non-specific number of stages.

#### 5.1. Deterministic Scheduling

Table A6 presents deterministic flow shop scheduling problems whose specific machine environment is also listed inside the parentheses after main process characteristics.

### 5.1.1. Flow Shops

Liu studied the  $F_3|moj(item, lot, batch), r_j, a_j, d_j|\sum w_j T_j$  [118]. They designed a GA whose chromosome includes three segments so that job formation, batch formation, and job sequencing decisions can be tackled simultaneously.

Yao et al. studied the  $F_2|r_j, p - batch \rightarrow discrete, B, (tw_j), (incompatible)|C_{max}$  considering a batch processing machine is followed by a discrete machine [119]. They proved that the above problem with incompatible job families, either with/without time window constraint, is NP-hard, and proposed time-symmetric EDD-FIFO heuristics (TSEDD-FIFO) whose worst-case performance ratios are no larger than 2. However, without considering incompatible families, the above problem either with/without time window constraint is polynomially solvable.

Kim and Lee studied the  $F_3|tw_{j1}, tw_{j2}|C_{max}$ , which considers overlapping time window constraints between consecutive machines [120]. Under this constraint, jobs need to satisfy the processing time window of both machines 1 and 2 and machines 1 and 3. They modeled the problem as a MILP and developed a B&B algorithm utilizing several dominance properties on partial schedules.

Celano et al. studied the  $F_m|incompatible, limited buffer capacity, s_{lk}|C_{max}$  in wafer probe operations [121]. Considering sequence-dependent setup times and limited buffer capacity, they developed a GA to solve the problem. Moreover, benchmark comparisons were carried out to the modified NEH [122] and the TS [123], which showed the GA performs better. Noroozi et al. studied the  $F_m|p - batch|F_l(\sum ET_j, C_{max})$  [124]. A hybrid GA, hybrid SA, and improved PSO combined with adaptive learning-based evolution procedures were applied.

Refs. [125–128] investigated flow shop scheduling problems in wet-etching operations, which consist of successive chemical and water baths with no-intermediate storage (NIS), and a single robot is used to transfer wafers between baths. Wafers that are processed in chemical baths must be moved out immediately, otherwise they will become contaminated. However, wafers in water baths could have a minimum residence time. These features are known as zero-wait (ZW) and local storage (LS) constraints, respectively. Decisions in these problems include job sequencing and robot move schedule. Ref. [125] developed a MILP model, while Refs. [126,127] proposed a MILP-based decomposition method such that the job schedule and transfer schedule can be dealt with separately. Ref. [128] addressed a similar problem, in which job sequencing is tackled by an integrated exact optimization approach and a GA-based heuristic method. Nevertheless, their results have not been compared.

### 5.1.2. Hybrid (Flexible) Flow Shops

Wang et al. studied the  $HF_2|r_j, discrete \rightarrow p - batch, M_j|C_{max}$  considering discrete machines in the first stage are followed by batch machines in the second stage [129]. They developed a MILP and a reduced MILP based upon a balanced workload among machines. Heuristics such as the batch first in first out (BFIFO) rule has also been applied to large instances. Qin et al. designed dispatching rules based on genetic programming (GP) to examine the  $HF_2|r_j, p - batch \rightarrow discrete, B, tw_j|C_{max}$  [130].

Wu and Chiou [131] examined the in-line steppers scheduling problem under a new process introduction, i.e.,  $HF_c|aux, limited buffer capacity|C_{max}$ . They designed a GA with four crossover operators and three mutation operators and showed the method outperforms the common rule-based heuristics. Fu et al. [132] studied a back-end scheduling problem represented as the  $HFF_c|s_{lk}, M_j, PM, aux, skip, incompatible|\sum T_j$ . They developed a two-stage deterministic scheduling system, which includes an optimizer to generate production plans based on a relaxed MILP model and a scheduler to schedule jobs based on priority rules.

Tan et al. [133] proved the  $HF_2|r_j, p - batch, incompatible|\sum w_j T_j$  is NP-hard. They developed an iterative stage-based decomposition method, where each stage was tackled

by a VNS method. Experiments indicated that stage-based decomposition is superior to the time window decomposition method.

Some studies also tackled the reentrant hybrid flow shops. For example, Hekmatfar et al. studied the  $HF_2|recrc, r_j, s_{lk}|C_{max}$ , where the first stage is a reentrant stage consisting of parallel machines [134]. A MILP model was formulated for the problem. In addition, two metaheuristics, i.e., an RKGA and a hybrid GA (HGA), and four constructive heuristics were designed. Experiments showed the HGA outperforms other methods. The  $HF_c|recrc, aux, s_{lk}|Lex(device\ shortage, throughput, machine\ utilization, C_{max})$  is studied in [135,136]. Bard et al. [135] proposed a three-phase approach based on the GRASP that tackles multi-pass and machine changeovers separately, while Gao et al. [136] developed a decomposition method that tackles machine setups, job sequencing, and machine resets sequentially. Experiments on real instances showed [136] generate better solutions than [135]. Lin et al. studied reentrant hybrid flow shop scheduling problems considering limited inventory buffers on each workstation and a central stocker [137]. By designing a hybrid harmony search GA (HSSGA) to tackle the problem, experiments showed the manufacturing condition with both inventory buffer and stocker could improve productivity.

## 5.2. Stochastic and Dynamic Scheduling

Table A7 presents stochastic and dynamic flow shop scheduling problems.

### 5.2.1. Flow Shops

Refs. [138–140] considered a flow shop that consists of two sequential batch machines. Ref. [138] studied the  $F_2|r_j, p - batch \rightarrow p - batch, tw_j|C_{max}$ , in which a MILP-based heuristic that limits the number of jobs scheduled each time was developed. Refs. [139,140] investigated the  $F_2|r_j, p - batch \rightarrow p - batch, tw_j, recrc, incompatible|C_{max}$ . Jia et al. [139] designed a pull-based heuristic algorithm to ensure full-batch processing, while Jia et al. [140] combined a slack-based MILP model with a periodic RH strategy to update schedules iteratively. Both methods are tested on a virtual semiconductor test line. However, their results have not been compared.

Branke et al. studied the  $F_m|r_j, aux, recrc|\sum F_j$ , in which machines and operators for loading and unloading operations should be scheduled simultaneously [141]. They proposed a simulation-based genetic programming method that evaluates and selects rules based on system status. Experiments showed this method provides better solutions than benchmark rules including SPT and FCFS.

### 5.2.2. Hybrid (Flexible) Flow Shops

Kim et al. studied the  $HF_2|r_j, p - batch, recrc, aux|\sum T_j$  from the burn-in and its related loading/unloading operations [142]. They developed a set of job dispatching rules and loading/unloading rules for the problem.

Bang and Kim studied the  $HF_4|r_j, s_{lk}|\sum T_j$  [143]. A bottleneck-focused scheduling method was employed, in which the bottleneck station is scheduled based on a progress-based scheduling heuristic, and the upstream/downstream workstations are scheduled based on backward and forward list scheduling algorithms. Furthermore, this method was integrated into an RH framework to deal with dynamic job arrivals.

As machines' processing times can be considerably different, Jung et al. found it difficult to define time window length in the RH framework [144]. Instead of periodic RH, they limited the number of runs of each machine and developed a MILP-based method within each period.

Dugardin et al. [145] developed a Lorenz non-dominated sorting genetic algorithm (L-NSGA) based on Lorenz dominance properties for the  $HF_c|r_j, recrc|\#(\sum C_j, machine\ utilization)$ . Gong et al. proposed an opposition-based symbiotic organisms search with a catastrophe phase algorithm (OBSOS-CA) for a scheduling problem related to in-line stepper workstations [146].

Since solutions generated by metaheuristics are usually evaluated by simulation for stochastic and dynamic scheduling problems, [147,148] integrated the OCBA technique into simulation so as to reduce long simulation replications.

Instead of finding a single-sophisticated dispatching rule, Kim et al. proposed a per-machine dispatching rule learning approach that generates a specific linear dispatching rule for each machine based on the gradient evolutionary strategy [149].

In a flow shops environment, the reentrant flows and processing time windows have also been considered. When the shop environment and processing characteristics becoming more complex, we see more papers rely on simulation to model the system instead of mathematical programming.

## 6. Job Shops

An m-stage job shop consists of m different machines where other than flow shops each job has its own predefined processing route. Flexible job shops indicate more than one machine exists in at least one stage. Sometimes, flexible job shops with complex processing constraints are referred to as complex flexible job shops. Since most job shops from SMOs are complex flexible job shops, we omit the related shop environment of each problem. Tables A8 and A9 present deterministic and stochastic job shop scheduling problems respectively.

### 6.1. Deterministic Scheduling

The shifting bottleneck heuristic (SBH) with disjunctive graph representation is an effective method for job shop scheduling problems. The main idea of SBH is to iteratively solve the most critical stage of unscheduled stages by utilizing specific subproblem solution procedures (SSPs) until all stages are scheduled [150]. The SBH has also been applied to SMOs' scheduling problems where new adaptations were developed to tackle batch machines, auxiliary resources, and reentrant flow constraints. Jampani and Mason studied the  $FJ_c|r_o, p - batch, recrc, moj(lot)|\sum w_o C_o$  [151]. They modeled the problem as a disjunctive network flow MILP formulation that was tackled by the column generation (CG) method. Experiments on the mini-fab model showed the CG method provides a computational advantage, especially in the case of equal job releasing times. Knopp et al. examined a resource-dependent flexible job shop scheduling problem, i.e., the  $FJ_c|r_j, aux, recrc|C_{max}$  [152]. They combined the disjunctive graph representation with route graphs to model resource dependencies. Then, they utilized a constructive heuristic to solve SSPs, and applied an SA to improve initial solutions. Knopp et al. studied the  $FJ_c|r_j, s_{lk}, p - batch, B, recrc|f_j(C_j)$  based on a batch-oblivious conjunctive graph that encodes batch decisions into edge weights [153]. They used a GRASP to tackle the resequencing and reassigning decisions, which were then improved by the local search and SA methods. Experiments on industrial-scale instances showed this method performs much better than rule-based methods. Tamssouet et al. [154] generalized the method of [153] by considering both batch machines and machine processing qualifications. They integrated the disjunctive graph representation into a batch-oblivious graph and a routing graph. In addition, Yugma et al. studied a multi-criteria flexible job shop scheduling problem [155]. They used a disjunctive graph representation, in which a priority-rule-based insertion algorithm (PBIA) was used to generate initial solutions that are then improved by iterative sampling and SA algorithms.

Flexible job shop scheduling problems considering multi-resources, i.e.,  $FJ_c|s_{lk}, M_j, aux|C_{max}$  are studied in [156–159]. Ref. [156] developed a hybrid estimation of distribution algorithm (HEDA), Ref. [157] proposed a knowledge-based multi-agent evolutionary algorithm (KMEA), and Ref. [158] designed a cooperative coevolutionary invasive weed optimization (IWO) algorithm that generates machine assignment and operation sequences simultaneously. Experiments showed that IWO outperforms HEDA and KEMA in relation to solution quality and computational time. In addition, Ref. [159] designed a cuckoo search algorithm combined with reinforcement learning and surrogate model-

ing, which was shown to outperform KMEA. However, the results of [158,159] have not been compared.

Chou et al. [160] studied the  $FJ_c|s_{lk}, M_j, incompatible, prec|\#(C_{\max}, TSC, \sum w_j U_j)$ . They proposed a multi-objective hybrid GA (MO-HGA) that hybridizes with a VND algorithm as a local search method and a TOPSIS-based technique to derive the best non-dominated solution. Chung et al. [161] studied the  $FJ_c|s_{lk}, prec|F_l(machin utilization, TSC)$ . They proposed a GA-based framework that includes a GA optimizer to generate initial job sequences, a simulator to obtain and evaluate feasible schedules, and a genetic operator recommender to accelerate convergence by selecting preferred operators.

A related flexible job shop scheduling problem from the die attach and wire bonding operations in the semiconductor packaging line is studied in [162,163]. By introducing state, action, and reward representations based on job and machine information, Park et al. [162] proposed a Q-learning method in which each agent determines setup decisions in a decentralized manner and learns a centralized policy by sharing a neural network among agents, while Park and Park [163] designed a deep reinforcement learning (DRL) approach in which a global agent determines job-machine pair in a centralized manner.

Wu et al. [164] examined a job shop scheduling problem from diffusion operations. Based on a DP formulation, they first identified several structural properties to reduce the solution space, and then applied a GA to rapidly generate near-optimal solutions. Experiments on practical production lines indicated the method leads to improved throughput and scheduling efficiency.

## 6.2. Stochastic and Dynamic Scheduling

Refs. [165,166] constructed nonlinear fluctuation smoothing rules to reduce the average and standard deviation of cycle times. Ref. [165] used fuzzy c-means (FCM) and fuzzy back propagation network (FBPN) to estimate job remaining cycle time, while Ref. [166] combined the HGA with FCM and FBPN to improve estimation accuracy. Ref. [167] used classification and regression trees to predict job remaining cycle time. The method was then combined with load balancing and job emergency rate to build a relevant list scheduling method. Ref. [168] used a deep neural network (DNN) to predict shop status, based on which machine allocation in consideration of limited AMHS capacity is carried out.

Some studies proposed rules to optimize jobs' on-time delivery. Lee and Kim studied a stepper scheduling problem based on an accumulated urgency index (AUI), which considers job delayed and advanced time as well as the WIP volume [169]. Chiang and Fu proposed an enhanced critical ratio (ECR3) rule based on job urgency and the due date extension procedure [170]. Experiments showed the ECR3 outperforms benchmark rules such as EDD and ATCSR. Yao et al. [171,172] considered multi-criteria-based rules. Ref. [171] applied the TOPSIS method to evaluate the performance of on-time delivery, mean and standard deviation of cycle time, and machine utilization. Ref. [172] developed a decentralized multi-objective method to control the workload on each workstation.

Rules for specific processing constraints have also been developed. Bard et al. [173] and Jia et al. [174] built rules to overcome frequent setups in assembly and test operations. Chung et al. [175] suggested a dedication load-based dispatching rule so that jobs could use the same machine in reentrant photolithography operations. Cui and Li [176] proposed a load balancing-based rule to distribute machine workloads.

Furthermore, Refs. [177,178] applied adaptive dispatching rules (ADR) to accommodate system status. Li et al. considered an ADR based on the due date, workload, and job occupation time [177]. The parameters of the ADR were dynamically adjusted by a backward propagation neural network (BPNN) and a PSO algorithm. In contrast to fixed performance measures, Li and Min applied the feature selection method to adapt the rule based on system information, and then used a linear regression method to adjust related parameters [178]. Lee et al. used a combination of multiple dispatching rules to schedule jobs in the photolithography and decomposition operations, in which the weights

of dispatching rules are adjusted by a sequential search method [179]. Yu et al. proposed a dynamic dispatching rule based on self-organization [180].

Maintaining a pool of rules while selecting the most suitable one for each workstation based on shop status could improve system performance. Two kinds of techniques have been utilized. One is the machine learning-based selection mechanism, which relies on a set of training examples carried out beforehand. For instance, Shiue et al. developed an intelligent multi-controller that includes a simulation model to generate training data, a data preprocessing mechanism to provide feature selection, and a self-organizing map (SOM)-based rule for real-time scheduling [181]. Ma et al. used an extreme learning machine algorithm to select appropriate rules [182]. Lim et al. applied the case-based reasoning method to implicitly extract rules for new problems based on similar cases from a case-base built from previously solved problems [183]. Chan et al. used a set of features to represent shop situations and applied a machine learning model to predict the best dispatching rules in multiple online simulation runs [184]. Similarly, Shiue et al. used the reinforcement learning method to dynamically select the appropriate dispatching rules based on the current shop information [185]. The other method is simulation optimization, which combines optimization procedures elaborately within the simulation routine [186]. Chang et al. proposed a GA-based simulation optimization method, which includes an on-line scheduler to monitor combined rules for lot dispatching, batch dispatching, and AGV dispatching, and an offline scheduler to search for the best-combined rules [187]. Kuck et al. proposed an adaptive simulation optimization approach. The optimization procedure used a GA, whose initial solutions also consider the best solutions to the problem before system changes [188]. Ghasemi et al. developed a related simulation optimization model that consists of a learning procedure using a genetic programming (GP) metamodel and an evolutionary optimization structure embedded within ordinal optimization to solve stochastic job shop scheduling problems [189]. Lin et al. designed a learning-based grey wolf optimizer to address a stochastic flexible job shop scheduling problem. Within the method, an OCBA procedure is used to reduce simulation samples and a reinforcement learning algorithm is used to dynamically adjust the parameters of the metaheuristic [190]. Other AI-based methods, such as convolutional neural network and asynchronous advanced actor critic-based method (CNN-A3C) [191], case-based reasoning [192], and DQL [193], have also been utilized recently.

Some studies have utilized closed-loop control strategies for production release and dispatching in order to control job flows. For example, Yoon and Kim proposed a strategy that uses constant work-in-progress (CONWIP) for job releasing, based on which the operations due date was derived for job sequencing so as to reduce variations in cycle times [194]. By predicting the average queuing time of the target WIP distribution, Li et al. proposed a so-called pull virtual production lines-based effective workload control for production release and a queuing time ratio (QTR) rule for job dispatching [195]. Experiments that were carried out on the Minifab model showed the method outperforms the open-loop release policy and common dispatching rules. Moreover, to accurately reflect system workload, Singh and Mathirajan proposed a job release control method to maintain shop floor workload distributed into stages at a specific level [196]. Qiao et al. proposed a closed-loop adaptive scheduling solution that consists of production data acquisition, dynamic disturbance identification, scheduling strategy adjustment, and schedule scheme generation four phase [197].

In addition, analytical methods have been integrated into the RH framework to solve stochastic and dynamic job shop scheduling problems. For instance, Drießel and Mönch studied the  $FJ_c|p - batch, r_j, s_{lk}, incompatible, recrc|\sum w_j T_j$  by considering the AMHS movement [198]. They integrated the SBH into the RH decomposition framework. They decomposed the whole problem into subproblems for workstations and transport operations, which were then tackled by the VNS method separately. Guo et al. studied a similar problem based on the RH decomposition, where a classified ACO algorithm was used to solve subproblems [199]. Kopanos et al. studied a relevant problem by developing a

hybrid-optimization scheduling tool, which includes iterative MILP or heuristic procedures for sub-problems identified as bottleneck toolsets [200]. Barhebwa-mushamuka et al. developed a global scheduling approach that aims at meeting product cycle time targets, based on which production quantities for each product at each operation could be derived at each scheduling horizon [201].

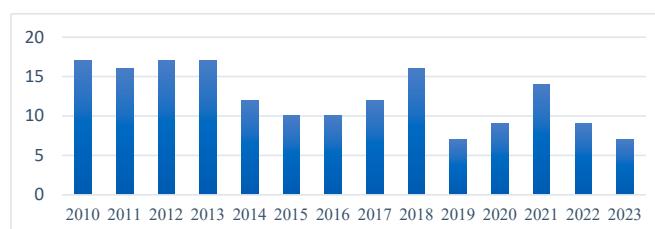
Furthermore, Refs. [202,203] applied event-driven schedule repair methods. Qiao et al. presented a partial repair rescheduling method, whose key is to detect a match-up point so that the segment of the original schedule to be repaired can be determined [202]. Zhong et al. developed an operation-group-based soft scheduling approach, in which the critical scheduling decisions were preset offline while the remaining decisions were determined online during the execution of the initial soft schedule [203]. Zhong et al. designed a job-priority-based soft scheduling approach to tackle stochastic job arrival and processing times in the oxidation/diffusion operation [204]. The approach involves two layers, in which the first offline layer uses a harmony search algorithm to generate initial soft schedules while the second online layer uses rule-based heuristics to make remaining decisions. Experiments indicate results outperform the methods in [202,205].

## 7. Discussion and Future Research Perspectives

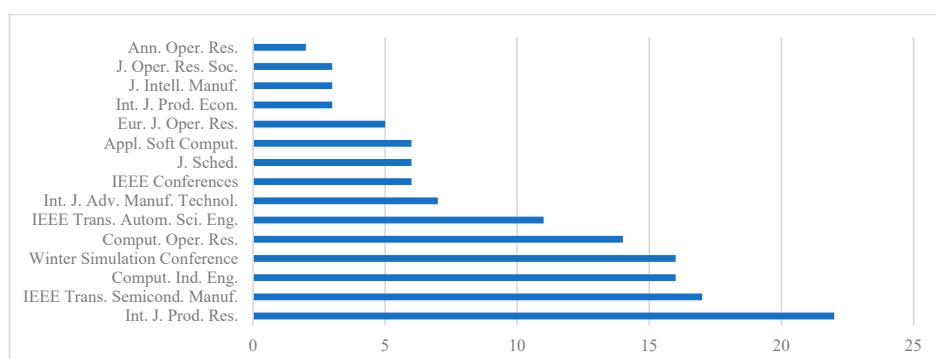
### 7.1. Current Research Progress

This survey includes more than 170 scheduling problems associated with semiconductor manufacturing processes, which appear in major academic journals and conferences. We do not claim to have documented every paper in this domain. However, this survey has covered a large percentage of the most relevant papers, based on which some observations on research progress could be traced.

First, it is noted that the average number of papers published each year stays around 13; see from Figure 1. Although there was a declining trend around 2015, the number increased subsequently, especially in recent years. Figure 2 below presents the distribution of publication places with more than three papers published. This figure indicates that SMOs' scheduling problems are covered in a variety of journals and conferences, mostly in the fields of engineering, computer science, automation, operations research, and management science.

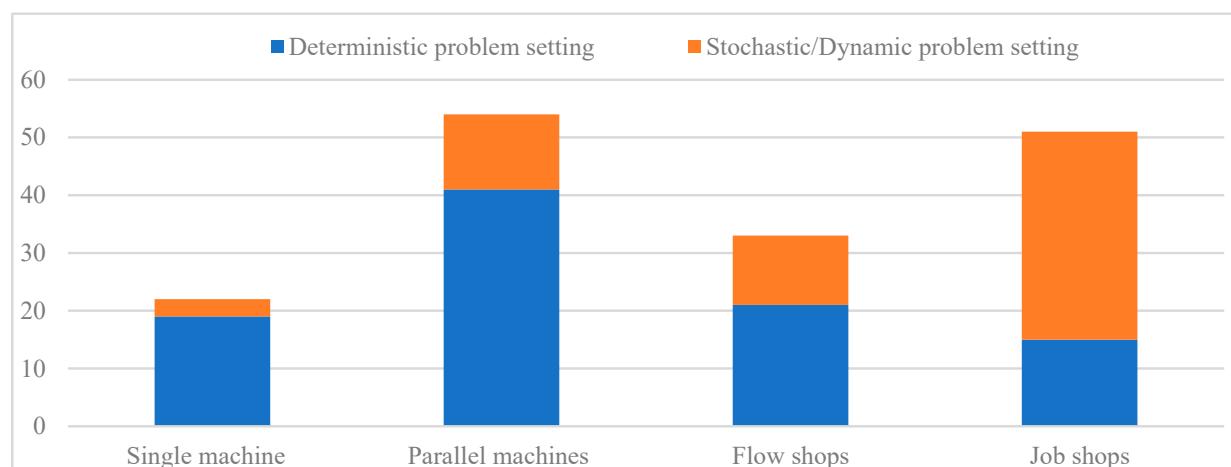


**Figure 1.** Number of relevant publications per year.



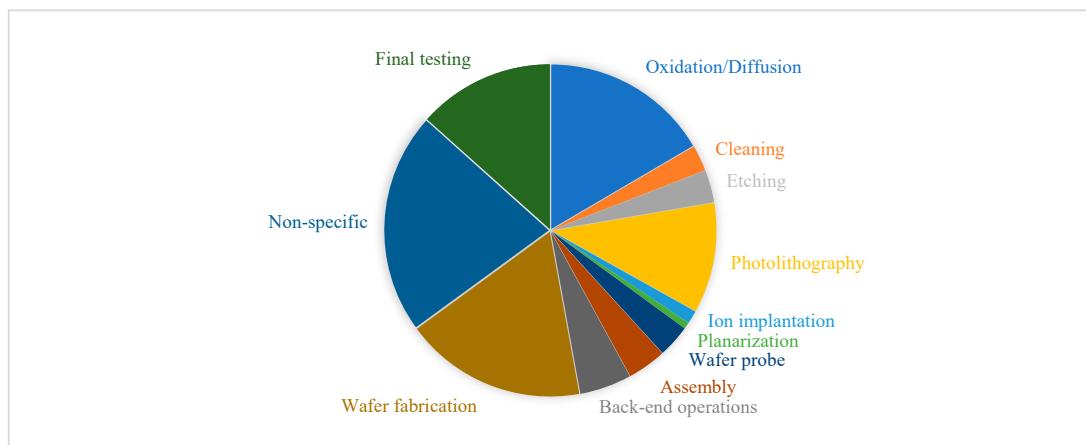
**Figure 2.** Distribution of publication places.

Second, regarding problem classification, Figure 3 presents problem types based on shop environment and problem settings. We classify related scheduling problems into deterministic and stochastic/dynamic settings. From the figure, it is found that more than 60% of the papers dealt with single-stage problems in the deterministic problem setting. Although multi-stage deterministic problems are tackled less, more papers have appeared in recent years, while more than 70% of papers tackled multi-stage problems in the stochastic/dynamic setting. This phenomenon is expected, since analytical methods such as MILP models are more applicable to single-stage scheduling problems, while simulation methods are less intractable to tackle multi-stage systems. Moreover, many studies attempted to examine rules-based methods in flow shops and job shops.



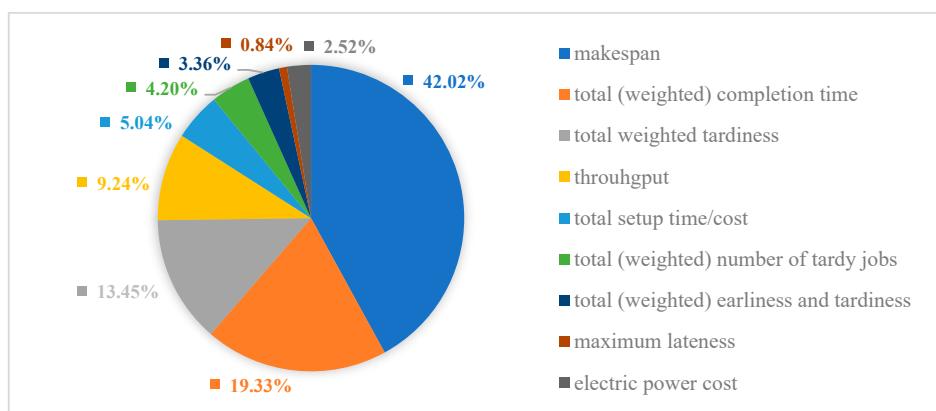
**Figure 3.** Classification of scheduling problems.

Third, through the distribution of considered operations from Figure 4, we find some SMOs' scheduling problems have been investigated more frequently than others. This actually relates to corresponding processing characteristics from each work center, which are studied at a different rate. For example, scheduling problems from diffusion furnaces in oxidation/diffusion operations and burn-in ovens in final testing operations usually investigate batch machine scheduling issues, either with compatible or incompatible job families, and equal or unequal job release times. It could be found that more than 40% of papers study batch machine scheduling problems in the single-stage shop environment. Scheduling problems from photolithography and wafer probes usually consider how to schedule machines and auxiliary resources simultaneously. Around 50% of the papers investigate auxiliary resource constraints in non-batch parallel machine scheduling problems. In these problems, better coordination between machines and resources is required to reduce cycle time. Time windows from cleaning operations, sequence-dependent setups from ion implantation and assembly operations, and MOJ from wafer fabrication have also been investigated. Some studies integrated APC and PM requirements into related scheduling decisions, where appropriate job schedules and test/maintenance schedules should be maintained at the same time. In addition, many papers that indicated their problems are from semiconductor manufacturing but did not mention the exact operation are put in the "non-specific" category. Those papers are usually more methodology-oriented. It also means semiconductor manufacturing is serving as a rich resource for many scheduling problems.



**Figure 4.** Distribution of scheduling problems from different operations.

Regarding performance measures, Figure 5 presents the percentage of examined performance measures in deterministic scheduling context. For stochastic and dynamic problem settings, the performance measures have a larger variation, e.g., throughput, cycle time, waiting time, movement time, and machine idle time, etc., which could be easily evaluated through simulation. From Figure 5, it is shown that  $C_{\max}$  has attracted the most interest, which is followed by  $\sum (w_j)C_j$ ,  $\sum (w_j)T_j$  and throughput.  $C_{\max}$  is used frequently, because it indicates machine utilization and relates directly to throughput and yield performance.  $\sum (w_j)C_j$  is important because it helps measure the WIP information on shop floors, while  $\sum (w_j)T_j$  relates to the on-time delivery performance. Since the increasing competition among semiconductor manufacturing companies, on-time delivery has become more important. Therefore, due date-related measures should be investigated more often. Measures such as total setup cost and  $\sum (w_j)ET_j$  has also been examined. Recently, [67,72,73] tried to consider the electric power cost because of the initiative of gas emission reduction, which actually relates to machine utilization rate. Moreover, numerous papers considered multiple measures in different ways. Sometimes, it is natural to consider multiple measures in related problems. For example, in addition to the above measures, the total number of machine disqualifications has been considered specifically in the APC scheduling problems, and the total number of testers used has been considered in some final testing scheduling problems. While an increasing number of papers have investigated the trade-off effect among commonly used measures recently, such as  $C_{\max}$ ,  $\sum (w_j)C_j$  and  $\sum (w_j)T_j$ . Moreover, using simulation models to compare different measures has been widely applied in stochastic/dynamic scheduling problems.



**Figure 5.** Distribution of studied performance measures (deterministic scheduling).

## 7.2. Existing Solution Methods

Solution methods for machine scheduling problems can be generally classified into analytical methods, heuristics, metaheuristics, and simulation methods. This classification is not exclusive, because hybrid methods can combine multiple solution techniques from these categories together. Since methods applied to deterministic and stochastic/dynamic scheduling problems are largely different, we separate their discussions below.

### 7.2.1. Deterministic Scheduling Problems

MILP models have been extensively used in deterministic SMOs scheduling problems. It is natural to formulate related scheduling problems as MILP models because many binary variables are involved. On one hand, MILP models can provide optimal solutions for small-scale instances, based on which researchers may characterize optimal schedules properties and then develop effective heuristics. On the other hand, MILP models can be used to provide upper bounds for large-scale instances where optimal solutions cannot be obtained or to provide lower bounds by relaxing several constraints.

Some theoretical findings of deterministic scheduling problems are summarized in Table A10 in the Appendix C. These theoretical findings can either prove the properties of optimal schedules, provided dominance properties of partial schedules, or prove the worst-case performance of relevant heuristics. Effective lower bounds can be designed to evaluate the performances of proposed algorithms. For example, Chen et al. [62] constructed two lower bounds by allowing jobs to be split and processed in different batches for the  $P_m|p - batch, r_j, a_j, B|C_{max}$ . Xu et al. [42] proposed a lower bound by considering the relationship between the latest job and other jobs for the  $1|p - batch, r_j, a_j, B|C_{max}$ . Based on the release time and job size of the latest job and other jobs, they derived a formulation for the maximum completion time of the latest job. Chung et al. proposed a lower bound for the  $1|cleaning|\sum C_j$  with irregular maintenance activities by introducing a dummy schedule [55]. It is worth noting that the designing of effective lower bounds depends mainly on the understanding of problem structures.

In addition, lower bounds have been applied in the B&B technique extensively. By integrating with upper bounds from heuristics, and fathoming criteria from partial schedules or dominance properties, B&B can converge to the optimal solution faster. Chung et al. applied the B&B to tackle the  $P_m|r_{batch}|\sum C_j$  [82]. Their proposed heuristic serves as the upper bound. Two lower bounds are used, which are based on equal batch arrival time and treating each batch individually. Multiple dominance properties serve as fathoming criteria, although experiments showed the effectiveness of those fathoming criteria is very different. Sarin et al. [49] and Kim and Lee [120] also applied the B&B to investigate the  $1|moj(item)|\sum C_o$  and the  $F_3|tw_{j1}, tw_{j2}|C_{max}$ , respectively. Though the underlying solution procedures are similar, how to develop effective upper bound, lower bound, and fathoming criteria relies on the exploitation of the optimal or near-optimal solutions. Before designing new bounds, it is usually beneficial to obtain insights from similar problems in past studies. Relatively few studies have applied valid inequalities to strengthen the lower bound of related MILP formulations. One example is from Dauzère-Pérès and Mönch for the  $1|p - batch, incompatible, a_j, B|\sum (w_j)U_j$ , in which three valid inequalities were proposed based on the observations of optimal schedules including the number of batches per family and job sequencing decisions [44].

The other exact solution technique is the DP. Shao and Chen tackled the  $1|p - batch, r_j, a_j, B|C_{max}$  with a hybrid method that uses an ACO to generate job sequences and a DP to form batches [40]. Yao et al. proposed DP algorithms for the  $F_2|r_j, p - batch \rightarrow discrete, B, tw_j, (incompatible)|C_{max}$  and proved their related time complexity [119]. Wu et al. developed a DP algorithm to tackle a job shop scheduling problem [164]. Moreover, the efficiency of the DP was improved based on several dominance properties that reduced the solution space. The successful applications of DP rely on the recognition of state variables and state transition equations, as well as their

connections with the objective function. Though DP is useful for getting optimal solutions, it is usually applied in small-instance problems because of the large memory and computational requirement.

SMOs scheduling problems generally involve a lot of constraints, especially for the auxiliary resources and APC processing. Recently, some researchers have incorporated the CP technique to solve relevant scheduling problems [81,83,96]. The concept of CP is to satisfy constraints by using deductive reasoning to reduce the solution space. CP usually can be faster than MILP models to find feasible solutions but may take some time to obtain the optimal solution which depends on the number of constraints. CP can also be combined with other heuristics to improve search efficiency.

Metaheuristics that are capable of providing high-quality solutions within a reasonable amount of time have been widely used to solve SMO scheduling problems. Table A11 in Appendix C presents the studies that have applied related metaheuristics. Studies from both problem settings are aggregated in the same table because their application procedures are similar. In general, metaheuristics can be classified into single solution-based methods and population-based methods, based on whether a single solution or a family of solutions is generated and updated during the implementation. Single solution-based metaheuristics that are adopted in SMO scheduling problems include VNS, SA, TS, and GRASP, which are distinguished by their solution exploration procedures and updating schemes. TS maintains a tabu list that forbids solutions in the list from being visited again; SA uses the strategy that worse solutions can be selected with certain probabilities; VNS explores solution space by following the steps of shaking, local searching, and moving; GRASP uses a randomized greedy heuristic to construct new solutions. Population-based metaheuristics, including GA, MA, ACO, and PSO, have been used in SMO scheduling problems. Furthermore, population-based metaheuristics can be divided into two subclasses: those that use evolutionary computation to modify a population of solutions through recombination and mutation operators, e.g., GA and MA, and those that use swarm intelligence to exploit a simple analogy of social interaction, e.g., ACO and PSO.

From Table A11, we find GA has been used the most often, which is followed by VNS, SA, GRASP, ACO, and PSO. MA and TS are used less frequently. The performance of these metaheuristics depends on their capability to explore and exploit neighborhood solutions as well as problem characteristics. For instance, ACO is applied frequently in batch scheduling problems, as ant pheromone information can be used to find similar jobs to form a batch. Indeed, more than 50% of papers that utilized the ACO algorithm tackled batch machine scheduling problems. Although some papers seem to use the same metaheuristics, they are quite different because the definition of neighborhood structures and updating schemes are not the same. For example, although both [48,50] applied grouping GA to deal with single-machine MOJ scheduling problems, the crossover, mutation, reinsertion, and mutation procedures are largely not the same. As experiments in [50] showed, the mutation procedure of [48] for tackling  $\sum w_j C_o$  is not effective for tackling  $\sum w_j T_o$ . Hence, to escape from local optima, they designed a distinct mutation procedure. Furthermore, it is known that combining metaheuristics with local search methods provides better solutions. Some metaheuristics, e.g., GA, SA, and TS, require initial solutions, which can be generated from list scheduling methods [37,89] or dispatching rules [62,198]. For example, batch formation decisions can utilize bin packing-related heuristics, and EDD can be used for due date-related measures problems.

### 7.2.2. Stochastic/Dynamic Scheduling Problems

Different from deterministic scheduling problems where mathematical models are used to describe corresponding scheduling problems, simulation has been applied widely in stochastic and dynamic scheduling problems. On one hand, it is difficult for mathematical models to capture the stochastic nature and complicated processing constraints of dynamic scheduling problems. However, these issues can be tackled by simulation programs [206]. On the other hand, simulation models help evaluate different scheduling

solutions. For more details on how to build simulation models for wafer fabrication processes, readers may refer to [207]. Dynamic scheduling methods can be generally classified into completely reactive scheduling, predictive-reactive scheduling, and robust proactive scheduling [208]. Most papers that are examined used completely reactive scheduling and predictive reactive scheduling.

In completely reactive scheduling, no firm schedules are generated in advance, and decisions are made locally in real-time. Priority-based rules are widely used, which select the next job with the highest priority from a set of jobs when a machine becomes available. Job priority is usually determined by job and machine attributes. Though rules that consider only local information are myopic, they have been frequently used in SMOs scheduling problems because of their relatively easy and fast implementation. No single rule consistently outperforms other rules in all situations. Therefore, to enhance the performance of rule-based methods, studies have specifically focused on the following aspects: (1) proposing new rules based on related processing conditions; (2) designing adaptive rules whose parameters can be dynamically adjusted based on system status; (3) maintaining multiple rules while selecting the best one for each machine whenever new events occur; and (4) forming a closed loop control strategy for production release and dispatching; see Table A12 in the Appendix C. Recent literature shows that rule-based methods have evolved from single rules to adaptive rules and multi-rule methods, which modify or change the rule based on system status, while job release strategies have also been integrated into dispatching rules to form a closed-loop control.

Predictive-reactive scheduling is a scheduling/rescheduling process that revises previous schedules in response to real-time events. Two important issues arise in the rescheduling process, i.e., how to reschedule and when to reschedule. Regarding how to reschedule, schedule repair and complete rescheduling can be used. Schedule repair builds a new schedule based on the simple adjustment of the previous schedule, while complete rescheduling constructs a new schedule from scratch. Schedule repair is commonly used in practice because it reduces computational complexity and preserves schedule stability. Similar to deterministic problems, metaheuristics can be utilized to obtain new schedules [116,199]. Regarding when to reschedule, periodic, event-driven, and hybrid methods can be used. In periodic rescheduling, time decomposition or the RH method is applied where rescheduling takes place at the beginning of each time interval, while in the event-driven method rescheduling takes place only when the system's status changes. Within this framework, dynamic scheduling problems can be decomposed into multiple static scheduling problems that can be solved iteratively. The length of the schedule span determines the effectiveness of these methods. Most papers used deterministic periods, while [144] examined the RH method by limiting the number of runs for each machine, and [113] used a variable time interval-based RH strategy. In addition, few papers applied event-driven rescheduling methods, where rescheduling takes place only when unexpected events change system status [202–204].

In addition, simulation optimization that combines optimization procedures in simulation routines can enhance the performance of simulation models [186]. Many studies have utilized the simulation optimization method to analyze SMO scheduling problems [148,187–190]. For example, [148] developed a simulation-optimization approach for a hybrid flow shop scheduling problem, in which the approach included a simulation model for performance evaluation, an optimization strategy that used a genetic algorithm, and an acceleration technique via an optimal computing budget allocation.

### 7.3. Future Research Perspectives

The discussions in previous sections including solution methodologies and problem composition, i.e., shop environment, processing constraints, and performance measures, make the fact clear that most SMOs' scheduling problems are different. To further the research efforts in this domain, it is expected that two kinds of research could be carried out, one is new-problems-oriented, and the other is new-methods-oriented.

### 7.3.1. New Problems Oriented

Regarding new-problems-oriented research, studies could consider the following ways to promote current research boundaries. It is known that the scheduling system in the fabs serves as the bridge between production planning and process control systems; new scheduling problems should consider the conditions and requirements of these two systems to achieve better coordination.

First, production planning and job releasing strategies need to be integrated into proposed scheduling problems. Only a few papers have tackled this issue. Lee et al. dealt with a daily production planning and scheduling problem from wafer probe operations [209]. They developed a mathematical programming model for production planning that provides planning data for a related scheduling heuristic. Xiao et al. developed an integrated model for a parallel machine capacitated lot-sizing and scheduling problem and applied Lagrangian relaxation to decompose the original problem into a lot-sizing subproblem and a machine-scheduling subproblem [210]. However, more integrated models are still lacking.

Second, in order to improve the production yields, SMOs scheduling problems should consider APC requirements, which can include machine health status, maintenance, and processing qualification. More studies have coped with the scheduling problems in consideration of APC recently [95–99]. Among those, Refs. [95,96] tackled the qualification run issue; Refs. [97,99] tackled the deteriorated machine health status issue; Ref. [98] considered the periodic maintenance issue. More studies are needed to continue this research stream, especially with other APC requirements such as the run-to-run control loop and wafer quality index identified in [27]. In addition, most papers tackling maintenance issues assumed static machine health status that requires machines to be stopped for maintenance after a certain amount of processing times or processed jobs. However, recent studies showed dynamic (deteriorated) machine health status is more practical in preventing yield loss [29,97].

Third, due to the increased order-mix and reduced order size, MOJ and interfering job set constraints become common situations in fabs. Currently, most MOJ scheduling problems considered the single-machine environment [47–50]. Parallel machines [52] and multi-stage shop environments [118] are less investigated. Since FOUPs usually carry the same jobs through fabs, it is desirable to study multi-stage scheduling problems with the MOJ. Moreover, interfering job sets with private performances is studied in [56]. Since the pilot line can be integrated into the production line, and customer orders can have distinct performance indicators, which compete for the same processing capability, it is interesting to investigate scheduling problems with interfering job sets and performances.

Fourth, SMOs' scheduling problems should be integrated with the AMHS scheduling. Most studies have treated AMHS scheduling problems independently, although they are one important branch of scheduling problems in fabs. Only few papers attempted to integrate machine scheduling into AMHS scheduling [198]. Since modern fabs are fully automated, it is interesting to combine AMHS with machine scheduling to create a more coordinated scheduling system.

Concerning performance measures, on one hand, new problems could be designed by generalizing previous performances, such as extending  $\sum C_j$  and  $\sum T_j$  to  $\sum w_j C_j$  and  $\sum w_j T_j$ . It is worth noting that a heuristic that performs well on one performance is not necessarily going to have good results on a more generalized performance. On the other hand, performances related to sustainable development have been examined more often recently. For example, the electrical cost measure has been studied recently in [67,72,73] in diffusion operations, because of its high energy consumption. Machine scheduling under carbon reduction policies is examined in [211]. Problems could also be examined more based on multiple measures.

In addition, since unpredicted events such as machine breakdowns, rush orders, and due dates changes are common in practice, it is desirable to investigate more stochastic/dynamic SMOs scheduling problems, especially for the single-stage environment,

in order to obtain more insights into the related scheduling problem. As [212] indicated, stochastic scheduling policies can give counterintuitive results when compared to their deterministic counterparts because idling machines can sometimes benefit from future information.

### 7.3.2. New Methodologies Oriented

Methodologies for deterministic and stochastic/dynamic SMOs scheduling problems are discussed below separately.

Regarding deterministic scheduling methods, it is first noticed that MILP formulations have been applied in both single-stage and multi-stage scheduling problems. To accelerate the convergence of optimal solutions, the B&B technique based on partial schedules or dominance properties has been frequently utilized. However, more advance exact solution techniques, such as CG [151], B&C [79], and branch and price, are seldom applied. It is thus interesting to examine the structure of MILP formulations of related scheduling problems and design advanced exact solution algorithms.

Second, metaheuristics such as GA, VNS, and ACO have been widely applied to SMOs scheduling problems. Different metaheuristics have been utilized for the same problem. For example,  $P_m | p - batch, a_j, B, incompatible | \sum w_j T_j$  have been tackled by the GA, ACO, and ALNS [68,69]. Sometimes, the same type of metaheuristics has tackled similar problems, even though the definition of neighborhood structure and solution updating schemes were different. For example, both Refs. [40,41] utilized the ACO to deal with the  $1 | p - batch, r_j, a_j, B | C_{max}$ . Hence, it is suggested that a set of benchmark problems could be established so that the performance of different methods can be compared. In addition, new studies that usually extend former studies by considering extra constraints or more generalized performance measures can be considered as special cases of former studies. However, the comparisons of methods that tackle these problems are mostly neglected.

Third, hybrid approaches that combine metaheuristics with constructive heuristics or analytical methods have been increasingly employed in recent years. The hybrid approaches help tackle SMOs' scheduling problems with complicated constraints, as well as large systems such as flexible job shop problems. For example, Ref. [96] combined the CP with MILP-based heuristics for APC scheduling problems; Ref. [133] combined the VNS with a decomposition heuristic for a two-stage flow shop scheduling problem. Providing better initial solutions or local search methods usually enhances the performance of some metaheuristics, hence it is desirable to investigate those combinations. Moreover, the better coordination of decomposition strategy and subproblem solution procedures of the SBH methods should also be dealt with [113,144].

Regarding stochastic/dynamic scheduling methods, first, past studies are aware that single static rule-based methods may lead to a large loss. Therefore, recent papers attempted to apply adaptive rules [149,190] or multiple rules [182,184] for a single system, by utilizing machine learning-related methods for updating rule parameters or choosing new rules based on system status. Since dispatching rules are fast and easy to implement in fabs, more studies should be conducted in this stream to design more effective methods.

Second, we view that integrating the RH strategy into deterministic scheduling methods is promising for dynamic scheduling problems, which should be examined more. Simulation-based optimization can be utilized for event-driven rescheduling problems, in which simulation is used to track real-time events while optimization is used to update related schedules [186]. Some current fab simulation testbeds can be used to compare different methodologies, such as the Minifab model [150], Kayton's model [213], and MIMAC dataset [214]. Although researchers tend to build their experimental models, it is suggested that more testbed problems should be set up.

Third, it is surprising that all papers used reactive and predictive-reactive methods for stochastic/dynamic SMOs scheduling problems. It is promising to apply robust opti-

mization and stochastic programming methods for robust proactive scheduling, at least for single-stage problems, in order to obtain more insights [207].

Fourth, given that current stochastic/dynamic SMOs scheduling papers only consider shop efficiency performance, future research should also study schedule stability performance, because frequent and vast changes to previous schedules may not be practical.

In addition, regarding the practical application of scheduling methods, practitioners may obtain some insights from the deterministic scheduling algorithms, though most of them focus on the theoretical aspects and thus are hard to implement in practice. Although fabs are commonly referred as complex job shops, decomposing them based on work station or work centers makes the scheduling problems much more tractable, through which scheduling methods proposed for single machine and parallel machines could be leveraged. Recently, some scheduling methods that are proposed in the stochastic and dynamic scheduling settings have been tested in large systems. Most of them are rule-based methods, which are relatively easy to be implemented. For instance, adaptive scheduling solutions that automatically choose the most suitable dispatching rules based on system status have been used in [60,197]. The AI-based technologies, e.g., DQL and case-based reasoning, could also be leveraged to train the system and reduce computational time [191–193].

**Author Contributions:** Conceptualization, J.F. and A.L.; methodology, J.F. and A.L.; writing—original draft preparation, J.F.; writing—review and editing, J.F. and B.C.; supervision, A.L. and B.C.; funding acquisition, J.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Research Development Fund at the Xi'an JiaoTong-Liverpool University [grant number RDF-21-01-035].

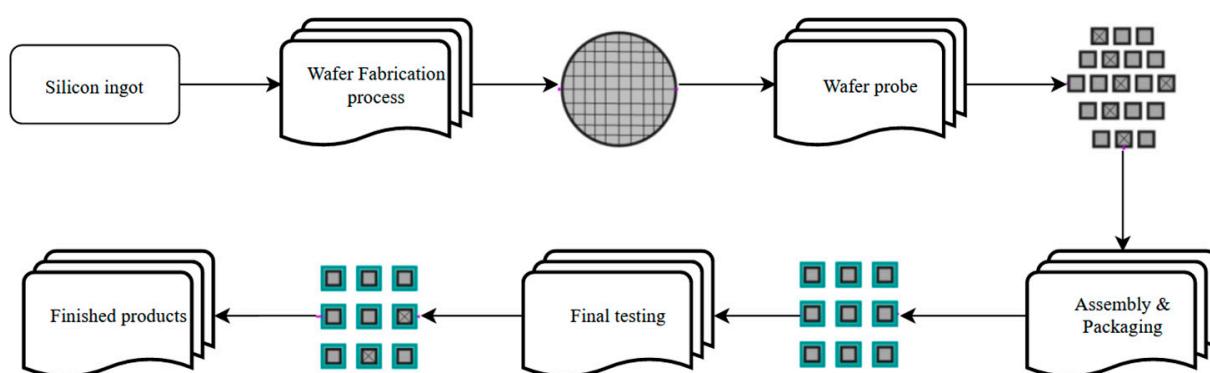
**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

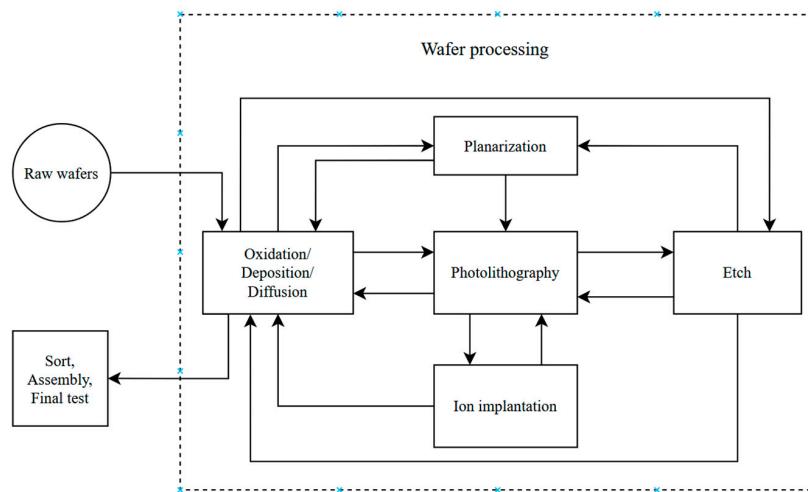
**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A



**Figure A1.** Semiconductor manufacturing stages.



**Figure A2.** Wafer fabrication process flow (adapted from [204]).

## Appendix B

**Table A1.** Deterministic single-batch-machine scheduling problems.

References	Operations	Main Characteristics	Objectives	Approach/Result
[34]	burn-in oven	p-batch, $a_j, B$	$C_{\max}$	WRB, CACB, $O(n^4)$
[36]	burn-in oven	p-batch, $a_j, B$	$C_{\max}$	WRB, decomposition-based MILP heuristics
[37]	cleaning	p-batch, 2D packing, $a_j, B$	$C_{\max}$	MILP, constructive heuristics, BRKGA, HBL
[38]	burn-in oven	p-batch, $a_j, B$	$\sum C_j$	MILP, HMMAS, $O(t_{\max} n_a n^2)$
[40]	burn-in oven	p-batch, $r_j, a_j, B$	$C_{\max}$	ACO, DP
[41]	burn-in oven	p-batch, $r_j, a_j, B$	$C_{\max}$	MILP, WIS, FFWIS-ERT, ACO
[42]	burn-in oven	p-batch, $r_j, a_j, B$	$C_{\max}$	job distances, FRS, MDS, UD
[43]	burn-in oven	p-batch, $r_j, a_j, B$	$\sum w_j T_j$	constructive heuristic, SA
[44]	oxidation /diffusion	p-batch, incompatible, $a_j, B$	$\sum (w_j) U_j$	MILP, lower bound, RKGA
[45]	non-specific	p-batch, incompatible, $a_j, B$	$C_{\max}, \sum C_b$	constructive heuristics, NP-hard, $O(mn \cdot \log(n))$
[46]	non-specific	P-batch, aux, deterioration	$C_{\max}$	VNS, structural properties

**Table A2.** Deterministic single non-batch machine scheduling problems.

References	Operations	Main Characteristics	Objectives	Approach/Result
[47]	non-specific	moj(lot), moj(item)	$\sum C_o$	MILP, constructive heuristics, structure properties, NP-hard
[48]	non-specific	moj(lot), moj(item)	$\sum w_j C_o$	GGA, RKGA
[49]	non-specific	moj(item)	$\sum C_o$	MILP, B&B, structural properties
[50]	non-specific	moj(item), moj(lot)	$\sum w_j T_o, \sum w_j U_o$	MILP, GGAs, NP-hard
[52]	non-specific	moj(lot), incompatible, common due date	$\sum ET_j$	MILP, GA, RKGA, NP-hard
[53]	non-specific	s-batch, $s_{kl}, APC$	$Lex(\sum U_j, C_{\max})$	MILP, structural properties, constructive heuristic, NP-hard
[54]	cleaning	PM, $r_j$	$\#(\sum w_j T_j, \sum C_j)$	MILP, DP, SSA, NP-hard
[55]	cleaning	PM	$\sum C_j$	MILP, heuristic, lower bound
[56]	non-specific	interfering job sets	$\sum C_j (\sum w_j C_j), L_{\max} (\sum w_j C_j)$	negotiation mechanism, VNS

**Table A3.** Deterministic parallel batch machine scheduling problems.

References	Operations	Main Characteristics (Machine Environment)	Objectives	Approach/Result
[62]	burn-in oven	p-batch, $r_j, a_j, B$	$C_{\max}$	lower bounds, ERT-LPT heuristic, GA, ACO
[63]	burn-in oven	p-batch, $r_j, a_j, B$	$C_{\max}$	MILP, DMBHs, GRLPT
[64]	burn-in oven	p-batch, $r_j, a_j, B (R_m)$	$C_{\max}$	MILP, lower bound, constructive heuristics
[65]	burn-in oven	p-batch, $a_j, B_m (R_m)$	$\sum w_j T_j$	MILP, PSO
[66]	non-specific	p-batch, $a_j, B (R_m)$	$\#(C_{\max}, \sum w_j C_j, \sum w_j T_j)$	LP-based heuristics, GA
[67]	non-specific	p-batch, $r_j, a_j, B$	$\#(C_{\max}, \text{EPC})$	MILP, PACO, $O(T_{\max} N_a m n^2)$
[68]	oxidation/ diffusion	p-batch, $a_j, B$ , incompatible	$\sum w_j T_j$	ATC-BATC, ACO, VNS
[69]	oxidation/ diffusion	p-batch, $a_j, B$ , incompatible	$\sum w_j T_j$	MILP, GA, ACO, ALNS
[70]	oxidation/ diffusion	p-batch, $a_j, B$ , incompatible	$\sum f_j(C_j)$	BRKGA, DH
[71]	oxidation/ diffusion	p-batch, $r_j, a_j, B$ , incompatible	$\sum w_j T_j$	MA, GA
[72]	oxidation/ diffusion	p-batch, $r_j, a_j, B$ , incompatible	$\#(\sum w_j C_j, \text{EPC})$	MILP, $\epsilon$ -constraint method, structural properties
[73]	oxidation/ diffusion	p-batch, $r_j, a_j, B$ , incompatible	$\#(\sum w_j T_j, \text{EPC})$	MILP, $\epsilon$ -constraint method, grouping GA, NSGA
[75]	burn-in oven	p-batch, aux, incompatible ( $R_m$ )	backorder, throughput	MILP, LP relaxation, greedy heuristic
[76]	non-specific	p-batch, $a_j, tw_j, M_j$ , incompatible, $B, (R_m)$	$\text{Lex}(C_{\max}, \sum w_j C_j, \sum T_j)$	VND
[77]	oxidation/ diffusion	p-batch, $r_j$ , prec, $a_j, B$ incompatible	$\sum w_j T_j$	VNS, GRASP

**Table A4.** Deterministic parallel non-batch machine scheduling problems.

References	Operations	Main Characteristics (Shop Environment)	Objectives	Approach/Result
[78]	photolithography	aux, $M_j, s_{kl} (R_m)$	$\text{Lex}(\text{throughput}, \text{load balancing}, \text{TSC})$	multi-stage MILP-based decomposition
[79]	photolithography	aux, $s_{kl} (R_m)$	$F_l(\text{load balancing}, \text{reticle expiration}, \text{TSC})$	B&C, two-phase MILP-based heuristic
[80]	photolithography	$s_{kl}$ , aux, $M_j (R_m)$	$\sum w_j C_j$ , throughput	MA
[81]	lithography	$r_j, s_{kl}$ , aux ( $R_m$ )	$C_{\max}$	MILP, CP
[82]	photolithography	aux, incompatible (P2)	$C_{\max}$	MILP, heuristics, worst-case performance
[83]	photolithography	aux, vehicles	$C_{\max}$	CP
[84]	back-end operations	$s_{lk}$ , aux	$F_l(\text{throughput}, \sum w_j U_j)$	MIP, GRASP
[85]	non-specific	aux, $r_j, s_{kl} (R_m)$	$F_l(C_{\max}, \sum U_j)$	MILP, EDDLC
[86]	ion implantation	$r_j, s_{kl}, tw_j (R_m)$	$\text{Lex}(\text{throughput}, C_{\max})$	MILP, hybrid TS
[87]	non-specific	$s_{kl}, M_j (R_m)$	$\text{Lex}(\sum T_j, C_{\max})$	MILP, SA
[88]	final testing	recrc, aux, $s_{kl} (R_m)$	throughput	SARSA

**Table A4.** Cont.

References	Operations	Main Characteristics (Shop Environment)	Objectives	Approach/Result
[89]	wafer probe	$s_{kl}$ , aux	$\text{Lex}(\text{TSC}, \text{testers used})$	ACO, TS, GA
[90]	wafer probe	$s_{kl}$ , aux	$\text{Lex}(\text{TSC}, \text{testers used})$	iterated greedy heuristic
[91]	wafer probe	$s_{kl}$ , aux	$\text{Lex}(\text{TSC}, \text{testers used})$	MILP, HAIS
[92]	photolithography	$r_j, s_{kl}$ , aux ( $R_m$ )	$\sum C_j$	MILP, constructive heuristic
[93]	photolithography	$r_j, s_{kl}$ , aux, PM deterioration ( $R_m$ )	$\text{Lex}(\text{total quality loss}, \sum T_j)$	MILP
[94]	photolithography	$r_j, s_{kl}$ , aux ( $R_m$ )	$\sum w_j C_j$	MILP, GA, VNS
[95]	non-specific	incompatible, $s_{kl}$ , APC ( $R_m$ )	$F_l(\sum C_j, \sum \text{machine disqualifications})$	NP-hard, MILP, SCH, QCH
[96]	non-specific	incompatible, $M_j$ , APC ( $R_m$ )	$F_l(\sum C_j, \sum \text{machine disqualifications})$	MILP, CP, recursive heuristic, SA
[97]	oxidation/diffusion	static/dynamic MHF, $M_j$ ( $R_m$ )	$F_l(\text{throughput, quality risk})$	MILP
[98]	cleaning	$r_j, d_j$ , PM ( $R_m$ )	$C_{\max}$	NP-hard, MILP, Feature-extraction, DP
[99]	non-specific	APC, PM, MHF	$F_l(\sum T_j, \text{quality risk})$	MILP, VNS, B&B
[100]	final testing	batch arrivals	$\sum C_j$	B&B, iterative heuristic
[101]	implantation	$r_j, s_{kl}, tw_j$ ( $R_m$ )	$\in (\sum w_j C_j / \sum \text{exceeded time})$	MINLP, GA, MSPHEDA
[102]	non-specific	Common due window ( $R_m$ )	$\sum w_j ET_j$	MILP, constructive heuristics
[103]	non-specific	moj(item), moj(lot), $d_j$ , incompatible	$\sum ET_j$	MILP, structural properties, BRKGA
[210]	non-specific	$r_j, s_{kl}$ , prec	$\sum w_j T_j$	ATCSR, VNS
[211]	non-specific	carbon emission ( $R_m$ )	$F_l(C_{\max}, \text{production cost})$	NSGA-II
[215]	planarization	s-batch, finite buffer capacity	$\sum C_j$	DP, structural properties
[216]	final testing	aux, $s_{kl}$	$C_{\max}$	SA, TS, GA
[217]	final testing	aux, $s_{kl}$ ( $R_m$ )	$C_{\max}$	MILP, GA

**Table A5.** Stochastic and dynamic parallel machine scheduling problems.

References	Operations	Main Characteristics (Shop Environment)	Objectives	Approach/Result
[104]	oxidation/diffusion	$r_j$ , p-batch, incompatible ( $R_m$ )	$C_{\max}, \sum w_j C_j, \sum w_j T_j$	MILP, time window decomposition
[105]	oxidation/diffusion	$r_j, M_j$ , p-batch, incompatible ( $R_m$ )	$\sum T_j$	VNS, time window decomposition
[106]	oxidation/diffusion	$r_j$ , p-batch, incompatible	$\sum C_j$	MILP, DP, MPC
[107]	photo-lithography	p-batch, recrc, incompatible	$\sum w_j T_j$	batch-oriented and family-oriented scheduling algorithm
[108]	oxidation/diffusion	$r_j, s_{lk}$ , p-batch, incompatible	$\sum T_j$	dispatching rules, look-ahead checks
[109]	oxidation/diffusion	$r_j$ , p-batch, incompatible, recrc	$\sum w_j T_j$	real-time closed loop control
[110]	oxidation/diffusion	$r_j$ , p-batch, incompatible	$F_l(\sum w_j T_j, C_{\max})$	LBADM, adaptive dispatching rule
[111]	photo-lithography	aux, $M_j, s_{kl}$ ( $R_m$ )	$F_l(C_{\max}, L_{\max}, \text{TSC})$	SA, TS, GA
[112]	photo-lithography	$s_{lk}, M_j$ , aux ( $R_m$ )	$C_{\max}, \sum C_j$ , load balancing	MILP, RTD
[113]	photo-lithography	$r_j, M_j$ , aux, recrc ( $R_m$ )	$\sum C_j$	improved ICA, variable time interval-based RH, local search
[114]	photo-lithography	$r_j, s_{lk}, M_j$ , aux ( $R_m$ )	$\sum w_j T_j$	DQN
[115]	non-specific	$r_j, s_{lk}$ , incompatible ( $R_m$ )	$C_{\max}$	DRL, HGA
[116]	non-specific	$p_j, s_{lk}$ ( $R_m$ )	$C_{\max}$	GA, OCBA
[117]	non-specific	$p_j, s_{lk}, M_j$ , PM ( $R_m$ )	$F_l(\sum s_{lk}, \sum T_j)$	DQL

**Table A6.** Deterministic flow shop scheduling problems.

References	Operations	Main Characteristics(Shop Environment)	Objectives	Approach/Result
[118]	non-specific	$r_j, a_j, d_j, \text{moj}$ (item, lot, and batch), (F3)	$\sum w_j T_j$	MILP, GA
[119]	oxidation/ diffusion	$r_j, p - \text{batch} \rightarrow \text{discrete}, B, \text{incompatible}, tw_j$ (F2)	$C_{\max}$	P/NP analysis, DP, TS EDD-FIFO, worst-case performance
[120]	non-specific	$tw_{j1}, tw_{j2}$ , overlapping time window (F3)	$C_{\max}$	MILP, dominance properties, B&B, lower bounds
[121]	wafer probe	$s_{lk}$ , limited buffer capacity, incompatible (Fm)	$C_{\max}$	MILP, GA, TS, NEH
[124]	final testing	$a_j, p - \text{batch}$ (Fm)	$F_l(\sum ET_j, C_{\max})$	MILP, HGA, HSA, PSO, adaptive learning
[125]	wet-etch station	NIS, ZW, LS, single robot (Fm)	$C_{\max}$	MILP
[126]	wet-etch station	NIS, ZW, LS, single robot (Fm)	$C_{\max}$	MILP-based decomposition
[127]	wet-etch station	ZW, LS, NIS, single robot (Fm)	$C_{\max}$	MILP-based decomposition and aggregation heuristic
[128]	wet-etch station	$s_{lk}$ , p-batch, LS, ZW, NIS, single robot (Fm)	$C_{\max}$	GA
[129]	non-specific	$r_j, \text{discrete} \rightarrow p - \text{batch}, M_j$ (HF2)	$C_{\max}$	MILP, BFIFO, local search
[130]	back-end operations	$r_j, p - \text{batch} \rightarrow \text{discrete}, B, tw_j$ (HF2)	$C_{\max}$	MILP, GP, constructive heuristic
[131]	photolithography	aux, limited buffer capacity (HFc)	$C_{\max}$	GA
[132]	back-end operations	$s_{lk}, M_j, \text{PM}, \text{aux}, \text{skip jobs}$ , incompatible (HFFc)	$\sum T_j$	MILP, two-stage heuristic
[133]	oxidation/ diffusion	$r_j, p - \text{batch}, \text{incompatible}$ (HF2)	$\sum w_j T_j$	MILP, NP-hard, stage-based decomposition, VNS
[134]	wafer fabrication and probe	recrc, $s_{lk}$ (HF2)	$C_{\max}$	MILP, RKGA, HGA, constructive heuristics
[135]	back-end operations	$s_{lk}, \text{recrc}, \text{aux}$ (HFc)	$\text{Lex}(\text{device shortages, throughput, utilization}, C_{\max})$	partial MILP, three-phase approach, GRASP
[136]	back-end operations	$s_{lk}, \text{recrc}, \text{aux}$ (HFc)	$\text{Lex}(\text{device shortages, throughput, utilization}, C_{\max})$	three-phase decomposition, MILP
[137]	non-specific	recrc, inventory buffer, stocker (HFc)	$F_l(C_{\max}, \sum F_j)$	HSSGA
[218]	wet-etch station	recrc (HFc)	$\#(\sum T_j, C_{\max})$	self-braking symbiotic organisms search algorithm
[219]	photolithography	$r_j, \text{recrc}, \text{cluster tools}$ (HFc)	$C_{\max}$	MILP, constructive heuristic, GA
[220]	non-specific	$tw_j, \text{skip jobs}$ (F2)	waiting time variation	MILP, dominance properties, constructive heuristic

**Table A7.** Stochastic and dynamic flow shop scheduling problems.

References	Operations	Main Characteristics (Machine Environment)	Objectives	Approach/Result
[138]	diffusion	$r_j$ , p – batch $\rightarrow$ p – batch, $tw_j$ (F2)	$C_{\max}$	IP-based RTD heuristic
[139]	non-specific	$r_j$ , p – batch $\rightarrow$ p – batch, $tw_{j'}$ recrc, incompatible (F2)	$C_{\max}$	pull-based scheduling algorithm
[141]	non-specific	$r_j$ , aux, recrc (Fm)	$\sum F_j$	simulation-based genetic programming
[142]	burn-in stations	$r_j$ , p – batch, recrc, aux (HF2)	$\sum T_j$	list scheduling algorithms
[143]	wafer probe	$r_j, s_{lk}$ (HF4)	$\sum T_j$	bottleneck-focused scheduling, progress-based scheduling, RH
[144]	oxidation/ diffusion	$r_j, s_{lk}, tw_j$ , p – batch (HFc)	$F_l(\sum C_j, \text{time window violations})$	MILP, RH
[145]	wafer probe	$r_j$ , recrc (HFc)	$\#(\sum C_j, \text{machine utilization})$	L-NGSA, NSGA2, SPEA2, simulation
[146]	photolithography	$s_{lk}$ (HFc)	$C_{\max}$	OBSOS-CA, simulation
[147]	assembly	$r_j$ , stochastic processing time (HFc)	$\sum w_j F_j$	simulation optimization, PSO, OCBA
[148]	back-end operations	$r_j, s_{lk}$ , demand and supply variations (HFc)	$\sum w_j F_j$	simulation optimization, OCBA, GA
[149]	non-specific	recrc (HFc)	$\sum C_j, \sum T_j$	learning-based dispatching rule

**Table A8.** Deterministic job shops scheduling problems.

References	Operations	Main Characteristics	Objectives	Approach/Result
[151]	wafer fabrication	$r_j$ , moj(lot), p-batch, recrc	$\sum w_o C_o$	MIP, CG heuristic
[152]	oxidation/ diffusion	$r_j$ , aux, recrc	$C_{\max}$	route graph, constructive heuristic, SA
[153]	oxidation/ diffusion	$r_j, s_{lk}$ , p-batch, $B$ , recrc	$f_j(C_j)$	GRASP, batch-oblivious route-aware graph, SA
[154]	oxidation/ diffusion	$s_{lk}, M_j, tw_j$ , p-batch, $B$	$Lex(\sum w_j C_j, \text{throughput, batch coefficient})$	batch-oblivious route-aware graph
[155]	oxidation/ diffusion	$r_j, M_j, tw_j$ , p-batch, $B$ , prec	$F_l(\text{throughput, waiting time, batching coefficient})$	SBH, PBIA, SA
[156]	final testing	$s_{lk}, M_j$ , aux	$C_{\max}$	KMEA
[157]	final testing	$s_{lk}, M_j$ , aux	$C_{\max}$	HEDA
[158]	final testing	$s_{lk}, M_j$ , aux	$C_{\max}$	IWO
[159]	final testing	$s_{lk}, M_j$ , aux	$C_{\max}$	cuckoo search, reinforcement learning
[160]	assembly	$s_{lk}, M_j$ , incompatible, prec	$\#(C_{\max}, TSC, \sum w_j U_j)$	MILP, MO-HGA, VND
[161]	packaging	$s_{lk}$ , prec	$F_l(\text{machine utilization, TSC})$	MILP, GA, simulation
[162]	packaging	recrc, $s_{lk}$	$C_{\max}$	Q-learning
[163]	packaging	recrc, $s_{lk}$	$C_{\max}$	DRL
[164]	oxidation/ diffusion	p-batch, recrc, $r_j, d_j, tw_j, B$	throughput	DP, GA
[205]	non-specific	$s_{lk}$ , prec	$C_{\max}$	hybrid PSO, GA

**Table A9.** Stochastic and dynamic job shop scheduling problems.

References	Operations	Main Characteristics	Objectives	Approach/Result
[165]	non-specific	stochastic processing time	mean and standard deviation of cycle time	FCM, FBPN, nonlinear fluctuation smoothing rule
[166]	wafer fabrication	stochastic processing time	mean and standard deviation of cycle time	nonlinear fluctuation smoothing rule, GA, FBNP
[167]	wafer fabrication	p-batch	cycle time, machine utilization, on-time delivery, throughput	dispatching rule
[168]	wafer fabrication	limited AMHS capacity	utilization, throughput	DNN
[169]	photolitho-graphy	demand variation	order fill rate, inventory, shortage, cycle time	AUI rule
[170]	wafer fabrication	$r_j, s_{lk}$ , p-batch, recrc, machine breakdown	on-time delivery rate, mean tardiness, $L_{\max}$	ECR3 rule
[171]	wafer fabrication	dynamic job arrivals and line balance information	on-time delivery, mean and standard deviation of cycle time, machine utilization	multi-objective dispatching rule, TOPSIS
[172]	wafer fabrication	dynamic job arrivals and line balance information	mean and standard deviation of cycle time, on-time delivery	decentralized multi-objective scheduling method
[173]	back-end operations	$r_j, s_{lk}, M_j$ , recrc, aux	key device shortages, number of machines used, throughput	MILP, GRASP-based dispatching rules, simulation
[174]	back-end operations	$r_j, s_{lk}, M_j$ , recrc, aux	key device shortages, number of machines used, throughput	GRASP-based dispatching rules
[175]	photolitho-graphy	$r_j, M_j$ , aux	machine utilization, on-time delivery	dedication load-based dispatching rules
[176]	wafer fabrication	p-batch, recrc	wafer movement, utilization, throughput	closed-loop control based on load balancing
[177]	wafer fabrication	hot jobs, p-batch	WIP, bottleneck utilization	ADR, BPNN, PSO
[178]	wafer fabrication	hot jobs, p-batch	WIP movement	ADR, linear regression, GA
[179]	wafer fabrication	p-batch, aux, recrc	throughput, TSC	dispatching rules
[180]	wafer fabrication	$r_j, s_{lk}$ , etc.	throughput, cycle time, on-time delivery rate, movement	dynamic dispatching rule
[181]	wafer fabrication	dynamic job arrivals, load balancing	throughput, cycle time	SOM-based multi-rules selection method
[182]	wafer fabrication	$r_j$ , p-batch, aux	throughput, utilization	extreme learning stochastic machine, multiple dispatching rules
[183]	assembly	$r_j, s_{lk}$ , aux, recrc	machine utilization	case-based reasoning, GA
[184]	wafer fabrication	p-batch, recrc	throughput, cycle time	machine learning, simulation, dispatching rules
[185]	wafer fabrication	p-batch, aux, recrc	cycle time	reinforcement learning
[187]	wafer fabrication	$r_j$ , lot processing, p-batch, AMHS	average delay, average WIP, average cycle time	simulation optimization, GA, multiple dispatching rules
[188]	wafer fabrication	$r_j$ , recrc	average cycle time	adaptive simulation-based optimization, GA
[189]	non-specific	$r_j, s_{lk}$ , etc.	cycle time	simulation optimization, genetic programming
[190]	non-specific	$r_j, s_{lk}$ , etc.	$C_{\max}$	learning-based grey wolf optimizer
[191]	wafer fabrication	uncertain processing times, urgent orders etc.	on-time delivery rate	CNN-A3C, DRL
[192]	packaging	aux, recrc etc.	machine loss times, job waiting times	case-based reasoning
[193]	wafer fabrication	p-batch, aux, recrc etc.	throughput	DRL, dispatching rules
[194]	wafer fabrication	$r_j$ , p-batch, recrc	mean and standard deviation of cycle time	closed-loop control, CONWIP, operation due date rule

**Table A9.** Cont.

References	Operations	Main Characteristics	Objectives	Approach/Result
[195]	wafer fabrication	p-batch, machine breakdown	average and standard deviation of cycle time, WIP	closed-loop control, production release strategy, QTR rule
[196]	wafer fabrication	$r_j, s_{lk}$ , etc.	average and standard deviation of cycle time, WIP, throughput	release control
[197]	non-specific	$r_j, s_{lk}$ , etc.	throughput, mean cycle time, flow time, movement	closed-loop adaptive scheduling
[198]	wafer fabrication	$r_j, s_{lk}$ , p-batch, recrc, incompatible, transportation	$\sum w_j T_j$	RH, SBH, VNS
[199]	wafer fabrication	$r_j$ , prec, recrc	$C_{\max}$	RH, SBH, MILP, ACO
[200]	wafer fabrication	p-batch, recrc	$\sum w_j C_j$	hybrid-optimization scheduling
[201]	wafer fabrication	p-batch, aux, recrc	cycle time	global scheduling approach
[202]	wafer fabrication	p-batch, recrc, prec	schedule stability, machine utilization	scheduling repair method
[203]	wafer fabrication	uncertain processing time, machine breakdown	cycle time	operation-group-based soft scheduling approach
[204]	oxidation/ diffusion	p-batch, $r_j, p_j$	$\sum C_j$	job-priority based soft scheduling
[221]	wafer fabrication	lot merging and splitting	$\sum T_j$ , cycle time, throughput	listing scheduling
[222]	wafer fabrication	$r_j$ , recrc	operational due date	two-dimensional dispatching rule, local search, simulation

## Appendix C

**Table A10.** Theoretical findings for some deterministic scheduling problems.

Shop Environment	Reference	Problems	Theoretical Properties
Single machine	[34]	$1 p - batch, a_j, B C_{\max}$	Batch jobs with similar processing time, WRB with $O(n^4)$ complexity >> the BFLPT
	[41]	$1 p - batch, r_j, a_j, B C_{\max}$	Minimize $C_{\max}$ is equivalent to minimizing the WIS
	[42]	$1 p - batch, r_j, a_j, B C_{\max}$	Batch jobs with close processing and release times if there is residual capacity
	[44]	$1 p - batch, incompatible, a_j, B \sum (w_j) U_j$	Optimal schedule indicates all on-time jobs are processed before tardy jobs
	[47]	$1 moj(item) \sum C_o, 1 moj(lot) \sum C_o$	Optimal schedules are proved
	[48]	$1 moj(item) \sum w_o C_o, 1 moj(lot) \sum w_o C_o$	Optimal job sequencing decisions are proved for both cases
	[49]	$1 moj(item) \sum C_o$	B&B with structural properties of smaller lot first sequencing
	[53]	$1 s_{kl}, APC \sum Lex(C_{\max}, U_j)$	Optimal schedule for the two job-family case is proved
	[55]	$1 cleaning \sum C_j$	Lower bound based on a dummy schedule considering jobs and contamination sequencing
Parallel machines	[62]	$P_m p - batch, r_j, a_j, B C_{\max}$	Lower bounds based on job splitting
	[64]	$R_m p - batch, r_j, a_j, B C_{\max}$	Lower bound is provided
	[82]	$P_2 aux C_{\max}$	LPT-based heuristics with worst-case performance ratio of 3/2
	[100]	$P_m r_{batch} \sum C_j$	B&B with dominance properties from partial schedules
Flow shops	[103]	$P_m moj(item, lot), incompatible \sum ET_j$	Structure properties of optimal schedules are proved
	[119]	$F_2 r_j, p - batch \rightarrow discrete, (tw_j), (incompatible) C_{\max}$	Optimal schedules when not considering incompatible jobs, and heuristics with worst-case performance less than 2 is provided
	[120]	$F_3 tw_{j1}, tw_{j2} C_{\max}$	Dominance properties based on partial schedules are provided
	[220]	$F_2 tw_j, skip jobs waiting time variation$	Dominance properties based on partial schedules are provided

**Table A11.** Studies of applying relevant metaheuristics (both problem settings).

Metaheuristics	References
Variable neighborhood search (VNS)	[46,56,68,77,87,94,99,105,133,198,210]
Simulated annealing (SA)	[96,111,124,152,153,155,216]
Tabu search (TS)	[86,89]
Greedy randomized adaptive search procedure (GRASP)	[77,84,107,135,153,173]
Genetic algorithms (GA)	[52,62,66,69,71,73,89,94,111,116,118,121,128,131,134,141,148,157,161,164,166,178,183,187,188,205,211,216,217,219]
Memetic algorithms (MA)	[71,80]
Ant colony optimization (ACO)	[40,41,62,67–69,89,199]
Particle swarm optimization (PSO)	[65,124,147,177,205]

**Table A12.** Studies of applying relevant completely reactive scheduling methods.

Methods	References
New dispatching rules	[138,165,166,169–172,175]
Adaptive rules	[110,149,177,178,180,188,190,191,193]
Maintain multiple rules	[60,182–185]
Closed-loop control	[109,139,176,194–197]

## References

- Chen, W.-K. *VLSI Technology*; CRC Press: Boca Raton, FL, USA, 2003.
- Mönch, L.; Uzsoy, R.; Fowler, J.W. A survey of semiconductor supply chain models part III: Master planning, production planning, and demand fulfilment. *Int. J. Prod. Res.* **2017**, *56*, 4565–4584. [[CrossRef](#)]
- Mönch, L.; Uzsoy, R.; Fowler, J.W. A survey of semiconductor supply chain models part I: Semiconductor supply chains, strategic network design, and supply chain simulation. *Int. J. Prod. Res.* **2018**, *56*, 4524–4545. [[CrossRef](#)]
- Uzsoy, R.; Fowler, J.W.; Mönch, L. A survey of semiconductor supply chain models Part II: Demand planning, inventory management, and capacity planning. *Int. J. Prod. Res.* **2018**, *56*, 4546–4564. [[CrossRef](#)]
- Wang, Q.; Huang, N.; Chen, Z.; Chen, X.; Cai, H.; Wu, Y. Environmental data and facts in the semiconductor manufacturing industry: An unexpected high water and energy consumption situation. *Water Cycle* **2023**, *4*, 47–54. [[CrossRef](#)]
- Eom, Y.-S.; Hong, J.-H.; Lee, S.-J.; Lee, E.-J.; Cha, J.-S.; Lee, D.-G.; Bang, S.-A. Emission Factors of Air Toxics from Semiconductor Manufacturing in Korea. *J. Air Waste Manag. Assoc.* **2006**, *56*, 1518–1524. [[CrossRef](#)] [[PubMed](#)]
- Li, K.; Zhang, X.; Leung, J.Y.-T.; Yang, S.-L. Parallel machine scheduling problems in green manufacturing industry. *J. Manuf. Syst.* **2016**, *38*, 98–106. [[CrossRef](#)]
- Lu, C.; Gao, L.; Li, X.; Zheng, J.; Gong, W. A multi-objective approach to welding shop scheduling for makespan, noise pollution and energy consumption. *J. Clean. Prod.* **2018**, *196*, 773–787. [[CrossRef](#)]
- Pinedo, M.L. *Scheduling: Theory, Algorithms, and Systems*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2008.
- Vidal, J.C.; Mucientes, M.; Bugarín, A.; Lama, M. Machine scheduling in custom furniture industry through neuro-evolutionary hybridization. *Appl. Soft Comput.* **2011**, *11*, 1600–1613. [[CrossRef](#)]
- Cheng, C.-Y.; Chen, T.-L.; Wang, L.-C.; Chen, Y.-Y. A genetic algorithm for the multi-stage and parallel-machine scheduling problem with job splitting—A case study for the solar cell industry. *Int. J. Prod. Res.* **2013**, *51*, 4755–4777. [[CrossRef](#)]
- Schulze, M.; Rieck, J.; Seifi, C.; Zimmermann, J. Machine scheduling in underground mining: An application in the potash industry. *OR Spectr.* **2016**, *38*, 365–403. [[CrossRef](#)]
- Pfund, M.E.; Mason, S.J.; Fowler, J.W. Semiconductor manufacuring scheduling and dispatching, State of the art and survey of needs. In *Handbook of Production Scheduling*; Springer: Boston, MA, USA, 2006; pp. 213–241.
- Chien, C.F.; Peres, S.D.; Ehm, H.; Fowler, J.W.; Jiang, Z.; Krishnaswamy, S.; Lee, T.E.; Monch, L.; Uzsoy, R. Modelling and analysis of semiconductor manufacturing in a shrinking world: Challenges and successes. *Eur. J. Ind. Eng.* **2011**, *5*, 254. [[CrossRef](#)]
- Uzsoy, R.; Lee, C.-Y.; Martin-Vega, L.A. A review of production planning and scheduling models in semiconductor industry Part I: System characteristics, performance evaluation and production planning. *IIE Trans.* **1992**, *24*, 47–60. [[CrossRef](#)]
- Uzsoy, R.; Lee, C.-Y.; Martin-Vega, L.A. A review of production planning and scheduling models in the semiconductor industry Part II: Shop-Floor Control. *IIE Trans.* **1994**, *26*, 44–55. [[CrossRef](#)]
- Mathirajan, M.; Sivakumar, A. A literature review, classification and simple meta-analysis on scheduling of batch processors in semiconductor. *Int. J. Adv. Manuf. Technol.* **2006**, *29*, 990–1001. [[CrossRef](#)]

18. Gupta, A.K.; Sivakumar, A.I. Job shop scheduling techniques in semiconductor manufacturing. *Int. J. Adv. Manuf. Technol.* **2006**, *27*, 1163–1169. [[CrossRef](#)]
19. Mönch, L.; Fowler, J.W.; Dauzère-Pérès, S.; Mason, S.J.; Rose, O. A survey of problems, solution techniques, and future challenges in scheduling semiconductor manufacturing operations. *J. Sched.* **2011**, *14*, 583–599. [[CrossRef](#)]
20. Pan, C.; Zhou, M.; Qiao, Y.; Wu, N. Scheduling Cluster Tools in Semiconductor Manufacturing: Recent Advances and Challenges. *IEEE Trans. Autom. Sci. Eng.* **2017**, *15*, 586–601. [[CrossRef](#)]
21. May, G.S.; Spanos, C.J. *Fundamentals of Semiconductor Manufacturing and Process Control*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2006.
22. Mönch, L.; Fowler, J.W.; Mason, S.J. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*; Springer: Berlin/Heidelberg, Germany, 2013.
23. Koo, P.-H.; Moon, D.H. A Review on Control Strategies of Batch Processing Machines in Semiconductor Manufacturing. In Proceedings of the 7th IFAC Conference on Manufacturing Modelling, Management, and Control, Saint Petersburg, Russia, 19–21 June 2013; International Federation of Automatic Control: Saint Petersburg, Russia, 2013.
24. Fowler, J.W.; Mönch, L. A survey of scheduling with parallel batch (p-batch) processing. *Eur. J. Oper. Res.* **2022**, *298*, 1–24. [[CrossRef](#)]
25. Allahverdi, A. The third comprehensive survey on scheduling problems with setup times/costs. *Eur. J. Oper. Res.* **2015**, *246*, 345–378. [[CrossRef](#)]
26. Lin, D.; Lee, C.K.M. A review of the research methodology for the re-entrant scheduling problem. *Int. J. Prod. Res.* **2011**, *49*, 2221–2242.
27. Yugma, C.; Blue, J.; Dauzère-Pérès, S.; Obeid, A. Integration of scheduling and advanced process control in semiconductor manufacturing: Review and outlook. *J. Sched.* **2015**, *18*, 195–205. [[CrossRef](#)]
28. Obeid, A.; Dauzère-Pérès, S.; Yugma, C. Scheduling on parallel machines with time constraints and Equipment Health Factors. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Seoul, Republic of Korea, 20–24 August 2012; pp. 401–406. [[CrossRef](#)]
29. Duc, L.M.; Tan, C.M.; Luo, M.; Leng, I.C.H. Maintenance Scheduling of Plasma Etching Chamber in Wafer Fabrication for High-Yield Etching Process. *IEEE Trans. Semicond. Manuf.* **2014**, *27*, 204–211. [[CrossRef](#)]
30. Jia, J.; Mason, S.J. Semiconductor manufacturing scheduling of jobs containing multiple orders on identical parallel machines. *Int. J. Prod. Res.* **2009**, *47*, 2565–2585. [[CrossRef](#)]
31. T'Kindt, V.; Billaut, J.C. *Multicriteria Scheduling: Theory, Models and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2002.
32. Ikura, Y.; Gimple, M. Efficient Scheduling Algorithms for a Single Batch Processing Machine. *Oper. Res. Lett.* **1986**, *5*, 61–65. [[CrossRef](#)]
33. Uzsoy, R. Scheduling a Single Batch Processing Machine with Non-Identical Job Sizes. *Int. J. Prod. Res.* **1994**, *32*, 1615–1635. [[CrossRef](#)]
34. Chen, H.; Du, B.; Huang, G.Q. Scheduling a batch processing machine with non-identical job sizes: A clustering perspective. *Int. J. Prod. Res.* **2011**, *49*, 5755–5778. [[CrossRef](#)]
35. Damodaran, P.; Velez-Gallego, M. Heuristics for makespan minimisation on parallel batch processing machines with unequal job ready times. *Int. J. Adv. Manuf. Technol.* **2010**, *49*, 1119–1128. [[CrossRef](#)]
36. Lee, Y.H.; Lee, Y.H. Minimising makespan heuristics for scheduling a single batch machine processing machine with non-identical job sizes. *Int. J. Prod. Res.* **2013**, *51*, 3488–3500. [[CrossRef](#)]
37. Li, X.; Zhang, K. Single batch processing machine scheduling with two-dimensional bin packing constraints. *Int. J. Prod. Econ.* **2018**, *196*, 113–121. [[CrossRef](#)]
38. Parsa, N.R.; Karimi, B.; Husseini, S.M. Minimizing total flow time on a batch processing machine using a hybrid max-min ant system. *Comput. Ind. Eng.* **2016**, *99*, 372–381. [[CrossRef](#)]
39. Jolai, F.; Dupont, L. Minimizing mean flow times criteria on a single batch processing machine with non-identical jobs sizes. *Int. J. Prod. Econ.* **1998**, *55*, 273–280. [[CrossRef](#)]
40. Shao, H.; Chen, H.P. Minimising makespan for single burn-in oven scheduling problems using ACO+DP approach. *Int. J. Manuf. Res.* **2010**, *5*, 271. [[CrossRef](#)]
41. Xu, R.; Chen, H.; Li, X. Makespan minimization on single batch-processing machine via ant colony optimization. *Comput. Oper. Res.* **2012**, *39*, 582–593. [[CrossRef](#)]
42. Zhou, S.; Chen, H.; Xu, R.; Li, X. Minimizing makespan on a single batch processing machine with dynamic job arrivals and non-identical job sizes. *Int. J. Prod. Res.* **2014**, *52*, 2258–2274. [[CrossRef](#)]
43. Mathirajan, M.; Bhargav, V.; Ramachandran, V. Minimizing total weighted tardiness on a batch-processing machine with non-agreeable release times and due dates. *Int. J. Adv. Manuf. Technol.* **2010**, *48*, 1133–1148. [[CrossRef](#)]
44. Dauzère-Pérès, S.; Mönch, L. Scheduling jobs on a single batch processing machine with incompatible job families and weighted number of tardy jobs objective. *Comput. Oper. Res.* **2013**, *40*, 1224–1233. [[CrossRef](#)]
45. Cheng, B.; Cai, J.; Yang, S.; Hu, X. Algorithms for scheduling incompatible job families on single batching machine with limited capacity. *Comput. Ind. Eng.* **2014**, *75*, 116–120. [[CrossRef](#)]
46. Lu, S.; Kong, M.; Zhou, Z.; Liu, X.; Liu, S. A hybrid metaheuristic for a semiconductor production scheduling problem with deterioration effect and resource constraints. *Oper. Res.* **2022**, *22*, 5405–5440. [[CrossRef](#)]

47. Mason, S.J.; Chen, J.-S. Scheduling multiple orders per job in a single machine to minimize total completion time. *Eur. J. Oper. Res.* **2010**, *207*, 70–77. [[CrossRef](#)]
48. Sobeyko, O.; Mönch, L. Genetic algorithms to solve a single machine multiple orders per job scheduling problem. In Proceedings of the Winter Simulation Conference, Baltimore, MD, USA, 5–8 December 2010; pp. 2493–2503. [[CrossRef](#)]
49. Sarin, S.C.; Wang, L.; Cheng, M. A single-machine, single-wafer-processing, multiple-lots-per-carrier scheduling problem to minimize the sum of lot completion times. *Comput. Oper. Res.* **2012**, *39*, 1411–1418. [[CrossRef](#)]
50. Sobeyko, O.; Mönch, L. Grouping genetic algorithms for solving single machine multiple orders per job scheduling problems. *Ann. Oper. Res.* **2015**, *235*, 709–739. [[CrossRef](#)]
51. Qu, P.; Mason, S. Metaheuristic scheduling of 300-mm lots containing multiple orders. *IEEE Trans. Semicond. Manuf.* **2005**, *18*, 633–643. [[CrossRef](#)]
52. Rocholl, J.; Mönch, L. Hybrid algorithms for the earliness-tardiness single-machine multiple orders per job scheduling problem with a common due date. *RAIRO-Oper. Res.* **2018**, *52*, 1329–1350. [[CrossRef](#)]
53. Cai, Y.; Kutanoglu, E.; Hasenbein, J.; Qin, J. Single-machine scheduling with advanced process control constraints. *J. Sched.* **2012**, *15*, 165–179. [[CrossRef](#)]
54. Pang, J.; Zhou, H.; Tsai, Y.-C.; Chou, F.-D. A scatter simulated annealing algorithm for the bi-objective scheduling problem for the wet station of semiconductor manufacturing. *Comput. Ind. Eng.* **2018**, *123*, 54–66. [[CrossRef](#)]
55. Chung, T.-P.; Xue, Z.; Wu, T.; Shih, S.C. Minimising total completion time on single-machine scheduling with new integrated maintenance activities. *Int. J. Prod. Res.* **2019**, *57*, 918–930. [[CrossRef](#)]
56. Ramacher, R.; Mönch, L. An automated negotiation approach to solve single machine scheduling problems with interfering job sets. *Comput. Ind. Eng.* **2016**, *99*, 318–329. [[CrossRef](#)]
57. Sobeyko, O.; Mönch, L. A comparison of heuristics to solve a single machine batching problem with unequal ready times of the jobs. In Proceedings of the Winter Simulation Conference, Phoenix, AZ, USA, 11–14 December 2011; pp. 2006–2016. [[CrossRef](#)]
58. Jia, W.; Jiang, Z.; Li, Y. A job-family-oriented algorithm for re-entrant batch processing machine scheduling. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Madison, WI, USA, 17–20 August 2013; pp. 1022–1027. [[CrossRef](#)]
59. Huang, J.; Down, D.G.; Lewis, M.E.; Wu, C. Dynamically scheduling and maintaining a flexible server. *Nav. Res. Logist.* **2022**, *69*, 223–240. [[CrossRef](#)]
60. Jun, S.; Lee, S. Learning dispatching rules for single machine scheduling with dynamic arrivals based on decision trees and feature construction. *Int. J. Prod. Res.* **2021**, *59*, 2838–2856. [[CrossRef](#)]
61. Lee, C.-Y.; Uzsoy, R.; Martin-Vega, L.A. Efficient Algorithms for Scheduling Semiconductor Burn-In Operations. *Oper. Res.* **1992**, *40*, 764–775. [[CrossRef](#)]
62. Chen, H.; Du, B.; Huang, G.Q. Metaheuristics to minimise makespan on parallel batch processing machines with dynamic job arrivals. *Int. J. Comput. Integr. Manuf.* **2010**, *23*, 942–956. [[CrossRef](#)]
63. Zhou, S.; Chen, H.; Li, X. Distance matrix based heuristics to minimize makespan of parallel batch processing machines with arbitrary job sizes and release times. *Appl. Soft Comput.* **2017**, *52*, 630–641. [[CrossRef](#)]
64. Arroyo, J.E.C.; Leung, J.Y.-T. Scheduling unrelated parallel batch processing machines with non-identical job sizes and unequal ready times. *Comput. Oper. Res.* **2017**, *78*, 117–128. [[CrossRef](#)]
65. Hulett, M.; Damodaran, P.; Amouie, M. Scheduling non-identical parallel batch processing machines to minimize total weighted tardiness using particle swarm optimization. *Comput. Ind. Eng.* **2017**, *113*, 425–436. [[CrossRef](#)]
66. Lin, Y.-K.; Fowler, J.W.; Pfund, M.E. Multiple-objective heuristics for scheduling unrelated parallel machines. *Eur. J. Oper. Res.* **2013**, *227*, 239–253. [[CrossRef](#)]
67. Jia, Z.-H.; Zhang, Y.-L.; Leung, J.Y.-T.; Li, K. Bi-criteria ant colony optimization algorithm for minimizing makespan and energy consumption on parallel batch machines. *Appl. Soft Comput.* **2017**, *55*, 226–237. [[CrossRef](#)]
68. Almeder, C.; Mönch, L. Metaheuristics for scheduling jobs with incompatible families on parallel batching machines. *J. Oper. Res. Soc.* **2011**, *62*, 2083–2096. [[CrossRef](#)]
69. Lausch, S.; Mönch, L. Metaheuristic approaches for scheduling jobs on parallel batch processing machines. In *Heuristics, Metaheuristics and Approximate Methods in Planning and Scheduling*; International Series in Operations Research and Management Science; Springer: Cham, Switzerland, 2016; Volume 236.
70. Mönch, L.; Roob, S. A matheuristic framework for batch machine scheduling problems with incompatible job families and regular sum objective. *Appl. Soft Comput.* **2018**, *68*, 835–846. [[CrossRef](#)]
71. Chiang, T.-C.; Cheng, H.-C.; Fu, L.-C. A memetic algorithm for minimizing total weighted tardiness on parallel batch machines with incompatible job families and dynamic job arrival. *Comput. Oper. Res.* **2010**, *37*, 2257–2269. [[CrossRef](#)]
72. Rocholl, J.; Mönch, L.; Fowler, J.W. Electricity power cost-aware scheduling of jobs on parallel batch processing machines. In Proceedings of the Winter Simulation Conference, Gothenburg, Sweden, 9–12 December 2018; pp. 3420–3431. [[CrossRef](#)]
73. Rocholl, J.; Mönch, L.; Fowler, J. Bi-criteria parallel batch machine scheduling to minimize total weighted tardiness and electricity cost. *J. Bus. Econ.* **2020**, *90*, 1345–1381. [[CrossRef](#)]
74. Mönch, L.; Balasubramanian, H.; Fowler, J.W.; Pfund, M.E. Heuristic scheduling of jobs on parallel batch machines with incompatible job families and unequal ready times. *Comput. Oper. Res.* **2005**, *32*, 2731–2750. [[CrossRef](#)]

75. Jula, P.; Leachman, R.C. Coordinated multistage scheduling of parallel batch-processing machines under multi resource constraints. *Oper. Res.* **2010**, *58*, 933–947. [\[CrossRef\]](#)
76. Kohn, R.; Rose, O.; Laroque, C. Study on multi-objective optimization for parallel batch machine scheduling using variable neighbourhood search. In Proceedings of the Winter Simulation Conference, Washington, DC, USA, 8–11 December 2013; pp. 3654–3670. [\[CrossRef\]](#)
77. Bilyk, A.; Mönch, L.; Almeder, C. Scheduling jobs with ready times and precedence constraints on parallel batch machines using metaheuristics. *Comput. Ind. Eng.* **2014**, *78*, 175–185. [\[CrossRef\]](#)
78. Klemmt, A.; Lange, J.; Weigert, G.; Lehmann, F.; Seyfert, J. A multistage mathematical programming based scheduling approach for the photolithography area in semiconductor manufacturing. In Proceedings of the Winter Simulation Conference, Baltimore, MD, USA, 5–8 December 2010; pp. 2474–2485. [\[CrossRef\]](#)
79. Yan, B.; Chen, H.Y.; Luh, P.B.; Wang, S.; Chang, J. Litho machine scheduling with convex hull analyses. *IEEE Trans. Autom. Sci. Eng.* **2013**, *10*, 928–937. [\[CrossRef\]](#)
80. Bitar, A.; Dauzère-Pérès, S.; Yugma, C.; Roussel, R. A memetic algorithm to solve an unrelated parallel machine scheduling problem with auxiliary resources in semiconductor manufacturing. *J. Sched.* **2016**, *19*, 367–376. [\[CrossRef\]](#)
81. Ham, A. Scheduling of Dual Resource Constrained Lithography Production: Using CP and MIP/CP. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 52–61. [\[CrossRef\]](#)
82. Chung, T.; Gupta, J.N.D.; Zhao, H.; Werner, F. Minimizing the makespan on two identical parallel machines with mold constraints. *Comput. Oper. Res.* **2019**, *105*, 141–155. [\[CrossRef\]](#)
83. Ham, A.; Park, M.-J.; Shin, H.-J.; Choi, S.-Y.; Fowler, J.W. Integrated Scheduling of Jobs, Reticles, Machines, AMHS and ARHS in a Semiconductor Manufacturing. In Proceedings of the Winter Simulation Conference (WSC), Orlando, FL, USA, 14–18 December 2020; pp. 1966–1973. [\[CrossRef\]](#)
84. Deng, Y.; Bard, J.F.; Chacon, G.R.; Stuber, J. Scheduling back-end operations in semiconductor manufacturing. *IEEE Trans. Semicond. Manuf.* **2010**, *23*, 210–220. [\[CrossRef\]](#)
85. Wang, I.-L.; Wang, Y.-C.; Chen, C.-W. Scheduling unrelated parallel machines in semiconductor manufacturing by problem reduction and local search heuristics. *Flex. Serv. Manuf. J.* **2013**, *25*, 343–366. [\[CrossRef\]](#)
86. Chen, C.; Fathi, M.; Khakifirooz, M.; Wu, K. Hybrid tabu search algorithm for unrelated parallel machine scheduling in semiconductor fabs with setup times, job release, and expired times. *Comput. Ind. Eng.* **2022**, *165*, 107915. [\[CrossRef\]](#)
87. Moser, M.; Musliu, N.; Schaerf, A.; Winter, F. Exact and metaheuristic approaches for unrelated parallel machine scheduling. *J. Sched.* **2022**, *25*, 507–534. [\[CrossRef\]](#)
88. Zhang, Z.; Zheng, L.; Hou, F.; Li, N. Semiconductor final test scheduling with Sarsa(k,k) algorithm. *Eur. J. Oper. Res.* **2011**, *215*, 446–458. [\[CrossRef\]](#)
89. Lin, S.-W.; Lee, Z.-J.; Ying, K.-C.; Lin, R.-H. Meta-heuristic algorithms for wafer sorting scheduling problems. *J. Oper. Res. Soc.* **2011**, *62*, 165–174. [\[CrossRef\]](#)
90. Ying, K.-C. Scheduling identical wafer sorting parallel machines with sequence-dependent setup times using an iterated greedy heuristic. *Int. J. Prod. Res.* **2012**, *50*, 2710–2719. [\[CrossRef\]](#)
91. Ying, K.-C.; Lin, S.-W. Efficient wafer sorting scheduling using a hybrid artificial immune system. *J. Oper. Res. Soc.* **2014**, *65*, 169–179. [\[CrossRef\]](#)
92. Munoz, L.; Villalobos, J.R.; Fowler, J.W. Exact and heuristic algorithms for the parallel machine total completion time scheduling problem with dual resources, ready times, and sequence-dependent setup times. *Comput. Oper. Res.* **2022**, *143*, 105787. [\[CrossRef\]](#)
93. Chen, L.; Yang, W.; Qiu, K.; Dauzère-Pérès, S. A lexicographic optimization approach for a bi-objective parallel-machine scheduling problem minimizing total quality loss and total tardiness. *Comput. Oper. Res.* **2023**, *155*, 106245. [\[CrossRef\]](#)
94. Chen, H.; Guo, P.; Jimenez, J.; Dong, Z.S.; Cheng, W. Unrelated Parallel Machine Photolithography Scheduling Problem with Dual Resource Constraints. *IEEE Trans. Semicond. Manuf.* **2023**, *36*, 100–112. [\[CrossRef\]](#)
95. Obeid, A.; Dauzère-Pérès, S.; Yugma, C. Scheduling job families on non-identical parallel machines with time constraints. *Ann. Oper. Res.* **2014**, *213*, 221–234. [\[CrossRef\]](#)
96. Nattaf, M.; Dauzère-Pérès, S.; Yugma, C.; Wu, C.-H. Parallel machine scheduling with time constraints on machine qualifications. *Comput. Oper. Res.* **2019**, *107*, 61–76. [\[CrossRef\]](#)
97. Kao, Y.-T.; Dauzère-Pérès, S.; Blue, J.; Chang, S.-C. Impact of integrating equipment health in production scheduling for semiconductor fabrication. *Comput. Ind. Eng.* **2018**, *120*, 450–459. [\[CrossRef\]](#)
98. Pang, J.; Tsai, Y.-C.; Chou, F.-D. Feature-Extraction-Based Iterated Algorithms to Solve the Unrelated Parallel Machine Problem with Periodic Maintenance Activities. *IEEE Access* **2021**, *9*, 139089–139108. [\[CrossRef\]](#)
99. Zhang, X.; Chen, L. A general variable neighborhood search algorithm for a parallel-machine scheduling problem considering machine health conditions and preventive maintenance. *Comput. Oper. Res.* **2022**, *143*, 105738. [\[CrossRef\]](#)
100. Chung, T.-P.; Liao, C.-J.; Su, L.-H. Scheduling on identical machines with batch arrivals. *Int. J. Prod. Econ.* **2010**, *123*, 179–186. [\[CrossRef\]](#)
101. Wang, H.-K.; Chien, C.-F.; Gen, M. An Algorithm of Multi-Subpopulation Parameters with Hybrid Estimation of Distribution for Semiconductor Scheduling with Constrained Waiting Time. *IEEE Trans. Semicond. Manuf.* **2015**, *28*, 353–366. [\[CrossRef\]](#)
102. Rolim, G.A.; Nagano, M.S.; Prata, B.D.A. Formulations and an adaptive large neighborhood search for just-in-time scheduling of unrelated parallel machines with a common due window. *Comput. Oper. Res.* **2023**, *153*, 106159. [\[CrossRef\]](#)

103. Rocholl, J.; Mönch, L. Decomposition heuristics for parallel-machine multiple orders per job scheduling problems with a common due date. *J. Oper. Res. Soc.* **2019**, *72*, 1737–1753. [[CrossRef](#)]
104. Klemmt, A.; Weigert, G.; Werner, S. Optimisation approaches for batch scheduling in semiconductor manufacturing. *Eur. J. Ind. Eng.* **2011**, *5*, 338–359. [[CrossRef](#)]
105. Kohn, R.; Rose, O. Study on optimization potential influencing factors in simulation studies focused on parallel batch machine scheduling using Variable Neighbourhood Search. In Proceedings of the Winter Simulation Conference, Berlin, Germany, 9–12 December 2012. [[CrossRef](#)]
106. Tajan, J.B.C.; Sivakumar, A.I.; Gershwin, S.B. Heuristic control of multiple batch processors with incompatible job families and future job arrivals. *Int. J. Prod. Res.* **2012**, *50*, 4206–4219. [[CrossRef](#)]
107. Jia, W.; Jiang, Z.; Li, Y. Combined scheduling algorithm for re-entrant batch-processing machines in semiconductor wafer manufacturing. *Int. J. Prod. Res.* **2015**, *53*, 1866–1879. [[CrossRef](#)]
108. Kim, Y.-D.; Joo, B.-J.; Choi, S.-Y. Scheduling wafer lots on diffusion machines in a semiconductor wafer fabrication facility. *IEEE Trans. Semicond. Manuf.* **2010**, *23*, 246–254. [[CrossRef](#)]
109. Jia, W.; Jiang, Z.; Li, Y. Closed loop control-based real-time dispatching heuristic on parallel batch machines with incompatible job families and dynamic arrivals. *Int. J. Prod. Res.* **2013**, *51*, 4570–4584. [[CrossRef](#)]
110. Chen, L.; Xu, H.; Li, L.; Chen, L. Learning-based adaptive dispatching method for batch processing machines. In Proceedings of the Winter Simulations Conference, Washington, DC, USA, 8–11 December 2013; pp. 3756–3765.
111. Hung, Y.-F.; Liang, C.-H.; Chen, J.C. Sensitivity search for the rescheduling of semiconductor photolithography operations. *Int. J. Adv. Manuf. Technol.* **2013**, *67*, 73–84. [[CrossRef](#)]
112. Ham, A.M.; Cho, M. A Practical Two-Phase Approach to Scheduling of Photolithography Production. *IEEE Trans. Semicond. Manuf.* **2015**, *28*, 367–373. [[CrossRef](#)]
113. Zhang, P.; Lv, Y.; Zhang, J. An improved imperialist competitive algorithm based photolithography machines scheduling. *Int. J. Prod. Res.* **2018**, *56*, 1017–1029. [[CrossRef](#)]
114. Kim, T.; Kim, H.; Lee, T.-E.; Morrison, J.R.; Kim, E. On Scheduling a Photolithography Toolset Based on a Deep Reinforcement Learning Approach with Action Filter. In Proceedings of the Winter Simulation Conference (WSC), Phoenix, AZ, USA, 12–15 December 2021; pp. 1–10. [[CrossRef](#)]
115. Chien, C.-F.; Lan, Y.-B. Agent-based approach integrating deep reinforcement learning and hybrid genetic algorithm for dynamic scheduling for Industry 3.5 smart production. *Comput. Ind. Eng.* **2021**, *162*, 107782. [[CrossRef](#)]
116. Cao, Z.; Lin, C.; Zhou, M.; Zhou, C.; Sedraoui, K. Two-Stage Genetic Algorithm for Scheduling Stochastic Unrelated Parallel Machines in a Just-in-Time Manufacturing Context. *IEEE Trans. Autom. Sci. Eng.* **2022**, *20*, 936–949. [[CrossRef](#)]
117. Lee, D.; Lee, D.; Kim, K. Self-growth learning-based machine scheduler to minimize setup time and tardiness in OLED display semiconductor manufacturing. *Appl. Soft Comput.* **2023**, *145*, 110600. [[CrossRef](#)]
118. Liu, C.-H. A genetic algorithm based approach for scheduling of jobs containing multiple orders in a three-machine flowshop. *Int. J. Prod. Res.* **2010**, *48*, 4379–4396. [[CrossRef](#)]
119. Yao, F.S.; Zhao, M.; Zhang, H. Two-stage hybrid flow shop scheduling with dynamic job arrivals. *Comput. Oper. Res.* **2012**, *39*, 1701–1712. [[CrossRef](#)]
120. Kim, H.-J.; Lee, J.-H. Three-machine flow shop scheduling with overlapping waiting time constraints. *Comput. Oper. Res.* **2019**, *101*, 93–102. [[CrossRef](#)]
121. Celano, G.; Costa, A.; Fichera, S. Constrained scheduling of the inspection activities on semiconductor wafers grouped in families with sequence-dependent set-up times. *Int. J. Adv. Manuf. Technol.* **2010**, *46*, 695–705. [[CrossRef](#)]
122. Nawaz, M.; Encore, E.E.; Ham, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega* **1983**, *11*, 91–95. [[CrossRef](#)]
123. Nowicki, E. The permutation flow shop with buffers. A tabu search approach. *Eur. J. Oper. Res.* **1999**, *116*, 205–219. [[CrossRef](#)]
124. Noroozi, A.; Mokhtari, H.; Kamal Abadi, I.N. Research on computational intelligence algorithms with adaptive learning approach for scheduling problems with batch processing machines. *Neurocomputing* **2013**, *101*, 190–203. [[CrossRef](#)]
125. Aguirre, A.M.; Méndez, C.A.; Castro, P.M. A novel optimization method to automated wet-etch station scheduling in semiconductor manufacturing systems. *Comput. Chem. Eng.* **2011**, *35*, 2960–2972. [[CrossRef](#)]
126. Aguirre, A.M.; Méndez, C.A.; Gutierrez, G.; De Prada, C. An improvement-based MILP optimization approach to complex AWS scheduling. *Comput. Chem. Eng.* **2012**, *47*, 217–226. [[CrossRef](#)]
127. Aguirre, A.M.; Méndez, C.A.; Castro, P.M. A hybrid scheduling approach for automated flowshops with material handling and time constraints. *Int. J. Prod. Res.* **2014**, *52*, 2788–2806. [[CrossRef](#)]
128. Rotondo, A.; Young, P.; Geraghty, J. Sequencing optimisation for makespan improvement at wet-etch tools. *Comput. Oper. Res.* **2015**, *53*, 261–274. [[CrossRef](#)]
129. Wang, I.-L.; Yang, T.; Chang, Y.-B. Scheduling two-stage hybrid flow shops with parallel batch, release time, and machine eligibility constraints. *J. Intell. Manuf.* **2012**, *23*, 2271–2280. [[CrossRef](#)]
130. Qin, M.; Wang, R.; Shi, Z.; Liu, L.; Shi, L. A Genetic Programming-Based Scheduling Approach for Hybrid Flow Shop with a Batch Processor and Waiting Time Constraint. *IEEE Trans. Autom. Sci. Eng.* **2021**, *18*, 94–105. [[CrossRef](#)]
131. Wu, M.-C.; Chiou, C.-W. Scheduling semiconductor in-line steppers in new product/process introduction scenarios. *Int. J. Prod. Res.* **2010**, *48*, 1835–1852. [[CrossRef](#)]

132. Fu, M.; Askin, R.; Fowler, J.; Haghnevis, M.; Keng, N.; Pettinato, J.S.; Zhang, M. Batch production scheduling for semiconductor back-end operations. *IEEE Trans. Semicond. Manuf.* **2011**, *24*, 249–260. [[CrossRef](#)]
133. Tan, Y.; Mönch, L.; Fowler, J.W. A hybrid scheduling approach for a two-stage flexible flow shop with batch processing machines. *J. Sched.* **2018**, *21*, 209–226. [[CrossRef](#)]
134. Hekmatfar, M.; Fatemi Ghomi, S.; Karimi, B. Two stage reentrant hybrid flow shop with setup times and the criterion of minimizing makespan. *Appl. Soft Comput.* **2011**, *11*, 4530–4539. [[CrossRef](#)]
135. Bard, J.F.; Gao, Z.; Chacon, R.; Stuber, J. Daily scheduling of multi-pass lots at assembly and test facilities. *Int. J. Prod. Res.* **2013**, *51*, 7047–7070. [[CrossRef](#)]
136. Gao, Z.; Bard, J.F.; Chacon, R.; Stuber, J. An assignment-sequencing methodology for scheduling assembly and test operations with multi-pass requirements. *IIE Trans.* **2015**, *47*, 153–172. [[CrossRef](#)]
137. Lin, C.-C.; Liu, W.-Y.; Chen, Y.-H. Considering stockers in reentrant hybrid flow shop scheduling with limited buffer capacity. *Comput. Ind. Eng.* **2020**, *139*, 106154. [[CrossRef](#)]
138. Ham, M.; Lee, Y.H.; An, J. IP-Based Real-Time Dispatching for Two-Machine Batching Problem with Time Window Constraints. *IEEE Trans. Autom. Sci. Eng.* **2011**, *8*, 589–597. [[CrossRef](#)]
139. Jia, W.; Chen, H.; Liu, L.; Jiang, Z.; Li, Y. Full-batch-oriented scheduling algorithm on batch processing workstation of  $\beta_1 \rightarrow \beta_2$  type with re-entrant flow. *Int. J. Comput. Integr. Manuf.* **2017**, *30*, 1029–1042. [[CrossRef](#)]
140. Jia, W.; Chen, H.; Liu, L.; Jiang, Z.; Li, Y. A slack optimization unified model of regrouping and sequencing batches for  $\beta_1 \rightarrow \beta_2$  manufacturing system. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* **2017**, *233*, 665–672. [[CrossRef](#)]
141. Branke, J.; Groves, M.J.; Hildebrandt, T. Evolving control rules for a dual-constrained job scheduling scenario. In Proceedings of the Winter Simulation Conference, Washington, DC, USA, 11–14 December 2016; pp. 2568–2579. [[CrossRef](#)]
142. Kim, Y.-D.; Kang, J.-H.; Lee, G.-E.; Lim, S.-K. Scheduling algorithms for minimizing tardiness of orders at the burn-in workstation in a semiconductor manufacturing system. *IEEE Trans. Semicond. Manuf.* **2011**, *24*, 14–26. [[CrossRef](#)]
143. Bang, J.-Y.; Kim, Y.-D. Scheduling algorithms for a semiconductor probing facility. *Comput. Oper. Res.* **2011**, *38*, 666–673. [[CrossRef](#)]
144. Jung, C.; Pabst, D.; Ham, M.; Stehli, M.; Rothe, M. An Effective Problem Decomposition Method for Scheduling of Diffusion Processes Based on Mixed Integer Linear Programming. *IEEE Trans. Semicond. Manuf.* **2014**, *27*, 357–363. [[CrossRef](#)]
145. Dugardin, F.; Yalaoui, F.; Amodeo, L. New multi-objective method to solve reentrant hybrid flow shop scheduling problem. *Eur. J. Oper. Res.* **2010**, *203*, 22–31. [[CrossRef](#)]
146. Gong, S.; Huang, R.; Cao, Z. An improved symbiotic organisms search algorithm for low-yield stepper scheduling problem. In Proceedings of the IEEE Conference on Automation Science and Engineering, Xi'an, China, 20–23 August 2017; pp. 289–294. [[CrossRef](#)]
147. Lin, J.T.; Chen, C.-M.; Chiu, C.-C.; Fang, H.-Y. Simulation optimization with PSO and OCBA for semiconductor back-end assembly. *J. Ind. Prod. Eng.* **2013**, *30*, 452–460. [[CrossRef](#)]
148. Lin, J.T.; Chen, C.-M. Simulation optimization approach for hybrid flow shop scheduling problem in semiconductor back-end manufacturing. *Simul. Model. Pract. Theory* **2015**, *51*, 100–114. [[CrossRef](#)]
149. Kim, N.; Barde, S.; Bae, K.; Shin, H. Learning per-machine linear dispatching rule for heterogeneous multi-machines control. *Int. J. Prod. Res.* **2021**, *61*, 162–182. [[CrossRef](#)]
150. Mason, S.J.; Fowler, J.W.; Carlyle, W.M. A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. *J. Sched.* **2002**, *5*, 247–262. [[CrossRef](#)]
151. Jampani, J.; Mason, S.J. A column generation heuristic for complex job shop multiple orders per job scheduling. *Comput. Ind. Eng.* **2010**, *58*, 108–118. [[CrossRef](#)]
152. Knopp, S.; Dauzère-Peres, S.; Yugma, C. Flexible job-shop scheduling with extended route flexibility for semiconductor manufacturing. In Proceedings of the Winter Simulation Conference 2014, Savannah, GA, USA, 7–10 December 2014; pp. 2478–2489. [[CrossRef](#)]
153. Knopp, S.; Dauzère-Pérès, S.; Yugma, C. A batch-oblivious approach for Complex Job-Shop scheduling problems. *Eur. J. Oper. Res.* **2017**, *263*, 50–61. [[CrossRef](#)]
154. Tamssaouet, K.; Dauzère-Peres, S.; Yugma, C.; Knopp, S.; Pinaton, J. A study on the integration of complex machines in complex job shop scheduling. In Proceedings of the Winter Simulation Conference, Gothenburg, Sweden, 9–12 December 2018; pp. 3561–3572. [[CrossRef](#)]
155. Yugma, C.; Dauzère-Pérès, S.; Artigues, C.; Derreumaux, A.; Sibille, O. A batching and scheduling algorithm for the diffusion area in semiconductor manufacturing. *Int. J. Prod. Res.* **2012**, *50*, 2118–2132. [[CrossRef](#)]
156. Wang, S.; Wang, L. A knowledge-based multi-agent evolutionary algorithm for semiconductor final testing scheduling problem. *Knowl.-Based Syst.* **2015**, *84*, 1–9. [[CrossRef](#)]
157. Wang, S.; Wang, L.; Liu, M.; Xu, Y. A hybrid estimation of distribution algorithm for the semiconductor final testing scheduling problem. *J. Intell. Manuf.* **2015**, *26*, 861–871. [[CrossRef](#)]
158. Sang, H.-Y.; Duan, P.-Y.; Li, J.-Q. An effective invasive weed optimization algorithm for scheduling semiconductor final testing problem. *Swarm Evol. Comput.* **2018**, *38*, 42–53. [[CrossRef](#)]
159. Cao, Z.; Lin, C.; Zhou, M.; Huang, R. Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling. *IEEE Trans. Autom. Sci. Eng.* **2018**, *16*, 825–837. [[CrossRef](#)]

160. Chou, C.-W.; Chien, C.-F.; Gen, M. A Multi objective Hybrid Genetic Algorithm for TFT-LCD Module Assembly Scheduling. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 692–705. [[CrossRef](#)]
161. Chung, B.-S.; Lim, J.; Park, I.-B.; Park, J.; Seo, M.; Seo, J. Setup change scheduling for semiconductor packaging facilities using a genetic algorithm with an operator recommender. *IEEE Trans. Semicond. Manuf.* **2014**, *27*, 377–387. [[CrossRef](#)]
162. Park, I.-B.; Huh, J.; Kim, J.; Park, J. A Reinforcement Learning Approach to Robust Scheduling of Semiconductor Manufacturing Facilities. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1420–1431. [[CrossRef](#)]
163. Park, I.-B.; Park, J. Scalable Scheduling of Semiconductor Packaging Facilities Using Deep Reinforcement Learning. *IEEE Trans. Cybern.* **2021**, *53*, 3518–3531. [[CrossRef](#)] [[PubMed](#)]
164. Wu, K.; Huang, E.; Wang, M.; Zheng, M. Job scheduling of diffusion furnaces in semiconductor fabrication facilities. *Eur. J. Oper. Res.* **2022**, *301*, 141–152. [[CrossRef](#)]
165. Chen, T. An optimized tailored nonlinear fluctuation smoothing rule for scheduling a semiconductor manufacturing factory. *Comput. Ind. Eng.* **2010**, *58*, 317–325. [[CrossRef](#)]
166. Chen, T. A localised fuzzy-neural fluctuation smoothing rule for job scheduling in a wafer fab. *Int. J. Manuf. Res.* **2012**, *7*, 409–425. [[CrossRef](#)]
167. Li, L.; Yu, Q. Scheduling strategy of semiconductor production lines with remaining cycle time prediction. In Proceedings of the Winter Simulation Conference, Las Vegas, NV, USA, 3–6 December 2017; pp. 3679–3690. [[CrossRef](#)]
168. Kim, H.; Lim, D.-E.; Lee, S. Deep Learning-Based Dynamic Scheduling for Semiconductor Manufacturing with High Uncertainty of Automated Material Handling System Capability. *IEEE Trans. Semicond. Manuf.* **2020**, *33*, 13–22. [[CrossRef](#)]
169. Lee, Y.H.; Kim, J.W. Daily stepper scheduling rule in the semiconductor manufacturing for MTO products. *Int. J. Adv. Manuf. Technol.* **2011**, *54*, 323–336. [[CrossRef](#)]
170. Chiang, T.-C.; Fu, L.-C. Rule-based scheduling in wafer fabrication with due date-based objectives. *Comput. Oper. Res.* **2012**, *39*, 2820–2835. [[CrossRef](#)]
171. Yao, S.; Jiang, Z.; Li, N.; Zhang, H.; Geng, N. A multi-objective dynamic scheduling approach using multiple attribute decision making in semiconductor manufacturing. *Int. J. Prod. Econ.* **2011**, *130*, 125–133. [[CrossRef](#)]
172. Yao, S.; Jiang, Z.; Li, N.; Geng, N.; Liu, X. A decentralised multi-objective scheduling methodology for semiconductor manufacturing. *Int. J. Prod. Res.* **2011**, *49*, 7227–7252. [[CrossRef](#)]
173. Bard, J.F.; Jia, S.; Chacon, R.; Stuber, J. Integrating optimisation and simulation approaches for daily scheduling of assembly and test operations. *Int. J. Prod. Res.* **2014**, *53*, 2617–2632. [[CrossRef](#)]
174. Jia, S.; Bard, J.F.; Chacon, R.; Stuber, J. Improving performance of dispatch rules for daily scheduling of assembly and test operations. *Comput. Ind. Eng.* **2015**, *90*, 86–106. [[CrossRef](#)]
175. Chung, Y.H.; Cho, K.H.; Park, S.C.; Kim, B.H. Dedication load based dispatching rule for photolithograph machines with dedication constraint. In Proceedings of the Winter Simulation Conference, Washington, DC, USA, 11–14 December 2016; pp. 2731–2739. [[CrossRef](#)]
176. Cui, M.; Li, L. A closed loop dynamic scheduling method based on load balancing for semiconductor wafer fabrication facility. In Proceedings of the IEEE Conference on Smart Manufacturing, Industrial and Logistics Engineering, Hsinchu, Taiwan, 8–9 February 2018; pp. 1–6. [[CrossRef](#)]
177. Li, L.; Sun, Z.; Zhou, M.; Qiao, F. Adaptive dispatching rule for semiconductor wafer fabrication facility. *IEEE Trans. Autom. Sci. Eng.* **2013**, *10*, 354–364. [[CrossRef](#)]
178. Li, L.; Min, Z. An efficient adaptive dispatching method for semiconductor wafer fabrication facility. *Int. J. Adv. Manuf. Technol.* **2016**, *84*, 315–325. [[CrossRef](#)]
179. Lee, J.-H.; Kim, Y.; Kim, Y.B.; Kim, B.-H.; Jung, G.-H.; Kim, H.-J. A Sequential Search Method of Dispatching Rules for Scheduling of LCD Manufacturing Systems. *IEEE Trans. Semicond. Manuf.* **2020**, *33*, 496–503. [[CrossRef](#)]
180. Yu, Q.; Yang, H.; Lin, K.-Y.; Li, L. A self-organized approach for scheduling semiconductor manufacturing systems. *J. Intell. Manuf.* **2021**, *32*, 689–706. [[CrossRef](#)]
181. Shiue, Y.-R.; Guh, R.-S.; Lee, K.-C. Study of SOM-based intelligent multi-controller for real-time scheduling. *Appl. Soft Comput.* **2011**, *11*, 4569–4580. [[CrossRef](#)]
182. Ma, Y.; Qiao, F.; Lu, J. Learning-based dynamic scheduling of semiconductor manufacturing system. In Proceedings of the 2016 IEEE International Conference on Automation Science and Engineering, Fort Worth, TX, USA, 21–25 August 2016; pp. 1394–1399.
183. Lim, J.; Chae, M.-J.; Yang, Y.; Park, I.-B.; Lee, J.; Park, J. Fast Scheduling of Semiconductor Manufacturing Facilities Using Case-Based Reasoning. *IEEE Trans. Semicond. Manuf.* **2016**, *29*, 22–32. [[CrossRef](#)]
184. Chan, C.W.; Ping Gan, B.; Cai, W. Towards Situation Aware Dispatching in a Dynamic and Complex Manufacturing Environment. In Proceedings of the Winter Simulation Conference (WSC), Orlando, FL, USA, 14–18 December 2020; pp. 528–539. [[CrossRef](#)]
185. Shiue, Y.-R.; Lee, K.-C.; Su, C.-T. A Reinforcement Learning Approach to Dynamic Scheduling in a Product-Mix Flexibility Environment. *IEEE Access* **2020**, *8*, 106542–106553. [[CrossRef](#)]
186. Fu, M.C. Optimization for simulation: Theory vs. Practice. *INFORMS J. Comput.* **2002**, *14*, 192–215. [[CrossRef](#)]
187. Chang, X.; Dong, M.; Yang, D. Multi-objective real-time dispatching for integrated delivery in a Fab using GA based simulation optimization. *J. Manuf. Syst.* **2013**, *32*, 741–751. [[CrossRef](#)]

188. Kuck, M.; Broda, E.; Freitag, M.; Hildebrandt, T.; Frazzon, E.M. Towards adaptive simulation-based optimization to select individual dispatching rules for production control. In Proceedings of the Winter Simulation Conference, Las Vegas, NV, USA, 3–6 December 2017; pp. 3852–3863. [CrossRef]
189. Ghasemi, A.; Ashoori, A.; Heavey, C. Evolutionary Learning Based Simulation Optimization for Stochastic Job Shop Scheduling Problems. *Appl. Soft Comput.* **2021**, *106*, 107309. [CrossRef]
190. Lin, C.; Cao, Z.; Zhou, M. Learning-Based Grey Wolf Optimizer for Stochastic Flexible Job Shop Scheduling. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 3659–3671. [CrossRef]
191. Liu, J.; Qiao, F.; Zou, M.; Zinn, J.; Ma, Y.; Vogel-Heuser, B. Dynamic scheduling for semiconductor manufacturing systems with uncertainties using convolutional neural networks and reinforcement learning. *Complex Intell. Syst.* **2022**, *8*, 4641–4662. [CrossRef]
192. Park, I.-B.; Huh, J.; Park, J. A Generation and Repair Approach to Scheduling Semiconductor Packaging Facilities Using Case-Based Reasoning. *IEEE Access* **2023**, *11*, 50631–50641. [CrossRef]
193. Sakr, A.H.; Aboelhassan, A.; Yacout, S.; Bassetto, S. Simulation and deep reinforcement learning for adaptive dispatching in semiconductor manufacturing systems. *J. Intell. Manuf.* **2023**, *34*, 1311–1324. [CrossRef]
194. Yoon, H.J.; Kim, J.G. Heuristic scheduling policies for a semiconductor wafer fabrication facility: Minimizing variation of cycle times. *Int. J. Adv. Manuf. Technol.* **2013**, *67*, 171–180. [CrossRef]
195. Li, Y.; Jiang, Z.; Jia, W. An integrated release and dispatch policy for semiconductor wafer fabrication. *Int. J. Prod. Res.* **2014**, *52*, 2275–2292. [CrossRef]
196. Singh, R.; Mathirajan, M. A New Stage-Wise Control Release Policy for Semiconductor Wafer Fabrication Systems. *IEEE Trans. Semicond. Manuf.* **2021**, *34*, 115–134. [CrossRef]
197. Qiao, F.; Liu, J.; Ma, Y. Industrial big-data-driven and CPS-based adaptive production scheduling for smart manufacturing. *Int. J. Prod. Res.* **2021**, *59*, 7139–7159. [CrossRef]
198. Driessel, R.; Mönch, L. An integrated scheduling and material-handling approach for complex job shops: A computational study. *Int. J. Prod. Res.* **2012**, *50*, 5966–5985. [CrossRef]
199. Guo, C.; Zhibin, J.; Zhang, H.; Li, N. Decomposition-based classified ant colony optimization algorithm for scheduling semiconductor wafer fabrication system. *Comput. Ind. Eng.* **2012**, *62*, 141–151. [CrossRef]
200. Kopanos, G.M.; Xenos, D.; Andreev, S.; O'Donnell, T.; Feely, S. Advanced Production Scheduling in a Seagate Technology Wafer Fab. In Proceedings of the Winter Simulation Conference (WSC), Orlando, FL, USA, 14–18 December 2020; pp. 1954–1965. [CrossRef]
201. Barhebwa-Mushamuka, F.; Dauzère-Pérès, S.; Yugma, C. A global scheduling approach for cycle time control in complex manufacturing systems. *Int. J. Prod. Res.* **2021**, *61*, 559–579. [CrossRef]
202. Qiao, F.; Ma, Y.; Zhou, M.; Wu, Q. A Novel Rescheduling Method for Dynamic Semiconductor Manufacturing Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *50*, 1679–1689. [CrossRef]
203. Zhong, H.; Liu, M.; Hao, J.; Jiang, S. An Operation-Group Based Soft Scheduling Approach for Uncertain Semiconductor Wafer Fabrication System. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *48*, 1332–1347. [CrossRef]
204. Zhong, H.; Liu, M.; Bao, L. A job-priority based soft scheduling approach for uncertain work area scheduling in Semiconductor Manufacturing. *Int. J. Prod. Res.* **2021**, *60*, 5012–5028. [CrossRef]
205. Jamrus, T.; Chien, C.F.; Gen, M.; Sethanan, K. Hybrid Particle Swarm Optimization Combined with Genetic Operators for Flexible Job-Shop Scheduling under Uncertain Processing Time for Semiconductor Manufacturing. *IEEE Trans. Semicond. Manuf.* **2018**, *31*, 32–41. [CrossRef]
206. Fowler, J.W.; Mönch, L.; Ponsignon, T. Discrete-Event simulation for semiconductor wafer fabrication facilities: A tutorial. *Int. J. Ind. Eng.* **2015**, *22*, 661–682.
207. Ouelhadj, D.; Petrovic, S. A survey of dynamic scheduling in manufacturing systems. *J. Sched.* **2009**, *12*, 417–431. [CrossRef]
208. Lee, Y.H.; Ham, M.; Yoo, B.; Lee, J.S. Daily planning and scheduling system for the EDS process in a semiconductor manufacturing facility. *Int. J. Adv. Manuf. Technol.* **2009**, *41*, 568–579. [CrossRef]
209. Xiao, J.; Yang, H.; Zhang, C.; Zheng, L.; Gupta, J.N.D. A hybrid Lagrangian-simulated annealing-based heuristic for the parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times. *Comput. Oper. Res.* **2015**, *63*, 72–82. [CrossRef]
210. Driessel, R.; Mönch, L. Variable neighborhood search approaches for scheduling jobs on parallel machines with sequence-dependent setup times, precedence constraints, and ready times. *Comput. Ind. Eng.* **2011**, *61*, 336–345. [CrossRef]
211. Fallahi, A.; Shahidi-Zadeh, B.; Niaki, S.T.A. Unrelated parallel batch processing machine scheduling for production systems under carbon reduction policies: NSGA-II and MOGWO metaheuristics. *Soft Comput.* **2023**. [CrossRef]
212. Uetz, M. When greediness fails: Examples from stochastic scheduling. *Oper. Res. Lett.* **2003**, *31*, 413–419. [CrossRef]
213. Asmundsson, J.; Rardin, R.L.; Uzsoy, R. Tractable Nonlinear Production Planning Models for Semiconductor Wafer Fabrication Facilities. *IEEE Trans. Semicond. Manuf.* **2006**, *19*, 95–111. [CrossRef]
214. Fowler, J.W.; Robinson, J. *Measurement and Improvement of Manufacturing Capacity (MIMAC) Final Report*; Technical Report No. 95062861A-TR; SEMATECH, Inc.: Austin, TX, USA, 1995.
215. Huang, E.; Wu, K. Job Scheduling at Cascading Machines. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 1634–1642. [CrossRef]
216. Hung, Y.-F.; Wang, C.-C.; Wu, G.-H. Scheduling semiconductor multihead testers using metaheuristic techniques embedded with lot-specific and configuration-specific information. *Math. Probl. Eng.* **2013**, *2013*, 436701. [CrossRef]

217. Wu, J.-Z.; Hao, X.-C.; Chien, C.-F.; Gen, M. A novel bi-vector encoding genetic algorithm for the simultaneous multiple resources scheduling problem. *J. Intell. Manuf.* **2012**, *23*, 2255–2270. [[CrossRef](#)]
218. Cao, Z.; Gong, S.; Zhou, M.; Liu, K. A Self-braking Symbiotic Organisms Search Algorithm for Bi-objective Reentrant Hybrid Flow Shop Scheduling Problem. In Proceedings of the IEEE International Conference on Automation Science and Engineering, Munich, Germany, 20–24 August 2018; pp. 803–808. [[CrossRef](#)]
219. Madathil, S.C.; Nambiar, S.; Mason, S.J.; Kurz, M.E. On scheduling a photolithography area containing cluster tools. *Comput. Ind. Eng.* **2018**, *121*, 177–188. [[CrossRef](#)]
220. Yu, T.-S.; Kim, H.-J.; Lee, T.-E. Minimization of waiting time variation in a generalized two-machine flowshop with waiting time constraints and skipping jobs. *IEEE Trans. Semicond. Manuf.* **2017**, *30*, 155–165. [[CrossRef](#)]
221. Bang, J.-Y.; Kim, Y.-D.; Choi, S.-W. Multiproduct Lot Merging–Splitting Algorithms for Semiconductor Wafer Fabrication. *IEEE Trans. Semicond. Manuf.* **2012**, *25*, 200–210. [[CrossRef](#)]
222. Li, Y.; Jiang, Z.; Jia, W. API-based two-dimensional dispatching decision-making approach for semiconductor wafer fabrication with operation due date-related objectives. *Int. J. Prod. Res.* **2016**, *55*, 79–95. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.