Computers & Operations Research 163 (2024) 106508

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

A simulation optimization framework to solve Stochastic Flexible Job-Shop Scheduling Problems—Case: Semiconductor manufacturing

Ensieh Ghaedy-Heidary [a,b], Erfan Nejati [b], Amir Ghasemi [c,*], S. Ali Torabi [b]

[a] Department of Management Sciences, University of Waterloo, Waterloo, Canada
[b] School of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[c] Department of IT & Logistics, Amsterdam School of International Business (AMSIB), Amsterdam, Netherlands

ARTICLE INFO

Keywords:
Flexible manufacturing systems
Stochastic flexible job-shop scheduling problem
Semiconductor manufacturing
Industry 4.0
Simulation optimization

ABSTRACT

This paper addresses a Stochastic Flexible Job-Shop Scheduling Problem (SFJSSP) in the context of semiconductor manufacturing. Semiconductor industry is among the most capital-intensive businesses whose operational excellence is of vital importance. Within the front-end fab of the semiconductor industry, the photolithography workstation is the well-known bottleneck process. To elevate the performance of the whole semiconductor manufacturing system, developing a competent schedule for its bottleneck is essential. However, the re-entrant product flows, high uncertainties in operations times, and rapidly changing products and technologies within the photolithography, make it difficult to develop a schedule for the whole semiconductor fab. Considering Industry 4.0, hybrid methods such as Simulation Optimization (SO) have proven their applicability in addressing complex production scheduling problems. Thus, this paper develops a mathematical model for SFJSSP of the semiconductor manufacturing considering special constraints of the photolithography workstation (machine process capability, machine dedication, and maximum reticles (masks) sharing constraints). Next, we transform the developed model into an SO model integrated with a computer simulation model capable of modeling the photolithography workstation. The simulation model develops an initial schedule based on the Least Work Remaining (LWR) dispatching rule. Moreover, the simulation model calculates the objective function of the SFJSSP. A tailored Genetic Algorithm (GA) is then developed, which attempts to optimize the initially proposed schedule. To validate the superiority of the presented SO methodology in addressing SJSSPs, it is compared with previously proposed methods. Furthermore, to assess the impact of the three special constraints of the photolithography work area on system performance, two sets of experiments are proposed. In the first set of experiments, the performance of two SFJSS environments, one with the special constraints and one without, is compared. The second set of experiments involves observing the system's performance while systematically varying the severity of the special constraints. The results indicate that improved performance levels can be accomplished by enhancing flexibility within both the operations of individual jobs and the machines within the manufacturing system.

1. Introduction

Industry 4.0 (I4.0) has become a synonym for increasing productivity in the 21st century by applying digital technologies in manufacturing. Modern Job-Shop production systems, such as semiconductor fabrication plants, are a perfect example of I4.0 adaptions (Herding and Mönch, 2022). In other words, semiconductor manufacturing using multi-mode sensors, intelligent tools, and robotics can be seen as a potential early adaptor of I4.0. Semiconductors, which became commercially available 60 years ago, are one of the main products for manufacturing integrated circuits (ICs). The process by which ICs are produced is one of the most complicated technological achievements of the twentieth-century (Nishi and Doering, 2000). On the one hand, due to several manufacturing requirements, mainly expensive machines that may cost up to 100 million US Dollars, the semiconductor production is highly cost-intensive. On the other hand, the soaring increase in the demand for ICs has resulted in a considerable rise in demand amounts for semiconductors. Thus, operational excellence of different stages within the semiconductor production process has been gaining importance, as it promises cost reductions and thereby a competitive advantage.

* Corresponding author.
E-mail addresses: eghaedyh@uwaterloo.ca (E. Ghaedy-Heidary), e.nejati@ut.ac.ir (E. Nejati), a.ghasemi2@hva.nl (A. Ghasemi), satorabi@ut.ac.ir (S.A. Torabi).

https://doi.org/10.1016/j.cor.2023.106508
Received 15 September 2022; Received in revised form 22 October 2023; Accepted 4 December 2023
Available online 5 December 2023

In this regard, there are four main stages for the semiconductor production process, namely, wafer fabrication, probe, assembly, and final test (Bang and Kim, 2011; Lee et al., 2009). Wafer fabrication and probe are commonly referred to as front-end operations, and assembly and test as back-end operations (Mönch et al., 2018a). The wafer fabrication process is the most capital-intensive and complex stage among all (Lee and Lee, 2022). To be more specific, since a similar set of unit processes is necessitated for the processing of each layer, wafers may visit a particular workstation several times, once for each layer of circuitry, resulting in re-entrant product flows. Thus, scheduling wafer fabrication plants can be seen as a complex Job-Shop Scheduling Problem (JSSP) (Waschneck et al., 2018). JSSP can be defined as a problem that a given set of jobs need to be scheduled on a set of machines in a way to minimize an objective function (usually the makespan) (Geyik and Cedimoglu, 2004). When assigning one job to one machine, some constraints must be met. Firstly, each job assigned to a machine is associated with a given order and a processing time. Secondly, each machine can perform only one job at any moment (Chen et al., 2012). Lastly, the processing time of a job is fixed, and once the job is started, it cannot be interrupted (Peng et al., 2015).

Additionally, more adoption of smart production practices in wafer fabs, such as the advanced process and equipment controls, has increased the uncertainty level in the processing times of wafer fabrication operations, adding to its level of complexity (Jamrus et al., 2018). Within the wafer fabrication system, the photolithography process is known as the bottleneck process. This is mainly because of the layered nature of wafer fabrication, particularly in the case of Application Specific Integrated Circuit (ASIC) fabrication environments with high mix product portfolios and low volumes (Ghasemi et al., 2020). The photolithography process includes three main steps, coat, expose, and develop. In the first step, the wafer is covered with a thin film of a photosensitive polymer called photoresist strip. Subsequently, in the "expose" step, the wafer is exposed to ultraviolet light (UV) to print the circuit pattern onto the wafer. To do so, a reticle is used. A diverse range of recipes exists in an ASIC fab due to the range of products produced. Finally, polymerized photoresist sections are removed from the exposed wafer. Since the circuits are composed of layers, with every wafer passing through the photolithography area up to 40 times, photolithography can be considered the bottleneck resource. Since the performance of a system is determined by the bottleneck resource, developing a competent schedule for the photolithography work area results in the improvement in the performance of the whole fab (Ghasemi et al., 2020). However, there are three specific constraints that differentiate the scheduling problem of the photolithography work area from other scheduling problems (Chung et al., 2008).

Firstly, certain machines within the photolithography tool will be competent for different recipes (machine process capability constraints). Secondly, to ensure the quality of the IC, for critical layers, certain machines within the toolset will be required to be used (machine dedication constraints). Thirdly, the number of times a reticle (mask) is shared between different layer productions should be limited (maximum reticles sharing constraints) (Ghasemi et al., 2020). In this context, some papers have considered the scheduling problem of the photolithography work area as a Flexible Job-Shop Scheduling Problem (FJSSP) (Mönch et al., 2011). FJSSP is an extension of JSSP (Liu et al., 2021), where operations can be conducted on a set of compatible machines (Zhang et al., 2012). FJSSP is therefore at least as complex as JSSP (Mokhtari and Dadgar, 2015). In fact, the flexibility makes the problem much more complex as FJSSPs belong to the class of NP-Hard problems (Dosdoğru et al., 2015).

To be more detailed, as represented in Fig. 1, each layer of a wafer entering the photolithography workstation is considered a job with a predefined order of operations. Based on the machine process capability constraints, within each layer, some operations can be conducted on a set of machines (grouped by - - - lines in Fig. 1). Other constraints of this workstation, such as the machine dedication and maximum reticle sharing constraints, are satisfied within a scheduling problem to minimize the objective function. Also, JSSPs can be classified as deterministic or stochastic. Where all parameters are known exactly, they are assumed to be deterministic (Xiong et al., 2022). On the other hand, Stochastic Job-Shop Scheduling Problem (SJSSP) deals with stochastic problem parameters and/or variables (typically operations processing times (Gu et al., 2010)). Thus, considering the uncertainties in the processing times of the photolithography workstation (ranging from 15 to 60 min), the problem of scheduling operations on photolithography machines can be considered a Stochastic Flexible Job-Shop Scheduling Problem (SFJSSP). Furthermore, the mentioned SFJSSP is amongst the most complex ones since simpler versions of this problem are NP-hard (e.g., Low and Fang (2005)). In this regard and in the era of I4.0, hybrid Simulation Optimization (SO) methods have been one of the most promising approaches to address complex manufacturing system problems (Malekpour et al., 2021). On the one hand, simulation models have been used to assess industrial systems with stochastic parameters and/or variables. On the other hand, optimization techniques are critical tools to improve decisions within almost all systems. In other words, SO methods benefit from the great detail provided by simulation and the ability of optimization techniques to find good or optimal solutions, simultaneously (Figueira and Almada-Lobo, 2014). Thus, combining simulation models with optimization methods could develop a promising tool for solving various complex and stochastic industrial problems, such as scheduling within semiconductor manufacturing systems (Ghasemi et al., 2018).

There are four significant gaps in the literature considering all those mentioned above. Firstly, although, currently, there is a massive demand for semiconductors enhancing their revenue to around $1000 per wafer (Ghasemi et al., 2020), a limited amount of research has addressed the scheduling problem within its bottleneck workstation (photolithography), which is crucial in terms of enhancing the operational excellence in such complex products (in terms of production systems). Secondly, among the papers that focused on scheduling operations on photolithography machines, none considered all characteristics of this workstation. In essence, despite the fact that machine process capability, machine dedication, maximum reticles sharing constraints, and uncertainties are interrelated and critical aspects of photolithography fabs from a manufacturing systems standpoint, previous research has yet to concurrently consider all of these factors. Thirdly, the majority of the existing research within the literature has used traditional optimization methodologies to address the SFJSSPs of the photolithography work area. These methods are not capable enough to solve such complex problems within a reasonable time. Lastly, there exists a gap regarding comparative and validation strategies to evaluate the performance of the proposed methodologies as well as insightful sensitivity analysis of the photolithography manufacturing environment.

To fill these gaps, we propose an SO-based approach to address the SFJSSP of the photolithography workstation. Firstly, we develop a mathematical model for SFJSSP considering unique photolithography workstation constraints (machine process capability constraints, machine dedication constraints, maximum reticles sharing constraints) as well as its uncertain nature. Subsequently, we transform the designed mathematical model into an SO model. Next, a detailed simulation model of the photolithography workstation, considering its critical constraints and stochastic uncertainty of the processing and sequence-dependent setup times, is developed. The model is in charge of developing an initial schedule based on the Least Work Remaining (LWR) dispatching rule. Furthermore, the simulation model calculates the objective function's value (makespan in this paper). As mentioned, the photolithography area is the well-known bottleneck step of semiconductor front-end fab. On the other hand, photolithography machines are very expensive and require a huge amount of capital investment. Thus, maximizing the machine utilization in this production area is

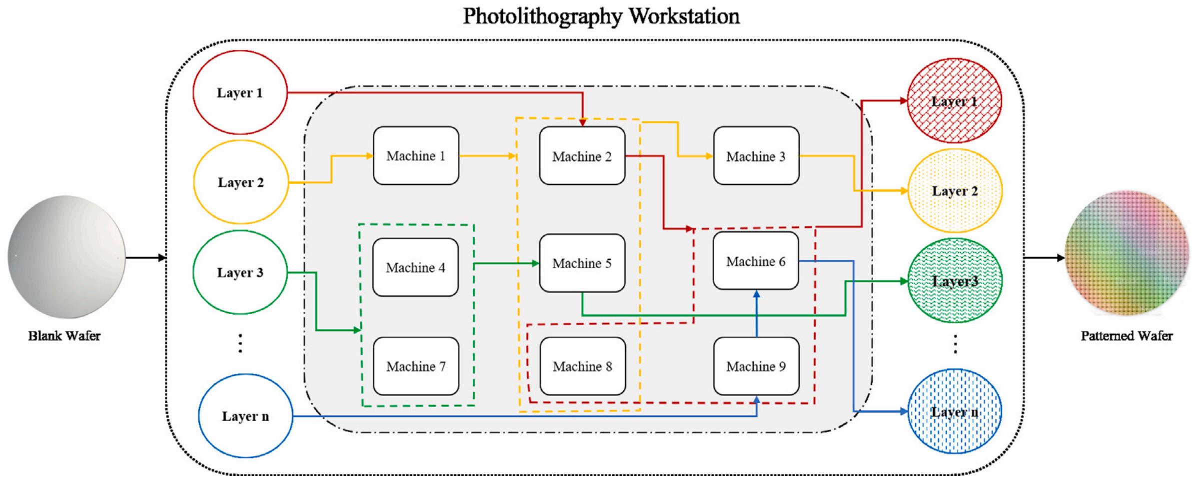

Fig. 1. Photolithography workstation as a FJSSP.

one of the most common targets of semiconductor production planners (Ghasemi et al., 2020). Accordingly, makespan (maximum completion time) minimization objective, which is one of the best-known objectives to enhance utilization levels in wafer fabs (Pfund et al., 2006), is considered within this research.

In the literature of stochastic SO, metaheuristics have been preponderantly used due to their flexibility to tackle any type of solution space and their ability to achieve good quality solutions within a reasonable amount of time (Ólafsson, 2006). Moreover, as stated by Geyik and Dosdoğru (2013), metaheuristics have also shown promising performance in achieving efficient solutions to FJSSPs. Among the proposed metaheuristics to address scheduling problems within the literature, Genetic Algorithm (GA) is respected as the most widely used metaheuristic algorithm to optimize FJSSPs due to its superior performance and strong universality (Shao et al., 2018). Thus, in this paper, we present a tailored GA to optimize the initially developed schedule iteratively integrated with the simulation model forming an SO structure. The GA proposed in this study incorporates crossover and mutation operators that ensure the preservation of precedence constraints among operations within the scheduling problem. Additionally, a significant contribution of the proposed GA lies in the development of a sorting approach, which facilitates the selection of the best solutions from each iteration. These selected solutions are then passed on to the subsequent iterations of the GA. To aid in this sorting process, a specialized distancing function, tailored for FJSSPs, is employed.

1.1. Motivation, research gaps, research question

In the following, the motivation behind undertaking the research presented in this paper is discussed. Subsequently, the research gaps identified by reviewing the literature and the research question addressed by this paper are presented.

1.1.1. Motivation

The main motivation for undertaking the research in this paper is to investigate the possibility of addressing the SFJSSP of the photolithography work area while considering all its main characteristics by developing computationally effective algorithms. The majority of the methods used to address the scheduling problem of the photolithography work area are traditional optimization methodologies that are not capable to address such complex problems within a reasonable time. In this regard, we were particularly looking for strong algorithms which are developed based on explicit mathematical formulations of the problem and account for the stochastic nature and three special constraints of the photolithography workstation, simultaneously.

1.1.2. Research gaps

We reviewed the literature on the scheduling problem of the photolithography work area which is presented in Section 2. Based on the result, a summary of the gaps within the literature is presented below (the following gaps are further detailed within Section 2):

- There is a limited number of research conducted on modeling the scheduling problem within the photolithography area of semiconductor manufacturing (the well-known bottleneck manufacturing step), taking into account the inherent characteristics of this work environment, including the three special photolithography constraints as well as the stochastic nature of the processing and the sequence-dependent setup times.
- There is a limited number of efficient solution approaches that adequately address the SFJSSP within the photolithography work area (while considering its complexities), in a reasonable time.
- Furthermore, the literature lacks comprehensive sensitivity analysis strategies that can shed light on the impact of various factors on the performance of the photolithography work area.

1.1.3. Research question

To address the research gaps mentioned in the previous subsection, the following research question is answered by this paper:

- How to devise an algorithmically efficient approach for tackling SFJSSP in the domain of photolithography, taking into account the inherent uncertainty associated with processing times and sequence-dependent setup times, while also accommodating the three constraints specific to this work area?

1.2. Contribution

In summary, the research question is addressed by the following contributions:

- A mathematical model is developed for the SFJSSP of the photolithography work area considering its uncertain nature and three special constraints (machine process capability, machine dedication, and maximum reticles (masks) sharing constraint), and further transformed into an SO model.
- A detailed simulation model of the photolithography workstation is developed that considers the stochasticity of the processing and sequence-dependent setup times and respects the special constraints of the workstation, simultaneously.
- A tailored GA is devised that incorporates an effective and noble approach as its sorting method and is capable of optimizing the schedule of complex and stochastic systems such as the semiconductors manufacturing.

- An SO model is finally developed by integrating the developed simulation model with the proposed GA.
- An insightful sensitivity analysis is provided to assess the impact of three factors, namely, *Flexibility Ratio*, *Machine Dedication Ratio*, and *Sequence-Dependent Setup Time Occurrence Ratio*, on minimizing the makespan in scheduling the photolithography work area. Using actual fab data, a test problem environment of the photolithography work area is proposed. This environment is exposed to stochastic uncertainty since the processing times and sequence-dependent setup times follow a Normal distribution. Furthermore, two different levels are considered for each factor mentioned above. Eight different SFJSSPs are constructed, with regard to these different levels, using the proposed problem environment. These scheduling problems are solved using the proposed SO model afterward. Consequently, based on the reported makespan for the problems, the effect of each factor on the system's performance is analyzed.

The remainder of this paper is then organized as follows. Section 2 provides a literature review of the articles addressing the scheduling problem in the photolithography work area. In Section 3 the developed mathematical model for the SFJSSP of the photolithography work area is presented. Also, in this section, an explanation is provided of how the mathematical model is transformed into an SO model. Section 4 elaborates on the SO model. The next section calibrates the SO model. Section 6 reports the results of the experiments carried out. Finally, Section 7 concludes the paper and looks at possible future studies.

2. Literature review

The constant increase in the use of ICs in industrial, commercial, and military products renders the semiconductor industry of critical importance to the global economy (Mönch et al., 2018b). With this rapidly increasing global demand and one of the most capital-intensive production systems (Cemernek et al., 2017), the semiconductor industry is well-positioned to invest in cost-saving process excellence. In this regard, the research on performance improvement in the semiconductor manufacturing has only come into existence in the last few decades (Gupta and Sivakumar, 2006). Furthermore, the existing literature in this context is mainly focused on two areas, the testing workstation (e.g., Cao et al. (2018), Chen et al. (1995), Ellis et al. (2004), Uzsoy et al. (1992), Wu and Chien (2008) and Xiong and Zhou (1998)) and the photolithography workstation. The photolithography workstation is a well-known bottleneck resource of most fabrication lines due to its expensive machines and complex process constraints (Lee and Lee, 2003). Therefore, effective scheduling for the photolithography work center is essential as it could significantly affect the whole semiconductor fab throughput, cycle time, and on-time delivery (Cakici and Mason, 2007). Thus, due to the importance of the photolithography work area, in this section, we focus on the scheduling problem of this work area. Surprisingly, published research on photolithography scheduling is limited within the literature. The papers published in this context within the last decades are reviewed in the following subsections. The first subsection focuses on the papers that addressed the scheduling problem of the photolithography work area as a form of JSSP, ignoring its uncertain nature. The papers reviewed in the second subsection, however, aimed at scheduling the operations of the photolithography workstation while considering the uncertainties involved, therefore addressed it as a form of SJSSP. In reviewing the papers within both of the subsections, special attention is given to the three special constraints of the photolithography as well as the solution method used to address the scheduling problem of this work area.

2.1. JSSP of the photolithography work area

This subsection presents a review of the papers that addressed the scheduling problem of the photolithography work area while ignoring the uncertain nature of this workstation. In this regard, one of the earliest works is proposed by Wein (1988). He used a simulation model to assess the effect of various input controls and sequencing rules on the performance of the semiconductor wafer fabrication. The developed simulation model represents a fictitious fab with parameters driven from data gathered at an actual fab. However, non of the three special constraints of the photolithography work area was taken into account by the author. Akcali and Uzsoy (2000) broke down the shift scheduling problem into capacity allocation and lot-sequencing subproblems. Capacity Allocation Routine (CAR) was developed for the capacity allocation problem. CAR uses an integration of a metaheuristic algorithm developed by Toktay and Uzsoy (1998) and a simulation model presented by Akcali and Uzsoy (2000). Further on, numerical experiments were proposed to study the effects of factors such as stepper capability, reticle, and setup constraints. Regarding the stepper capability, two cases were considered: fully flexible matrix and nested matrix. The former accounts for when the stepper is capable of processing all the operations while the latter happens when there are two groups of steppers: one group that can process any operation; another group capable of processing only a subset of the operations. Yugma et al. (2007) addressed the problem of scheduling the lots in the photolithography area while considering the reticle sharing constraints. To be more detailed, a simulator was developed that uses the data regarding each lot to prioritize them and determine the best equipment for processing the lots. Johnzén et al. (2008) used a scheduling simulator proposed by Yugma et al. (2007) to assess the impact of qualification management on semiconductor scheduling. They used different qualification sets and input data from a real fab to conduct their desired experiments. Klemmt et al. (2010) considered three special constraints of the photolithography work area. To address this work area's scheduling problem, a multistage optimization approach was proposed. To be more detailed, the problem was broken down into four optimization problems by creating data interfaces between them. However, they stated that the global optimum might be lost by using their proposed optimization method. Yan et al. (2012) proposed a mixed-integer optimization model, which was solved using the branch-and-cut-method. Reticle expiration and machine dedication constraints were taken into account. The proposed objective function minimized the load differences between every machine and the average. To reduce the computational intensity, the authors also proposed a two-phase model for the same problem. In the first phase, the range of the problem is decreased by relaxing some constraints while the second phase is in charge of developing a near-optimal schedule for the production system. In another study, Ham and Cho (2015) modified a scheduling model proposed by Ham (2012) for photolithography implementation. The modified model referred to as i-RTD (MIP-based real-time dispatching), is constructed so that only necessary jobs are scheduled in each decision time epoch. Moreover, all three special constraints of the photolithography work area were considered by the authors. Finally, Zhang et al. (2018) used a rolling horizon strategy to address the scheduling problem of the photolithography machines by dividing the problem into several local scheduling problems. They used an Imperialist Competitive Algorithm (ICA) to solve each scheduling task.

2.2. SJSSP of the photolithography work area

The uncertainties involved in the photolithography environment are one of the key features of this work area and have been addressed by a limited number of studies. One of these studies is conducted by Akcalt et al. (2001) in which several test and machine dedication policies were presented to assign the critical layers to steppers. In this regard, a simulation model of the photolithography work area was developed to

Table 1
Summary of publications on the photolithography work area with PT = Processing Time; ST = Setup Time; WIP = Work in Process; NOJ = Number of Operations per Job; MF = Machine Failure; MC = Machine Process Capability; MPC = Machine Dedication; RS = Reticle Sharing; SDST = Sequence-Dependent Setup Time; DR = Dispatching Rule; HE = Heuristic; MH = MetaHeuristic; S = Simulation; SO = Simulation-Optimization; EX = Exact Optimization; P = (Maximize) Profit; ATHT = (Minimize) Average Throughput Time; LL = (Minimize) Load Leveling; TH = (Maximize) Throughput; CT = (Minimize) Cycle Time; CMT = (Minimize) Completion Time; C = (Minimize) Cost; MS = (Minimize) Makespan; FT = (Minimize) Flow Time; DO = (Minimize) Day at Operation; JS = Job-Shop; FJS = Flexible Job-Shop; SJS = Stochastic Job-Shop; SFJS = Stochastic Flexible Job-Shop; FF = Fictitious Fab; RF = Real Fab; CPM = Compare with Previous Methods.

Reference	Environment	Uncertainty	Constraints			SDST	Methodology						Objective	Case	Validation
			MPC	MD	RS		DR	HE	MH	S	EX	SO			
Wein (1988)	JS	–	–	–	–	–	–	–	–	*	–	–	ATHT	FF	–
Akcali and Uzsoy (2000)	FJS	–	*	–	*	*	–	–	–	–	–	*	P-TH	FF	–
Akcalt et al. (2001)	SJS	MF	–	*	*	*	–	–	–	*	–	–	CT	FF	–
Dabbas et al. (2001)	SJS	PT	–	–	–	–	*	–	–	–	–	–	–	FF-RF	CPM
Lee et al. (2002)	SJS	PT	–	*	*	*	*	–	–	*	–	–	LL	–	–
Yugma et al. (2007)	JS	–	–	–	*	*	–	–	–	*	–	–	DO	RF	–
Cakici and Mason (2007)	SJS	PT-NOJ	–	*	–	*	–	*	–	–	–	–	CT	RF	CPM
Johnzén et al. (2008)	JS	–	–	–	*	*	–	–	–	*	–	–	DO	RF	–
Klemmt et al. (2010)	FJS	–	*	*	*	–	–	*	–	–	*	–	C-LL-TH	RF	–
Yan et al. (2012)	JS	–	–	*	*	–	–	–	–	–	*	–	LL	–	–
Bitar et al. (2016)	SJS	PT-ST	–	–	*	*	–	–	*	–	–	–	FT	–	–
Bitar et al. (2014)	SJS	PT-ST	–	–	*	*	–	–	*	–	–	–	CT	RF	–
Ham and Cho (2015)	FJS	–	*	*	*	*	–	–	*	–	–	–	CMT-CT-U	RF	–
Zhang et al. (2018)	FJS	–	*	*	*	–	–	–	*	–	–	–	CMT	RF	CPM
This paper	SFJS	PT-ST	*	*	*	*	–	–	–	–	–	*	MS	RF	CPM

assess the effect of the proposed policies on the cycle time. In another study, Dabbas et al. (2001) developed a dispatching rule by combining several dispatching criteria such as throughput, flow control, and line balancing. Then, a previously proposed wafer fabrication model was used to compare the developed dispatching rule with single criterion ones (e.g., Critical Ratio (CR) and Shortest Processing Time (SPT)). A hypothetical "Mini-Fab" model presented by Spier and Kempf (1995) and also a model based on an actual semiconductor factory were used to validate their work. Lee et al. (2002) proposed a simulation model of a wafer fab as well as several dispatching rules. Their numerical experiments proved that pull-type scheduling rules are more effective than push-type rules for most performance measures. In another study by Cakici and Mason (2007), a mathematical model was proposed for scheduling the photolithography area while considering machine process capability and reticle sharing constraints. Processing times and the number of operations per job were assumed to follow a uniform distribution. Moreover, two heuristic algorithms were developed to solve the model. The heuristic with superior performance (H2) was identified after conducting thorough numerical experiments using real wafer fab data. Further on, they presented an enhanced version of the Tabu Search (TS) algorithm by improving neighborhood research. Bitar et al. (2016) proposed a memetic algorithm for parallel machine scheduling in the photolithography work area. They considered the reticle sharing constraints and optimized the weighted flow time (to minimize) and the number of processed products (to maximize). Their proposed metaheuristic was later on used by Bitar et al. (2014) to develop four scheduling algorithms. The algorithms were further compared using Operating Curves (OCs). OCs were developed by IBM and have been used in semiconductor manufacturing to manage the trade-off between cycle time and throughput.

As was reviewed in the previous paragraphs, the mentioned scheduling problem has been addressed by a limited number of papers. On the one hand, more than half of the reviewed papers, presented in the first subsection, ignored the uncertain nature of the photolithography work area. On the other hand, non of the papers that considered the uncertainties involved in the photolithography work area addressed all three special constraints of this work area. Within the papers reviewed in the first subsection, only three of them (Ham and Cho (2015), Klemmt et al. (2010), and Zhang et al. (2018)), addressed the scheduling problem of the photolithography work area while considering the three special constraints, simultaneously. However, as mentioned before, the work presented by these papers did not account for the uncertainties involved in the photolithography operations. Thus, as proposed by the summary Table 1, no previous work has considered the uncertain nature and special operational constraints of the photolithography work area together while addressing the scheduling problem of this work area. Furthermore, within the papers reviewed in both of the subsections, the scheduling problem of the photolithography work area has been primarily addressed using conventional approaches and optimization methods that are generally intensive in computation time, even in addressing simple, small-sized scheduling problems (Gupta and Sivakumar, 2002). Thus, considering the complexity of the scheduling problem within the photolithography work area, since simpler versions of this problem are NP-hard (e.g., Low and Fang (2005)), there is a need to apply more competent and effective tools to address this problem. In this regard and in the era of I4.0, hybrid SO methods have been one of the most promising approaches to address complex manufacturing system problems (Arakawa et al., 2003; Ghasemi et al., 2021). On one hand, simulation models have become a popular technique for developing production schedules and dispatch rules in manufacturing environments (Gupta and Sivakumar, 2002). The benefits of using simulation models to schedule the semiconductor production system have been highlighted by publications since decades ago (Li et al., 1996; Lu et al., 1994; Zhang et al., 2009). In fact, as stated by Sivakumar (2001), the simulation method helps in overcoming many of the limitations of the conventional approaches in modeling the scheduling problem of semiconductor manufacturing. On the other hand, optimization techniques are critical tools to improve decisions within almost all systems. Combining simulation models with optimization methods could develop a promising tool for solving various complex and stochastic industrial problems (Ghasemi et al., 2018) such as scheduling the photolithography work area. In this regard, some papers have used SO-based methods in the context of semiconductor manufacturing (e.g., Gupta and Sivakumar (2002), Kuck et al. (2016), Sivakumar (2001) and Waschneck et al. (2018)). However, none of them has focused on the photolithography work area.

Considering all those mentioned above and the summary provided in Table 1, there are significant gaps in the literature:

- As was stated before, developing an effective schedule for the photolithography process is essential since it can lead to substantial improvements in the overall wafer fab performance (Cakici and Mason, 2007). However, a limited number of papers have addressed the scheduling problem of this work area. As it is evident in Table 1, through the last three decades, only around 20 papers have focused on the mentioned scheduling problem.
- Even though uncertainty is an indivisible characteristic of the photolithography area (Kim et al., 2002), less than half of the

Table 2
Notations table.

Indices and sets	
$j, j' =$	Jobs indexes, $j, j' \in \{1, \dots, N\}$.
$i, i' =$	Operations ids, $i, i' \in \{1, \dots, NO\}$.
$o, o' =$	Operations indices for job j and job j', respectively, $o \in \{1, \dots, NO_j\}$,$o' \in \{1, \dots, NO_{j'}\}$.
$m =$	Machine indexes, $m \in \{1, \dots, NM\}$.
$k =$	Queuing position index, $k \in \{1, \dots, NO\}$.
$PD =$	Precedence orders sets defining the execution precedence of operations of the same jobs.
$s =$	Simulation replication index, $s \in \{1, \dots, SL\}$.
$It =$	GA iteration index, $It \in \{1, \dots, T\}$.
Parameters	
N	Total Number of jobs.
NO	Total number of operations.
NM	Total Number of machines.
NO_j	Total number of operations for job j.
e_j	First critical operation of job j.
pd_i	The precedence orders set that operation id i is a member of.
SL	Number of simulation replications indexed by s.
T	Total number of GA iterations indexed by It.
A_{oj}	Set of alternative machines capable of processing operation o of job j.
B_m	An index set of operations that can be processed on machine m, $\{(o, j); o$ such that $m \in A_{oj}\}$.
M_{im}	Alternative machine, 1 if machine m is capable of processing operation i, 0, otherwise.
t'_{ojm}	Stochastic processing time of operation o of job j on machine m.
$d'_{ojo'j'}$	Stochastic sequence-dependent setup time between operation o of job j and operation o' of job j' on machine.
Cr_{oj}	Critical operation, 1 if operation o of job j is critical, 0, otherwise.
P'_i	Stochastic processing time of operation id i.
Q'_i	Stochastic sequence-dependent setup time of operation id i.
ϕ_{im}	Probability distribution of processing time of operation id i.
$\phi'_{ii'}$	Probability distribution of sequence-dependent setup time between operation ids i and i'.
Fl_{oj}	1, if there are more that one machines in A_{oj}, 0, otherwise.
$St_{ii'}$	1, if $\phi'_{ii'} > 0$, 0, if $\phi'_{ii'}$=0.
M	A large number.
Decision variables	
Y_{ojm}	1, if operation o of job j is processed on machine m, 0, otherwise.
S^P_{oj}	The starting time of operation o of job j.
$X_{ojo'j'm}$	1, if operation o of job j is to be processed right after operation o' of job j', 0, otherwise.
Z_{ik}	1, if operation id i is assigned to the kth position of the dispatching queue, 0, otherwise.
Functions	
C_{max}	The makespan of a schedule.
f_s	The objective calculation function for the simulation replication s.

presented papers have considered the stochastic nature of this work area. Also, regarding the three special constraints of this workstation, only three papers have taken them into account simultaneously. However, these three papers ignored the uncertainty in the photolithography area. In other words, no previous work has considered all the characteristics of this work area simultaneously.

- As it was mentioned earlier, the majority of the methods used to address the scheduling problem of the photolithography work area are traditional optimization methodologies. These methods are not capable enough to address such complex problems within a reasonable time.
- Moreover, as it is evident from Table 1, there is also a lack of validation and sensitivity analysis strategies within the reviewed literature. In other words, only a few papers have validated their solution approach by conducting relative comparisons. The same goes for presenting a sensitivity analysis to evaluate the performance level of the photolithography work area under different conditions.

This paper aims to fill the mentioned gaps by addressing the scheduling problem of the photolithography work area, considering all its characteristics while using a competent SO method. The proposed SO method is also validated through extensive comparative experiments. Moreover, an insightful sensitivity analysis is presented to assess the performance of the understudied work area under different conditions.

3. Problem formulation

In this section, the SFJSSP of the photolithography work area is formulated. The mathematical model developed for the SFJSSP of the photolithography work area is elaborated in the first subsection. Further on, transforming the mathematical model into an SO model is explained thoroughly in Section 3.2. Table 2 provides a summary of the notations used in this section. Before introducing the model formulation, the following example is proposed that represents a scheduling problem in the mentioned work area. This example is solved gradually to provide the reader with a better understanding of the SO mathematical model. Consider *Job1*, *Job2*, and *Job3* presented in Fig. 2 for which $NO_1 = 3$, $NO_2 = 3$, and $NO_3 = 1$, respectively. One of the operations is chosen to explain the operations' sequencing and assignment procedure. In this regard, consider *Op6* (note in this article we sequentially number the operations across starting from the lowest number job to the highest number job), which is the third operation of *Job2*. To satisfy the machine process capability constraints as one of the three special constraints of the photolithography work area, a set of machines are considered to process *Op6*. In this regard, the processing times of *Op6* are 8 and 9 time units on *Machine2*, and *Machine3*, respectively. Moreover, with respect to the third special constraint of the photolithography workstation, maximum reticles (masks) sharing constraints, sequence-dependent setup times are also considered for shifting from *Op6* to *Op1* and *Op7* with 1 and 3 time units, respectively (the rationale behind considering sequence-dependent setup times to satisfy the maximum reticles (masks) sharing constraints will be elaborated in the following subsection). Noteworthy is to mention that, according to previous production data provided by Ghasemi et al. (2020), the processing times are known but they follow a probability distribution. However, in Fig. 2, we ignored the stochastic factors in both processing times and sequence-dependent setup times for illustrative purposes. Furthermore, *Op6* is the second **critical** operation of *Job2* (this information will be used further on to satisfy the second special constraint of the photolithography work center, machine dedication constraints). To represent this scheduling problem, a mathematical model is presented in the next subsection.

3.1. Mathematical model

This subsection proposes the mathematical model for the SFJSSP of the photolithography work area. Noteworthy is mention that the following mathematical model is developed based on the work of Choi and Choi (2002) who proposed a mathematical model for an SFJSSP. However, some of the notations are modified to implement the model in the photolithography work area. Furthermore, constraint (9) is added to the model to represent the machine dedication constraints of the photolithography work.

As the primary goal of the proposed model is to minimize the makespan of the schedule, according to Choi and Choi (2002), the objective function can be formulated as follows:

$$Min\ F = Min(C_{max}) \tag{1}$$

Also, Choi and Choi (2002) presented the following set of constraints:

$$S^P_{oj} + t'_{ojm}.Y_{ojm} \le S^P_{o+1,j}, j \in \{1,\dots,N\}; o \in \{1,\dots,NO_j\}; m \in A_{oj}; \tag{2}$$

$$S^P_{NO_j j} + t'_{NO_j jm}.Y_{NO_j jm} \le C_{max}, j \in \{1,\dots,N\}; m \in A_{NO_j}; \tag{3}$$

$$S^P_{oj} + t'_{ojm} + d'_{ojo'j'} \le S^P_{o'j'} + (1 - X_{ojo'j'm}) \times M, (o,j) \in B_m; (o',j') \in B_m; m \in \{1,\dots,NM\}; \tag{4}$$

$$\sum_{m\in M_{oj}} Y_{ojm} = 1, j \in \{1,\dots,N\}; o \in \{1,\dots,NO_j\}; m \in A_{oj}; \tag{5}$$

$$\sum_{(o',j')\in B_m} X_{ojo'j'm} = Y_{ojm}, j \in \{1,\dots,N\}; o \in \{1,\dots,NO_j\}; m \in A_{oj}; \tag{6}$$

$$\sum_{(o,j)\in B_m} X_{ojo'j'm} = Y_{o'j'm}, j' \in \{1,\dots,N\}; o' \in \{1,\dots,NO_j\}; m \in A_{o'j'}; \tag{7}$$

$$S^P_{oj} \ge 0, j \in \{1,\dots,N\}; o \in \{1,\dots,NO_j\}; \tag{8}$$

As mentioned earlier, the following constraint is added to the original model proposed by (Choi and Choi, 2002):

$$\sum_{o\in NO_j} Y_{ojm} \times Cr_{oj} = \sum_{o\in NO_j} Y_{e_j jm} \times Cr_{oj}, j \in \{1,\dots,N\}; m \in A_{oj}; \tag{9}$$

Constraints (2) and (3) account for the precedence relationships of the operations of the same job. According to constraint (2), an operation cannot start processing unless its predecessor operation is processed completely. Constraint (3) further denotes that the completion time of each job cannot exceed the makespan of the schedule. According to constraint (4), each machine can only process one operation at a time. Constraint (4) also accounts for the maximum reticle sharing constraints of the photolithography work area. In a wafer fabrication process, an average of 20 to 30 reticle changes must be performed to process a specific lot (Cakici and Mason, 2007). These changes give rise to the need for sequence-dependent setup times (Park and Stefanski, 1998). Note the setup occurs just once when the machine starts processing operations requiring a new reticle (Ham and Cho, 2015). Thus, the maximum reticle sharing constraints have been considered in the model by including sequence-dependent setup times in constraint (4). Constraint (5), represents the machine process capability constraints and forces exactly one alternative machine option to be selected from B_m. Constraints (6) and (7) denote circular permutations of operations on each machine. Also, according to constraint (8), each job can be available at time zero. Constraint (9) accounts for the machine dedication constraints. According to constraint (9), all critical operations of each job must be processed on the same machine (the machine that processed the first critical operation of that job). Finally, according to Choi and Choi (2002), variable types and domains constraints are demonstrated in the following:

$$Y_{ojm} \in \{0,1\}, j \in \{1,\dots,N\}, o \in \{1,\dots,NO_j\}; m \in A_{oj}; \tag{10}$$

$$X_{ojo'j'm} \in \{0,1\}, (o,j) \in B_m; (o',j') \in B_m; m \in \{1,\dots,NM\}; \tag{11}$$

To solve the proposed SFJSSP of the photolithography work area, the developed mathematical model must be transformed into an SO model first. The underlying reason behind this transformation is that production scheduling problems are among the most complex optimization problems (Xiong and Zhou, 1998). Therefore, conventional search and optimization methods are ineffective in addressing these problems. This is mainly due to the fact that these methods are generally intensive in computation time as even the simple manufacturing scheduling problems are NP-hard (Gupta and Sivakumar, 2002). The complexity of scheduling problems increases even more in semiconductor manufacturing due to the presence of various work centers, special process constraints, sequence-dependent setup times, re-entrant process flow, and the stochastic nature of the problem. In this regard, SO methods have been one of the most promising approaches to addressing complex manufacturing system problems in a reasonable computation time (Linnéusson et al., 2020).

The following subsection is dedicated to transforming the proposed mathematical model into an SO model.

3.2. SO mathematical model transformation

This subsection presents the transformation of the previously proposed mathematical model into an SO. An SO consists of a simulation model and an optimization algorithm. Both the former and latter satisfy a subset of the constraints presented in Section 3.1. To be more detailed, some variables/parameters of the scheduling problem cannot be regarded as fixed and known in an stochastic environment. Simulation is one of the most suitable ways to derive experience-based solutions in these stochastic environments (Longo, 2010). Thus, in our SO approach, the simulation model is in charge of dealing with stochastic parameters. In this regard, all the previously elaborated constraints, consisting of stochastic parameters, are satisfied through the simulation model. In other words, the constraints of the proposed mathematical model are divided into two groups: group one, which contains stochastic parameters, and group two, with all deterministic parameters. In this context, the simulation model is in charge of dealing with the first group while the second group of constraints is satisfied by the optimizer. Consider the previously presented example in Fig. 2 for further elaboration.

As it can be seen in Fig. 2, *Op6* is assigned to the 6th position of the "Job-Shop Queue (Queue)". Considering the precedence constraints, *Op6* must be positioned after *Op4* and *Op5* (as it can be seen in the Queue). Thus, we denote that $Z_{42} = 1$ and $Z_{53} = 1$. The "Job-Shop Queue (Queue)" in Fig. 2 represents a feasible solution for the scheduling problem. That is to say, the operations are positioned in the Queue according to their precedent-dependent relationships. In fact, the "Job-Shop Queue (Queue)" can be seen as one of the solutions provided by the optimization algorithm for the scheduling problem. These alternative solutions, with different sequencing of operations, need to be tested and compared to find a good (or optimal) solution (Figueira and Almada-Lobo, 2014). In this setting, the simulation model calculates the objective value of each optimization solution. In other words, objectives (makespan of the solutions in this paper) are variable process parameters of the simulation model that result from a simulation run (Krug et al., 2002). The simulation model builds

Operations Environment (Inputs)

Job ID (j)	Operation ID (i)	IsCritical (Cr_i)	Stochastic processing times ($\emptyset_{im}$)			Stochastic sequence-dependent setup times ($\emptyset'_{ii'}$)						
			Machine1	*Machine2*	*Machine3*	*Op1*	*Op2*	*Op3*	*Op4*	*Op5*	*Op6*	*Op7*
Job1	***Op1***	0	-	8	9	-	-	-	2	0	1	0
	Op2	0	8	7	-	0	-	-	1	0	0	3
	Op3	0	-	7	8	0	0	-	1	2	0	1
Job2	***Op4***	1	6	-	6	2	1	1	-	-	-	0
	Op5	0	5	5	7	0	0	2	0	-	-	2
	Op6	1	-	8	9	1	0	0	0	0	-	3
Job3	***Op7***	0	8	-	9	0	3	1	0	2	3	-

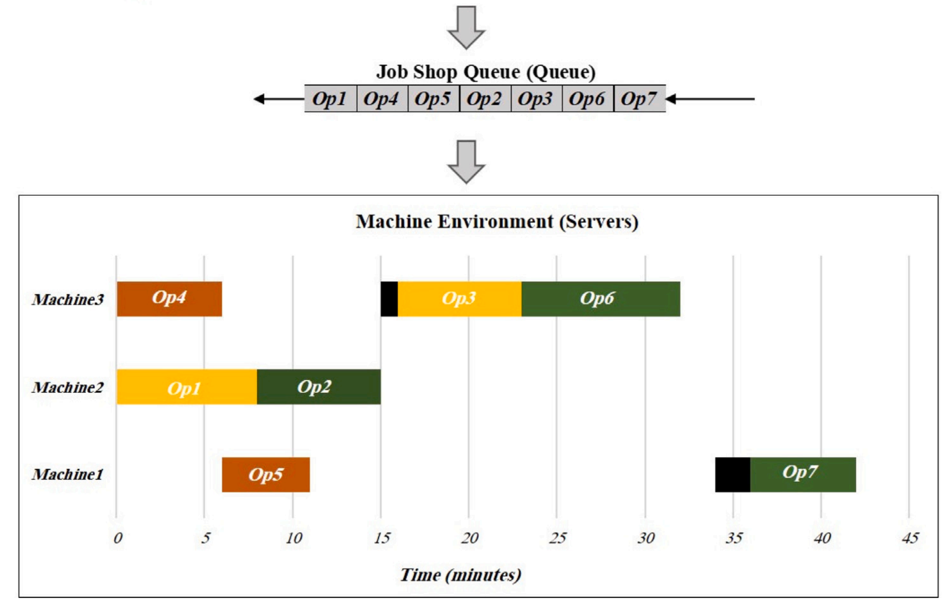

Fig. 2. Proposed Job-Shop scheduling problem example.

the schedule by sequencing the operations on machines' queues. Due to the flexibility of the understudied scheduling problem, there is a set of alternative machines to process each operation. Therefore, the simulation model applies the Least Work Remaining (LWR) dispatching rule to select a machine from the set of alternative machines capable of processing each operation. Once an operation finishes processing, the job moves to the next machine, and the dispatching rule is called again. When all operations are executed, the final makespan is then returned. In fact, the simulator receives as input a feasible solution (provided by the optimizer) and a Job-Shop problem instance and returns the respective makespan. However, since the problem is stochastic, one replication should not be enough to accurately evaluate the performance of each solution (Figueira and Almada-Lobo, 2014). Thus, Eq. (1) is transformed to Eq. (12), where f_s calculates the makespan of solution Z_{ik} from simulation replication s. Also, in this context, F defines the fitness value for an SFJSSP solution.

$$Min\ F = Min\left(\frac{1}{SL}\sum_{s=1}^{SL} f_s\left(\{\cup_{i=1}^{NO}\cup_{k=1}^{NO} Z_{ik}\}, P'_i, Q'_i\right)\right) \quad (12)$$

Some constraints need to be taken into consideration regarding the simulation model. In each solution, Z_{ik} is a binary variable that shows whether operation i is assigned to the kth position of the "Job-Shop Queue (Queue)". Thus, each operation should be assigned to one of the existing positions in the Queue:

$$\sum_{k\in NO} Z_{ik} = 1; i \in \{1,\ldots,NO\} \quad (13)$$

The following constraints ensure that for each existing position in dispatching queue k, at most one operation is assigned.

$$\sum_{i\in NO} Z_{ik} = 1; k \in \{1,\ldots,NO\} \quad (14)$$

Also, the precedence relationship between two operations i and i' from the same job j in the scheduling problem needs to be considered. In other words, when operation i (assigned to the position k) precedes operation i', operation i' must be assigned to a position k' ($k' >$ k) on the dispatching queue:

$$Z_{ik} \geq Z_{i'k'}; i, i' \in PD, k, k' \in \{1,\ldots,NO\}, k < k' \quad (15)$$

$$Z_{ik} - Z_{ik} \times Z_{i'k'} \geq Z_{i'k'}; i, i' \in PD, k, k' \in \{1,\ldots,NO\}, k > k' \quad (16)$$

Finally, the decision variable feature of the model can be defined as the following constraint:

$$Z_{ik} \in 0, 1; i, k \in \{1,\ldots,NO\} \quad (17)$$

Moreover, in addition to constraints (13) to (17), constraint (9) presented in Section 3.1 also needs to be taken into consideration.

According to all the above-mentioned, to finish solving the example provided in Fig. 2 consider the "Job-Shop Queue (Queue)", which is the output of the optimization algorithm. This Queue represents a feasible solution for the scheduling problem and is received by the simulation model as input. Also, the data in the operation environment table in Fig. 2 is the other input of the simulation model. In this stage, the simulator starts the scheduling process to calculate the fitness value. Since *Op6* is the second **critical** operation of *Job2*, it must be assigned to the same machine as the first critical operation of *Job2* to satisfy the machine dedication constraints. Since the first operation of *Job2*, which is *Op4*, had been assigned to *Machine3*, *Op6* needs to be assigned to *Machine3* as well. Thus, as shown in the Gantt chart, *Op4* and *Op6* are both processed on *Machine3*. Conducting the same assignment method for all the existing operations, each operation is assigned to a machine, and the decision variables are calculated. Consequently, the amount of objective function/fitness value, which is the makespan in this paper, is defined.

Algorithm 1: Scheduling algorithm.

Inputs : Z (feasible solution), $OpJob_i$ (the job that operation id i belongs to), ϕ_{im}, $\phi'_{ii'}$, N, NO, NM
Output: $Makespan$
for $j = 1$ **to** N **do**
  $FirstCriticalOpMachine_j = NAN$
  $PreviousOpFinishTime_j = 0$
  $CompletionTime_j = 0$
end
for $m = 1$ **to** NM **do**
  $MachineFreeAt_m = 0$
end
for $K = 1$ **to** NO **do**
  i = the operation in the kth position of Z
  $j^* = OpJob_i$
  if *i is not a critical operation of j^** **then**
    m^* = the machine from the machine set (satisfying the machine process capability constraints by considering a machine set to process each operation) that can operate i with the minimum of remaining work
  end
  if *i is the first critical operation of j^** **then**
    m^* = the machine from the machine set that can operate i with the minimum of remaining work
  end
  if *i is a critical operation of j^* but i is not the first critical operation of j^* (satisfying the machine dedication constraints)* **then**
    $m^* = FirstCriticalOpMachine_{j^*}$
  end
  if *i is the first operation to be processed by m^* or the last operation that had been processed by m^* also belongs to j^* (satisfying the maximum reticles (masks) sharing constraints)* **then**
    $Q'_i = 0$
  else
    $Q'_i = Rand(\phi'_{ii'})$
  end
  $OpStartTime_i = max(MachineFreeAt_{m^*} + Q'_i, PreviousOpFinishTime_{j^*})$
  $P'_i = Rand(\phi_{im^*})$
  $CompletionTime_{j^*} = OpStartTime_i + P'_i$
  $MachineFreeAt_{m^*} = OpStartTime_i + P'_i$
  $PreviousOpFinishTime_{j^*} = OpStartTime_i + P'_i$
end
$Makespan = max(CompletionTime_j \quad \forall j \in \{1, .., N\})$
Return $Makespan$

4. The proposed SO algorithm

This section presents the proposed SO algorithm to solve the SFJSSP of the photolithography work area. Hybrid SO-based approaches have proven their potential in practical applications of production and logistics systems through the years (März et al., 2011). An SO structure consists of a simulation model and an optimization algorithm. In this regard, a simulation model of the photolithography work area and a GA-based optimization algorithm are presented in the following subsections.

4.1. The proposed simulation model

Due to the competencies of simulation models in scheduling complex and stochastic manufacturing systems (Aydt et al., 2009), a simulation model of the photolithography work area is proposed by employing Python programming language. As mentioned in Section 3, the simulation model plays a critical role within the proposed SO structure. According to Table 1, the developed simulation model is the only model in the literature that considers all three special constraints of the photolithography work area simultaneously and respects the stochastic nature of this work center. Moreover, the developed simulation model is widely flexible to represent various scheduling problems which are solved using the proposed simulation model in Section 6.

A feasible solution for the scheduling problem, the number of jobs, the number of operations of each job and the precedence relationship among them, the set of machines capable of operating each of the operations, and the processing times of the operations on each of the capable machines are the input parameters of the simulation model. The simulation model, also, receives data on the critical operations and the sequence-dependent setup times required to process the operations of different jobs on the same machine, consecutively. In other words, all the data included in the Operations Environment and the Job Shop Queue (Queue) presented in Fig. 2 are given to the simulation model as the input parameters. The simulation model, therefore, is in charge of three main tasks within the SO structure:

- Dealing with the stochastic nature of the constraints of the SFJSSP of the photolithography work area. In fact, simulation models are one of the most competent tools to deal with stochasticity in manufacturing systems. Thus, one of the main applications of the proposed simulation model is to satisfy a subset of the constraints presented in Section 3 which contain stochastic parameters.
- Calculating the fitness value for each feasible solution obtained by the optimizer. As mentioned in Section 3, the optimization algorithm produces alternative feasible solutions for the scheduling problem. The fitness value (the makespan) is employed to evaluate the performance of these solutions. The stochastic nature of the photolithography work area renders the use of the simulation model for calculating the fitness value.
- Developing an initial schedule that is iteratively optimized by the optimization algorithm. The optimizer is in charge of developing a chromosome of operations (a feasible solution). In the next step, the proposed simulation model is responsible for assigning each operation to a competent machine using the LWR dispatching rule. Each operation is assigned to a machine, and an initial schedule is developed.

A more detailed elaboration of the scheduling process by the simulation model is illustrated in algorithm 1. Also, the proposed algorithm illustrates how the three special constraints of the photolithography work area are taken into consideration in the developed simulation model.

4.2. The proposed optimization algorithm

This section proposes the developed GA as the optimizer of the SO algorithm. GA imitates the phenomenon of biological evolution and natural selection, where the fittest individuals are selected for reproduction to produce offspring of the next generation. Over the previous two decades, the superior search capability (Chang and Liu, 2017), and short computation time of GA (Chen et al., 2016), have led to the refinement of numerous GA-based approaches for production scheduling optimization problems (e.g., Gong et al. (2014), Kawanaka et al. (2001) and Tay and Wibowo (2004)). Commonly, GA starts with a predefined size of the population, known as the *Initial Population*, composed of a certain number of individuals. These individuals are encoded as chromosomes and represent a feasible solution to the optimization problem (Zhang et al., 2011). Furthermore, three main operators are employed in GA to direct the population to the global optimum: selection, crossover, and mutation. The selection operator is applied to the *Initial Population* to choose the chromosomes for reproduction. Afterward, the crossover and mutation operators are adopted to form a new population from the selected population. This process continues until a termination condition is met (Defersha and Chen, 2009).

Key components of the proposed GA are presented in the following subsections. Fig. 3 showcases the incorporation of GA components and

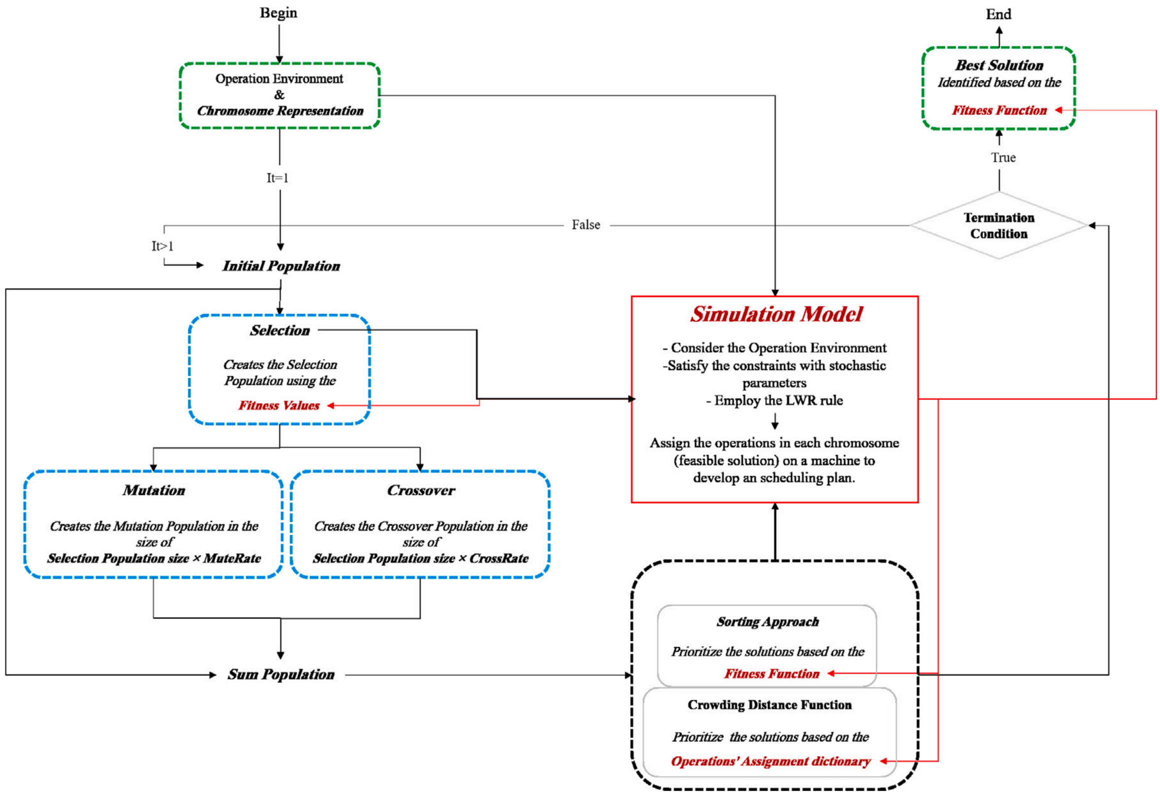

Fig. 3. SO structure.

the simulation model. The diagram highlights the inputs and output of the SO model through green boxes. Additionally, blue boxes represent the GA operators, while the process of selecting the population for the next iteration of the GA is depicted in a black box. Lastly, the red-colored elements signify the simulation model and its output, which are utilized by other components of the SO model.

4.2.1. Chromosome representation

Encoding a feasible solution to a chromosome is a key aspect in solving scheduling problems with GA (Djerid and Portmann, 1996). This paper presents each chromosome in a single-dimensional array with the same length as NO. Each chromosome's gene contains a unique Operation ID (as presented in the table of Operation Environment in Figure 1) that indicates one operation of a job. These IDs are assigned to the operations, starting from the first operation of the first job to the last operation of the last job. The "Job-Shop Queue (Queue)" presented in Figure 1 can be considered a sample of the proposed chromosome representation.

4.2.2. Population initialization

The fundamental underlying mechanism to start the search in GA initiates with a population of individuals (Driss et al., 2015). This *Initial Population* is crucial for GA since it directly influences the convergence rate of the fitness values and the ultimate quality of the optimal solutions (Chang and Liu, 2017). In this paper, the *Initial Population* is created by sequencing the operations randomly in each chromosome while preserving the precedent constraints of the operations of the same job. As illustrated in Fig. 3, the way that each solution is represented (Chromosome Representation) and the data regrading the specifications of the operations are used to create the *Initial Population*.

4.2.3. Genetic operators

As mentioned earlier, there are mainly three genetic operators: selection, crossover, and mutation. Genetic operators of the proposed GA are described in the following.

4.2.3.1. Selection. The selection operator is in charge of choosing individuals for reproduction (Driss et al., 2015). We adopted a selection operator by modifying the binary tournament method introduced by Brindle (1980). As Fig. 3 shows, the selection operator is applied on the *Initial Population*. The proposed selection operator chooses the best chromosomes from the *Initial Population* based on their fitness values (makespans) following the subsequent steps:

- Step 1: Two chromosomes are chosen randomly from the *Initial Population*.
- Step 2: Due to the stochastic nature of our scheduling problem, one fitness value is not enough to accurately evaluate the performance of each chromosome. Thus, for the chosen chromosomes in Step 1, the simulation model is independently executed *SL* times using the same chromosome as an input. The simulation runs result in a list of *SL* fitness values (makespans) for each chromosome.
- Step 3: The lists obtained from Step 2 are combined and sorted in an ascending order, resulting in a new sorted list.
- Step 4: In this step, we allocate a score value to each chromosome based on the sorted list derived from Step 3. It is important to note that each element in the sorted list corresponds to a fitness value obtained initially, in Step 2, by executing the simulation model with one of the two randomly selected chromosomes from Step 1. During this step, our focus lies on the first half of the sorted list, which represents the top-performing half of all fitness values obtained in Step 2. Within this range, starting from the first fitness value, we identify the corresponding chromosome and increment its score value by one (score values are initially set to zero). The same process is repeated for all items within the first half of the sorted list, resulting in the final score values for the chromosomes. To illustrate the scoring procedure, consider a solution population consisting of n chromosomes, labeled as X_i where i ranges from 1 to n. Each chromosome X_i undergoes a series of SL simulations, obtaining various objective values y_i^1, y_i^2, to y_i^{SL}. These objective

values calculated using simulation experiments are stored in a list. The list of objective values is then sorted in ascending order, and only the first half of the sorted list is considered. Finally, the score of each chromosome (solution) x_i refers to the number of its objective values being inside this subset (i.e., the score is obtained by counting the number of y_i in the subset for each solution x_i).

- Step 5: A comparison is conducted between the score values assigned to the chromosomes in Step 4, utilizing a selection threshold. The chromosome with a score value greater than or equal to the product of the selection threshold and the length of the first half of the sorted list is considered the fitter chromosome. If neither chromosome meets the criteria to be considered fitter, we go to Step 1.

The above-mentioned steps are repeated until the desired number of chromosomes are selected to form the *Selection Population*.[1] A detailed explanation of the selection operator is provided in algorithm 2.

Algorithm 2: Selection Operator

```
Inputs : InPop (Initial Population), PopSize (favorable number of
         chromosomes in Selection Population), SL, Selection Threshold
Output: SelPop (Selection Population)
SelPop = NAN
FitList_sorted = ()
while number of chromosomes in SelPop ≠ PopSize do
    Ch_1 = A chromosome randomly selected from InPop
    Ch_2 = A chromosome randomly selected from InPop
    FitList_Ch1 = ()
    FitList_Ch2 = ()
    Score_Ch1 = 0
    Score_Ch2 = 0
    for s = 1 to SL do
        FitValue_Ch1 = the fitness value for Ch_1 calculated by the
         simulation model
        FitList_Ch1 = FitList_Ch1 + FitValue_Ch1
        FitValue_Ch2 = the fitness value for Ch_2 calculated by the
         simulation model
        FitList_Ch2 = FitList_Ch2 + FitValue_Ch2
    end
    FitList_sorted = FitList_Ch1 + FitList_Ch2 and sorted in ascending
     order
    for i = 1 to len(FitList_sorted)/2 do
        if FitList_sorted(i) ∈ FitList_Ch1 then
            Score_Ch1 = Score_Ch1 + 1
        else
            Score_Ch2 = Score_Ch2 + 1
        end
    end
    if Score_Ch1 ≥ Selection Threshold × len(FitList_sorted)/2 then
        SelPop = SelPop + Ch_1
    end
    if Score_Ch1 ≥ (Selection Threshold × len(FitList_sorted)/2) then
        SelPop = SelPop + Ch_2
    end
end
Return SelPop
```

4.2.3.2. Crossover. The crossover operator aims to obtain better chromosomes by exchanging information contained in the currently selected ones. During the last decades, several types of crossover operators have been introduced, which do not respect the precedent constraints of the operations (Chen et al., 2020). Thus, it is necessary to apply a correcting algorithm to modify infeasible offspring created by these operators. However, adopting a correcting algorithm is time-consuming, and hence it is preferable to design operators such that precedence constraints are not violated (Pezzella et al., 2008). Therefore, in this paper, precedence preserving order-based crossover (POX), firstly proposed by Lee et al. (1998), is used. POX is among the superior crossover operators (Jiang and Zhang, 2018; Zhang et al., 2011), and respects the precedence constraints of the operations of the same job. In this regard, the proposed crossover operator randomly selects two chromosomes (known as parents) from the *Selection Population*. Then, an operation (i) is randomly chosen from the first parent. In the next step, the precedent orders set to which i belongs is identified (pd_i). Subsequently, all the operations of pd_i are placed in an empty chromosome (known as the child) in the same place as they appear in the first parent. The operator then completes the child with the remaining operations in the same order as they appear in the second parent. This procedure is repeated until the favorable number of offspring is created and the *Crossover Population* is formed.

Fig. 4 presents how the crossover operator functions in four steps. Moreover, algorithm 3 is provided for further elaborations.

Algorithm 3: Crossover Operator

```
Inputs : SelPop (Selection Population), PopSize (number of
         chromosomes in the Selection Population), NO, pd_i, CrossRate
Output: CrossPop (Crossover Population)
CrossPop = NAN
for p = 1 to (PopSize × CrossRate) do
    Parent_1 = A chromosome randomly selected from SelPop
    Parent_2 = A chromosome randomly selected from SelPop
    Child = NAN
    k = Rand(0, NO)
    i = Parent_1(k)
    for Op ∈ pd_i do
        k = find(k|Parent_1(k) = Op)
        Child(k) = Op
    end
    for k = 1 to NO do
        if Parent_2(k) ∉ Child then
            for k' = 1 to NO do
                if Child(k') == NAN then
                    Child(k') = Parent_2(k)
                    break!
                end
            end
        end
    end
    CrossPop = CrossPop + Child
end
Return CrossPop
```

4.2.3.3. Mutation. The mutation operator is also an important part of GA. This operator can enhance the diversity of the offspring population by introducing some additional variability into it (Driss et al., 2015). In this way, the local optimum phenomena can be avoided (Chen et al., 2020). This paper develops a mutation operator based on the Precedence Preserving Shift mutation (PPS), firstly presented by Lee et al. (1998). This operator also respects the precedence-dependant relationships among the operations of the same job. Firstly, a chromosome (known as the parent) is randomly chosen from the *Selection Population*. Afterward, an operation (i) is randomly selected from the parent chromosome. In the next step, using the precedent orders set to which i belongs (pd_i), the predecessor and successor operations of i are identified. The positions where these two operations are placed in the parent chromosome are also identified. A position (k) is randomly selected between these two positions and the child chromosome is created by moving i to k. This procedure is repeated until the favorable

[1] The literature on SO approaches in stochastic environments often employs the average of simulation replications for selecting chromosomes. However, later in Appendix, employing the experiments presented in Sections Section 6.1.2 and Section 6.2.2, it is shown that the proposed sorting methodology by this article outperforms the sorting approach using the average of the simulation replication results as the criterion

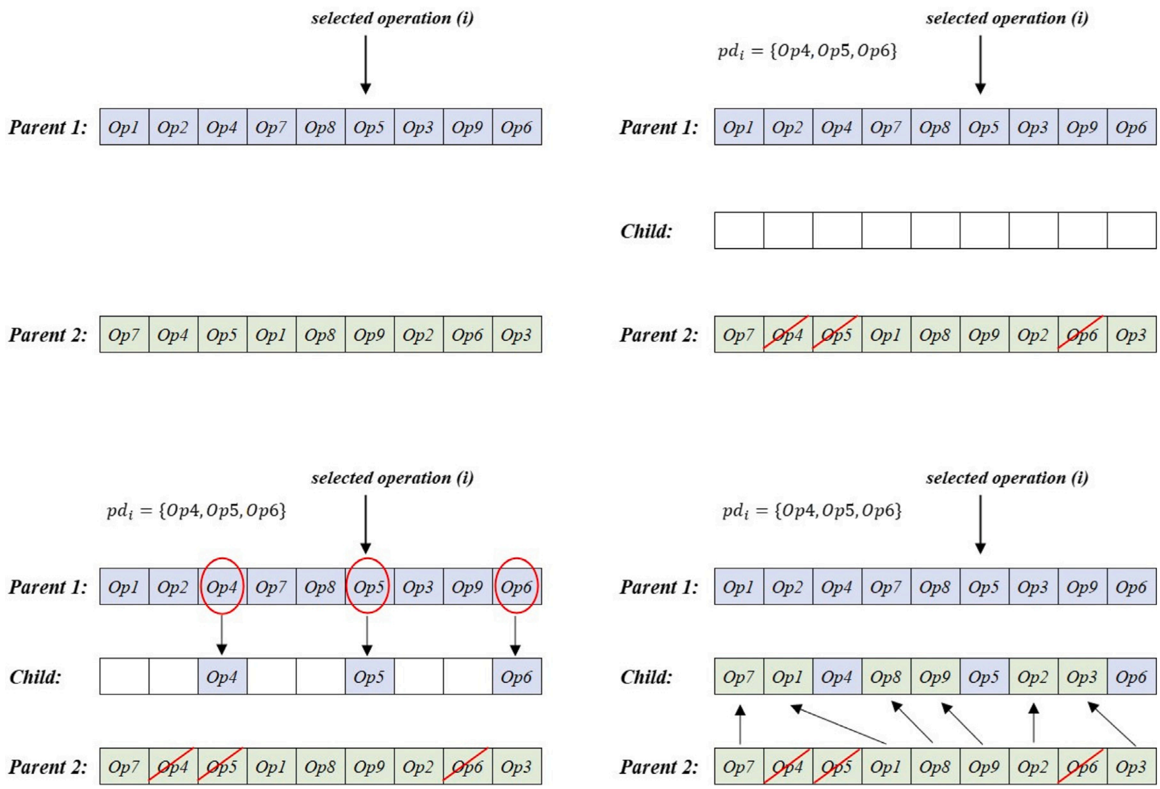

Fig. 4. Crossover operator.

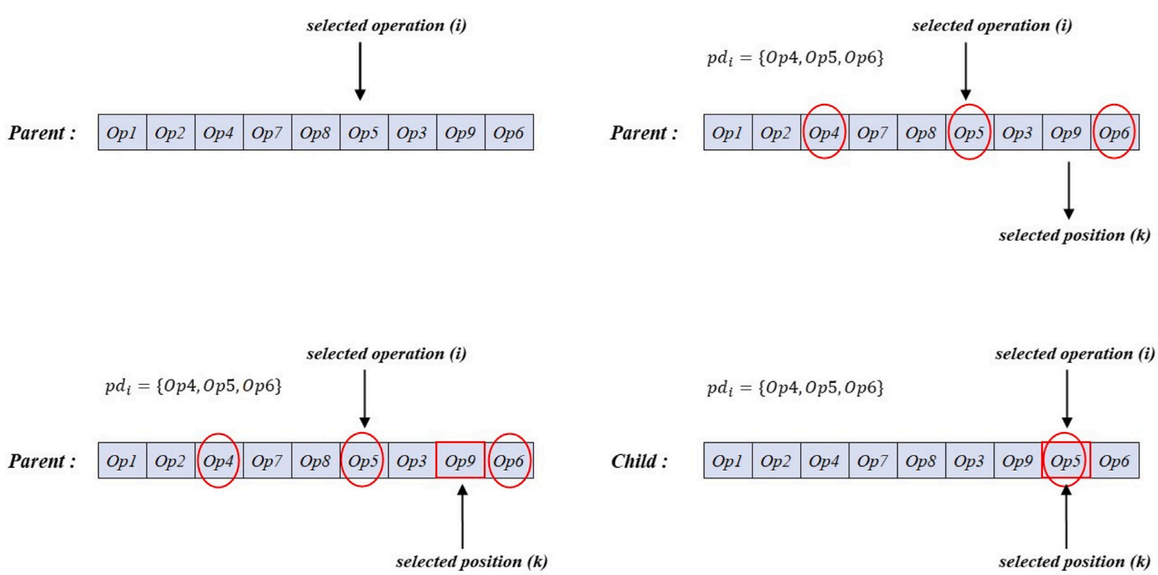

Fig. 5. Mutation operator.

number of offspring are created and the *Mutation Population* is formed. Fig. 5 represents the mutation operator. Moreover, algorithm 4 is provided for further elaborations.

It is noteworthy to mention that, in this paper, the offspring population is formed merely by the crossover and mutation operators. As mentioned in the boxes representing the *Crossover Population* and the *Mutation Population* in Fig. 3, the quotas of the offspring created by each of these operators are calculated using $CrossRate$ and $MuteRate$, respectively. These rates are applied to the *Selection Population* size to reach the exact number of chromosomes.

4.2.4. Next population selection

In the presented GA, the *Initial Population* to start the first iteration of the algorithm ($It = 1$) is created as explained in Section 4.2.2. However, starting from the second iteration ($It > 1$), a method is employed to choose the elite solutions from the summation of the *Initial*

Algorithm 4: Mutation Operator

```
Inputs : SelPop (Selection Population), PopSize (number of
         chromosomes in the Selection Population), pd_i, NO, MuteRate
Output: MutePop (Mutation Population)
MutePop = NAN
for p = 1 to (PopSize × MuteRate) do
    Parent = A chromosome randomly selected from SelPop
    Child = NAN
    Test = NAN
    k' = Rand(0, NO)
    i = Parent(k')
    k'' = find(k''|pd_i(k'') = i)
    if i == the last operation of pd_i then
        predecessor = Parent(k'' − 1)
        successor = i
    end
    if i == the first operation of pd_i then
        predecessor = i
        successor = Parent(k'' + 1)
    else
        predecessor = Parent(k'' − 1)
        successor = Parent(k'' + 1)
    end
    l = find(l|Parent(l) = predecessor)
    l' = find(l'|Parent(l') = successor)
    if (l' − l) ≤ 1 then
        Child = Parent
        MutePop = MutePop + Child
    else
        k = Rand(l, l')
        if k ≥ k' then
            Child(1 : k') = Parent(1 : k')
            Child = Child + Parent(k' + 1 : k)
            Child = Child + i
            Child = Child + Parent(k + 2 : NO)
            MutePop = MutePop + Child
        end
        if k == k' then
            Child = Parent
            MutePop = MutePop + Child
        end
        if k < k' then
            Child(1 : k − 1) = Parent(1 : k − 1)
            Child = Child + i
            Test = Parent(k : NO) − i
            Child = Child + Test
            MutePop = MutePop + Child
        end
    end
end
```

Population, *Crossover Population*, and *Mutation Population* (referred to as the *Sum Population* in Fig. 3) to initiate the subsequent iterations. The aforementioned method consists of a sorting approach similar to the one used in the selection operator and a distancing function presented by Xiong et al. (2012). Both the former and the latter are explained in the following subsections.

4.2.4.1. Sorting approach. The sorting approach used in this paper to prioritize the solutions for selecting the *Next Population*is conducted based on the amounts of fitness values obtained in different simulation runs. The following represents the main steps conducted by the proposed sorting approach:

- Step 1: Due to the stochastic nature of our scheduling problem, one fitness value is not enough to accurately evaluate the performance of each chromosome. Thus, to conduct a valid comparison among the chromosomes of the *Sum Population*, based on the fitness value, the simulation model is run independently for *SL* times for each of the chromosomes of the *Sum Population*. The resulting fitness values are recorded in separate lists for each chromosome.
- Step 2: The lists obtained from Step 1 are combined and sorted in an ascending order, resulting in a new sorted list.
- Step 3: In this step, similar to Step 4 of the selection operator, we allocate a score value to each chromosome based on the sorted list derived from Step 2. Each element in the sorted list corresponds to a fitness value obtained initially, in Step 1, by executing the simulation model with one of the chromosomes from the *Sum Population*. During this step, we focus on the first half of the sorted list, which represents the top-performing half of all fitness values obtained in Step 1. Within this range, starting from the first fitness value, we identify the corresponding chromosome and increment its score value by one (score values are initially set to zero). The same process is repeated for all items within the first half of the sorted list, resulting in the final score values for all of the chromosomes of the *Sum Population*.
- Step 4: We divide the *Sum Population* into distinct groups based on the score values that we assigned to each of the chromosomes previously. This process involves the creation of a group for each unique score value generated in Step 3, aligning each group with its corresponding score value. We then check the score values of all chromosomes within the *Sum Population* and allocate them to their respective groups accordingly. This grouping mechanism ensures that chromosomes sharing the same score value are consolidated into the corresponding group.
- Step 5: Chromosomes are added to the *Next Population* starting from the group with the highest score value until the desired number of chromosomes is selected for the *Next Population*.

Algorithm 5 provides more details on the proposed sorting approach.

4.2.4.2. Crowding distance. As mentioned in the previous subsection, the sorting approach divides the population into diffident groups based on their score values. Prioritizing the group with the highest score value, individuals are chosen and added to the *Next Population*. In this setting, initially, the chromosomes that belong to the group with the highest score value get chosen, then the ones in the group with the second highest score value are selected, and so on. This procedure continues until the favorable number of solutions for the *Next Population* is reached. However, a challenge is encountered if the number of solutions to be chosen to reach the perfect population size is less than the number of individuals in one of the groups from which we started selecting solutions. Since all the solutions in this subgroup have the same score value, none have priority over the others to get chosen. In this case, a crowding distance function is used to conquer the challenge. Noting that each chromosome represents a feasible solution to the scheduling problem, the crowding distance function ensures that diversity is maintained among the solutions (Xiong et al., 2012). In this way, a good variety of solutions can be provided to the decision-maker. In this paper, the crowding distance function proposed by Xiong et al. (2012) for FJSSPs is employed. They formulated the crowding distance function as presented in Eq. (18):

$$CrowdingDistance_Z = \frac{LevelSize - Assign_Z}{LevelSize} \tag{18}$$

where $LevelSize$ is the size of the group in which the challenge of prioritizing the solutions of the same score value was encountered. $Assign_Z$ represents the number of chromosomes in the aforementioned group with the same machine assignment as chromosome Z. In this setting, the chromosomes will be selected for the *Next Population* based on their $CrowdingDistanc$ value, from most to least. In this paper, we calculate the amount of $Assign_Z$ for each chromosome in the aforementioned group, and the ones with the least amount are prioritized. Furthermore, as represented in Fig. 3, the simulation model is called to obtain the required data about the assignment of the operations of each chromosome. In other words, the simulation model receives a chromosome and the Operation Environment as the input and develops a schedule accordingly. An *Operations' Assignment dictionary* is extracted from the developed schedule. This dictionary shows that each operation within the chromosome, representing a feasible solution, is assigned to which machine. This way, chromosomes with the same *Operations' Assignment dictionary* are identified, and each chromosome's $Assign_Z$ amount is calculated.

According to all those mentioned above and illustrated in Fig. 3, to form the *Next Population* of the proposed GA, an algorithm is applied using the sorting method and the crowding distance elaborated in previous subsections. Furthermore, the pseudo-code of the method proposed to form the *Next Population* is illustrated in algorithm 5.

4.2.5. Fitness function

The fitness function is a basis for measuring the quality of each chromosome (that represents a feasible solution for the scheduling problem) provided by the optimizer. In each generation, all the chromosomes are evaluated using the fitness function. As shown in Fig. 3, the fitness function is calculated via the simulation model and for three primary purposes: forming the *Selection Population*, forming the *Next Population*, and finding the *Best Solution* when GA's iterations end.

4.2.6. Termination condition

GA consists of iterative processes with the aim of improving the solutions in each iteration. In this regard, a termination condition defines when and how these iterations should end. The termination condition can vary from problem to problem. In the proposed GA, a time limit from the beginning of GA initialization is considered the termination condition.

All the components mentioned above are employed and integrated to form the proposed GA of this paper. An explanation of the way these components work together is provided in algorithm 6.

5. SO calibration

In this section, Taguchi's experimental design method is adopted via MINITAB 19 to calibrate the SO model. In this method, the arrangement of experiments is based on an Orthogonal Array (OA) that is useful for reducing experiment time and increasing convergence speed (Apornak et al., 2021). In the context of the proposed SO model, there are six key parameters: *Initial Population* size, *Selection Population* size, *Selection Threshold*, *MuteRate*, *CrossRate*, and SL. According to the explanation provided on the crossover and mutation operators in the previous section, the following can be stated:

$$CrossRate = 1 - MuteRate \quad (19)$$

Thus, since we can calculate the amount of *CrossRate* through the *MuteRate*, the number of the key parameters is reduced to five. As illustrated in Table 3, three levels are considered for each key parameter. Based on the L_{27} OA, 27 experiments were designed. The average signal-to-noise (S/N) ratio for objective function values is shown in Fig. 6. As indicated in this figure, the optimum combination of parameters for SO is as follows:

- *Initial Population* size = 100 ;
- *Selection Population* size = 200 ;
- *Selection Threshold* = 0.85
- *CrossRate* = 0.98 ;
- *MuteRate* = 0.02 ;
- *SL* = 40

After executing Taguchi and recognizing the desirable values for the key parameters, the proposed SO is employed to conduct numerical experiences in the following section. Noteworthy is to mention that, in all the experiments presented in Section 6, the proposed SO is employed with the above-mentioned values for the key parameters.

Algorithm 5: *Next Population* selection

```
Inputs : InPop, SelPop, CrossPop, MutePop, NextPopSize (number
    of chromosomes in the Next Population), SL, J, NO, N
Output: NextPop (Next Population)
SumPop = InPop + CrossPop + MutePop
NextPop = ()
FitList_all = ()
for Ch ∈ SumPop do
    FitDict_Ch = (FitList = (), Score = 0)
    for s = 1 to SL do
        FitValue_Ch = the fitness value for Ch calculated by the
          simulation model
        FitDict_Ch(1) = FitDict_Ch(1) + FitValue_Ch
        FitList_sorted = FitList_sorted + (Ch, FitValue_Ch) and sort in an
          ascending order based on the second item of each tuple in
          the FitList_sorted
    end
end
for i = 1 to len(FitList_sorted)/2 do
    for Ch ∈ SumPop do
        if FitList_sorted(i)(1) = Ch then
            FitDict_Ch(2) = FitDict_Ch(2) + 1
        end
    end
end
for g = 1 to the number of fitness values ∈ FitList_sorted do
    Group_g = ()
end
for Ch ∈ SumPop do
    Group_{FitDict_Ch(2)} = Group_{FitDict_Ch(2)} + Ch
end
g = the highest value from 1 to the number of fitness values
  ∈ FitList_sorted for which Group_g ≠ NAN
while the number of chromosomes ∈ NextPop ≠ NextPopSize do
    if the number of chromosomes
      ∈ (Group_g + NextPop) ≤ NextPopSize then
        NextPop = NextPop + Group_g
    else
        d = NextPopSize - the number of chromosomes∈ NextPop
        for Ch ∈ Group_g do
            AssignDict_Ch = a set of data provided by the simulation
              model that shows each operation in Ch is assigned to
              which of the available machines
            CrowdDistance_Ch = 0
        end
        for Ch ∈ Group_g do
            for Ch' ≠ Ch ∈ Group_g do
                if AssignDict_Ch == AssignDict_Ch' then
                    CrowdDistance_Ch = CrowdDistance_Ch + 1
                end
            end
            Sorted = a set of Ch ∈ Group_g sorted by their
              CrowdDistance_Ch in an ascending order.
            NextPop = NextPop + Sorted(1 : d)
        end
        break!
    end
    g = g-1
end
Return NextPop
```

6. Experiment results and sensitivity analysis

This section presents the numerical results of this paper. Firstly, comparative experiments are proposed to validate the applicability and superiority of our SO model in addressing various SJSSPs. In this regard, a set of comparisons are provided in the following subsections. Moreover, the second subsection is dedicated to sensitivity analysis to assess the effect of three special constraints of the photolithography

Algorithm 6: GA

Inputs : It
Output: $BestSolution_r$
$InPop$ = the initial population created to start the GA
for $r = 1$ **to** It **do**
  $SelPop$ = the *Selection Population* chosen from $InPop$ by the selection operator
  $CrossPop$ = the *Crossover Population* created by the crossover operator using $SelPop$
  $MutePop$ = the *Mutation Population* created by the mutation operator using $SelPop$
  $GenPop = CrossPop + MutePop$
  $SumPop = GenPop + InPop$
  for $Ch \in SumPop$ **do**
    $FitValue_{Ch}$ = the fitness value for Ch calculated by the simulation mode
  end
  $BestSolution_r = find(Ch|FitValue_{Ch} = min(FitValue_{Ch} \quad \forall Ch \in SumPop)$
  Return $BestSolution_r$
  $NextPop$ = the *Next Population* chosen from $SumPop$ by the sorting algorithm
  $InPop = NextPop$
  if *the termination condition is reached* **then**
    break!
  end
end

Table 3
Taguchi test parameters.

Parameter	Level		
	1	2	3
Initial Population size	100	200	300
Selection Population size	100	200	300
Selection Threshold	0.65	0.75	0.85
MuteRate	0.01	0.02	0.05
SL	30	40	50

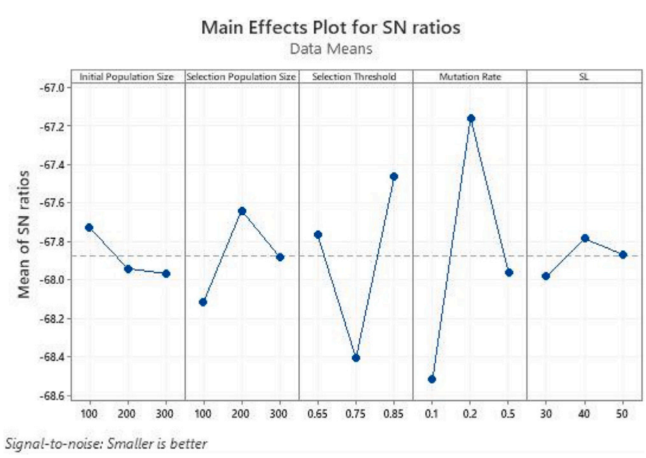

Fig. 6. Taguchi test results.

work area on the performance of the semiconductor manufacturing system.

6.1. Comparative experiments

This section presents the comparative experiments of this paper. These experiments are proposed to prove the excellence of SO in addressing SJSSPs compared to other algorithms in the literature. In this regard, this section is divided into two subsections. In the first part, the comparison is conducted adopting the test environment proposed by Horng et al. (2012) while the second part employs (Ghasemi et al., 2021)'s test environment to make the comparisons.

6.1.1. Horng et al. (2012)'s test environment

The test problem proposed by Horng et al. (2012) is illustrated in Table 4. There are eight jobs, with eight operations in each and eight machines in the proposed environment. Processing times of the operations are exposed to stochastic uncertainty and follow a probability distribution. In vectors (i, μ, σ^2), i refers to the sequence of the operation, μ the mean, and σ^2 the variance of the probability distribution to which the processing time of the operation belongs. Furthermore, the objective function for this instance is the sum of *tardiness* and *earliness*. Table 5 depicts the *due date* for each job. Moreover, the cost of both the *tardiness* and the *earliness* is equal to one. Horng et al. (2012) extracted three SJSSPs from the presented environment by considering various probability distributions for processing times. In the first scheduling problem, processing times follow a *Normal* distribution with μ and σ^2 as the mean and variance, respectively. In the second problem, *Uniform* distribution of $(\mu-3\sigma, \mu+3\sigma)$ is considered for processing times while in the third problem, processing times belong to an *Exponential* distribution with mean = μ. The following algorithms are nominated to be compared with the proposed SO in solving the elaborated SJSSPs:

- ESOO, an evolutionary SO method embedded in Ordinal Optimization (OO) presented by Horng et al. (2012).
- ESOO-OCBA, a modified version of ESOO using Optimal Computing Budget Allocation (OCBA) proposed by Yang et al. (2014).
- ELBSO, an evolutionary learning based SO developed by Ghasemi et al. (2021).

The results of solving the above-mentioned problems with ESOO, ESOO-OCBA, ELBSO, and SO are illustrated in Table 6. Due to the stochasticity of the test environment, each algorithm is replicated 20 times. The amount of objective function is calculated for 10^5 for the best solution in each replication and the mean of the obtained values is used to represent that replication (a value for each replication). Finally, the mean of these 20 values is calculated and depicted in Table 6 (for the case of ELBSO and SO, the standard deviation of the values is also reported in parenthesis). Moreover, the same time limitation (600 s) and the number of times that the optimization algorithm is iterated in each replication (100) in ESOO, ESOO-OCBA, and ELBSO are also considered for SO. In other words, in each replication, SO will terminate as soon as the optimization algorithm is iterated 100 times or as soon as 600 s are passed since the initiation of that replication, whichever occurs sooner. It is important to highlight that the 600 s time limitation was the one that ended the execution of the SO approach for these specific experiments. As can be seen, our SO model shows a promising performance in addressing the presented SJSSPs by resulting in considerably lower penalties for tardiness and earliness compared to the other similarly designed algorithms.

6.1.2. Ghasemi et al. (2021)'s test environment

In this section, the test environment proposed by Ghasemi et al. (2021) is employed to prove SO's superiority further. As illustrated in Table 7, the proposed environment consists of 15 jobs, with 15 operations belonging to each and 15 machines. Processing times of the operations are stochastic parameters and follow a *Normal* distribution. In vectors in the form of (i, μ, σ^2) in Table 7, i represents the operation id, while μ and σ^2 denote the mean and variance of the operation's processing time. The objective is to minimize the sum of *tardiness* and *earliness* for all jobs. In this regard, Table 8 depicts the due dates for each job. According to Ghasemi et al. (2021), eight diffident SJSSPs are extracted from the test environment presented in Table 7. Table 9 presents these problems which are varied in size and the cost of *tardiness* and *earliness*. For each problem, vector (N, NO_j, M) represents the number of jobs, operations in each job, and machines in the scheduling

Table 4
Horng et al. (2012)'s test environment.

Job id	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
1	(3, 70, 140)	(2, 80, 160)	(1, 90, 180)	(6, 50, 100)	(4, 40, 80)	(8, 60, 120)	(5, 70, 140)	(7, 50, 100)
2	(1, 80, 160)	(2, 40, 80)	(3, 50, 100)	(5, 90, 180)	(4, 40, 80)	(7, 50, 100)	(6, 60, 120)	(8, 40, 80)
3	(1, 50, 100)	(2, 40, 80)	(3, 80, 160)	(5, 60, 120)	(4, 70, 140)	(6, 40, 80)	(8, 40, 80)	(7, 70, 140)
4	(2, 60, 120)	(1, 50, 100)	(3, 60, 120)	(4, 70, 140)	(7, 80, 160)	(5, 40, 80)	(6, 50, 100)	(8, 80, 160)
5	(4, 50, 100)	(3, 50, 100)	(2, 70, 140)	(1, 40, 80)	(7, 50, 100)	(5, 60, 120)	(6, 90, 180)	(8, 60, 120)
6	(2, 60, 120)	(3, 80, 160)	(1, 90, 180)	(5, 70, 140)	(6, 50, 100)	(4, 40, 80)	(8, 80, 160)	(7, 90, 180)
7	(1, 40, 80)	(3, 60, 120)	(4, 40, 80)	(2, 80, 160)	(5, 60, 120)	(7, 70, 140)	(8, 50, 100)	(6, 60, 120)
8	(2, 90, 180)	(1, 70, 140)	(3, 50, 100)	(4, 60, 120)	(5, 90, 180)	(7, 80, 160)	(6, 40, 80)	(8, 40, 80)

Table 5
Job due dates in Horng et al. (2012)'s test environment.

Job id	1	2	3	4	5	6	7	8
Due Date	490	510	540	500	540	470	530	560

Table 6
Comparative experiments in an stochastic mode using Horng et al. (2012)'s test environment.

Probability distribution	Algorithm			
of the processing times	ESOO	ESOO-OCBA	ELBSO	SO
Normal	2280	2089	2269.56 (113.5)	2087.60 (187.14)
Uniform	2778	2452	2518.7 (119.12)	2396.43 (335.12)
Exponential	2683	2590	2550.71 (143.18)	2446.87 (256.02)

problem, respectively. Moreover, CT and CE account for the cost of *tardiness* and *earliness*. In Table 10, SO's results in solving the above-mentioned problems are compared against ESOO and ELBSO. The results are obtained in the same manner as explained in Section 6.1.1, with a time limitation of 600 s or reaching 100 iterations of the optimization algorithm as the termination condition. It is important to highlight that the first termination condition, i.e., the time limitation, was the one that ended the execution of the SO approach for these specific experiments. The results indicate that SO outperforms the other algorithms in solving the test problems and can be regarded as a potent tool to address SJSSPs of different settings.

6.2. Sensitivity analysis

This section focuses on the special constraints of the photolithography work area, namely, machine process capability constraints, machine dedication constraints, and maximum reticles (masks) sharing constraints. The following subsections, therefore, are presented to investigate the effect of the presence and severity of the aforementioned constraints on the performance of the system.

6.2.1. Presence of the special constraints of the photolithography work area

This section aims at comparing the performance of an SFJS manufacturing environment with and without the presence of the special constraints of the photolithography work area. In this regard, the operation environment proposed in Fig. 1 is employed to develop an SFJS environment without the presence of the photolithography work area, that is presented in Fig. 7, as well as the SFJS environment of the photolithography workstation, illustrated in Fig. 8. The processing times of the manufacturing environments presented in Figs. 7 and 8 are exposed to stochastic uncertainty and are represented in the form of (μ, σ^2) vectors (mean = μ and variance = σ^2). The stochastic sequence-dependent setup times are also presented in the form of (μ, σ^2) vectors (mean = μ and variance = σ^2) in Fig. 8. Since none of the special constraints of the photolithography work center are to be considered in the scheduling problem of the environment presented in Fig. 7, all operations are fully flexible (can be processed on all the present machines). In the context of the photolithography workstation, however, some operations can be processed by a subset of all the present machines enforcing the machine process capability constraints. Furthermore, no sequence-dependent setup times are considered for the operations environment of Fig. 7 since maximum reticles (masks) sharing constraints are not accounted for within this environment. Lastly, unlike the SFJS of Fig. 8, there is no reference to critical operations in the SFJS of Fig. 7 since the machine dedication constraints are not enforced in the latter. Three different probability distributions are used, similar to Section 6.1.1, to develop three scheduling problems based on each of the manufacturing environments. These problems, subsequently, are solved using the proposed SO model to study the effect of the presence of the photolithography special constraints on the performance of an SFJS environment. Using the makespan as the objective function of these SFJSSPs, the results are obtained in the same manner as explained in Section 6.1.1, with a time limitation of 600 s or reaching 100 iterations of the optimization algorithm as the termination condition (the time limitation was the criterion that ended the execution of the SO approach for these specific experiments). As presented in Table 11, forcing the photolithography special constraints results in lower performance levels for the manufacturing system. In other words, the lower values that are reported for the makespan of the scheduling problem in which the special constraints are absent compared to the SFJSSPs of the photolithography work area prove the difference caused by the presence of these constraints on the performance of the system.

6.2.2. Different levels of the special constraints of the photolithography work area

This section studies the effect of different severity of the machine process capability constraints, machine dedication constraints, and maximum reticles (masks) sharing constraints, on the scheduling problem of the photolithography work area. In this regard, three factors namely, *Flexibility Ratio*, *Machine Dedication Ratio*, and *Sequence-Dependent Setup Time Occurrence Ratio* are introduced in the following, using the notations presented in Table 2, each of which relates to one of the special constraints. A sensitivity analysis is, then, conducted to assess the performance of the photolithography work area under different levels of these factors.

- *Flexibility Ratio*: This factor indicates the number of flexible operations compared to the total number of operations. Flexible operations are the ones that can be processed on more than one

Table 7
Operations flow and their processing times in Ghasemi et al. (2021)'s test environment.

Job id	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
1	(3, 70, 140)	(2, 80, 160)	(1, 90, 180)	(5, 50, 100)	(4, 40, 80)	(8, 60, 120)	(6, 70, 140)	(7, 50, 100)
2	(1, 80, 160)	(2, 40, 80)	(3, 50, 100)	(5, 90, 180)	(4, 40, 80)	(7, 50, 100)	(6, 60, 120)	(8, 40, 80)
3	(1, 50, 100)	(2, 40, 80)	(3, 80, 160)	(5, 60, 120)	(4, 70, 140)	(10, 40, 80)	(8, 40, 80)	(7, 70, 140)
4	(2, 60, 120)	(1, 50, 100)	(3, 60, 120)	(4, 70, 140)	(5, 80, 160)	(7, 40, 80)	(10, 50, 100)	(8, 80, 160)
5	(4, 50, 100)	(3, 50, 100)	(2, 70, 140)	(1, 40, 80)	(5, 50, 100)	(7, 60, 120)	(9, 90, 180)	(10, 60, 120)
6	(2, 60, 120)	(3, 80, 160)	(1, 90, 180)	(5, 70, 140)	(4, 50, 100)	(6, 40, 80)	(9, 80, 160)	(10, 90, 180)
7	(1, 40, 80)	(3, 60, 120)	(4, 40, 80)	(2, 80, 160)	(5, 60, 120)	(7, 70, 140)	(8, 50, 100)	(6, 60, 120)
8	(2, 90, 180)	(1, 70, 140)	(3, 50, 100)	(4, 60, 120)	(5, 90, 180)	(7, 80, 160)	(6, 40, 80)	(10, 40, 80)
9	(5, 80, 160)	(4, 60, 120)	(3, 50, 100)	(2, 60, 120)	(1, 60, 120)	(7, 70, 140)	(10, 40, 80)	(8, 40, 80)
10	(2, 80, 160)	(1, 70, 140)	(3, 50, 100)	(4, 70, 140)	(5, 90, 180)	(8, 70, 140)	(6, 50, 100)	(7, 40, 80)
11	(5, 60, 120)	(1, 80, 160)	(3, 50, 100)	(4, 60, 120)	(2, 80, 160)	(10, 50, 100)	(6, 40, 80)	(8, 50, 100)
12	(3, 60, 120)	(1, 60, 120)	(2, 50, 100)	(5, 90, 180)	(4, 70, 140)	(10, 70, 140)	(9, 40, 80)	(8, 40, 80)
13	(1, 90, 180)	(2, 60, 120)	(3, 70, 140)	(4, 90, 180)	(5, 90, 180)	(6, 60, 120)	(7, 40, 80)	(8, 40, 80)
14	(2, 80, 180)	(1, 70, 140)	(3, 60, 120)	(4, 70, 140)	(5, 90, 180)	(7, 70, 140)	(8, 50, 100)	(6, 50, 100)
15	(2, 90, 180)	(1, 70, 140)	(3, 50, 100)	(4, 60, 120)	(5, 90, 180)	(7, 80, 160)	(6, 40, 80)	(8, 40, 80)

Job id	M_9	M_{10}	M_{11}	M_{12}	M_{13}	M_{14}	M_{15}
1	(10, 70, 140)	(9, 80, 160)	(11, 90, 180)	(13, 50, 100)	(12, 40, 80)	(15, 60, 120)	(14, 70, 140)
2	(10, 80, 160)	(9, 40, 80)	(15, 50, 100)	(12, 90, 180)	(11, 40, 80)	(13, 50, 100)	(14, 60, 120)
3	(6, 50, 100)	(9, 40, 80)	(13, 80, 160)	(12, 60, 120)	(11, 70, 140)	(15, 40, 80)	(14, 40, 80)
4	(9, 60, 120)	(6, 50, 100)	(15, 60, 120)	(11, 70, 140)	(13, 80, 160)	(12, 40, 80)	(14, 50, 100)
5	(6, 50, 100)	(8, 50, 100)	(11, 70, 140)	(12, 40, 80)	(15, 50, 100)	(14, 60, 120)	(13, 90, 180)
6	(7, 60, 120)	(8, 80, 160)	(15, 90, 180)	(14, 70, 140)	(13, 50, 100)	(12, 40, 80)	(11, 80, 160)
7	(9, 40, 80)	(10, 60, 120)	(11, 40, 80)	(12, 80, 160)	(14, 60, 120)	(13, 70, 140)	(15, 50, 100)
8	(8, 90, 180)	(9, 70, 140)	(12, 50, 100)	(13, 60, 120)	(15, 90, 180)	(11, 80, 160)	(14, 40, 80)
9	(9, 60, 180)	(6, 60, 120)	(14, 50, 100)	(15, 70, 140)	(11, 60, 120)	(12, 80, 160)	(13, 40, 80)
10	(9, 80, 160)	(10, 60, 120)	(11, 60, 120)	(12, 60, 120)	(13, 80, 160)	(14, 80, 160)	(15, 60, 120)
11	(7, 60, 120)	(9, 60, 120)	(14, 60, 120)	(15, 60, 120)	(13, 60, 120)	(12, 80, 160)	(11, 60, 120)
12	(7, 50, 100)	(6, 70, 140)	(13, 50, 100)	(14, 80, 160)	(15, 80, 160)	(11, 80, 160)	(12, 60, 120)
13	(9, 60, 120)	(10, 70, 140)	(11, 50, 100)	(12, 50, 100)	(13, 80, 160)	(14, 70, 140)	(15, 50, 100)
14	(10, 90, 180)	(9, 60, 120)	(15, 60, 120)	(14, 60, 120)	(13, 90, 180)	(11, 80, 160)	(12, 40, 80)
15	(9, 90, 180)	(10, 70, 140)	(11, 50, 100)	(12, 60, 120)	(13, 90, 180)	(14, 80, 160)	(15, 40, 80)

Table 8
Jobs due dates in Ghasemi et al. (2021)'s test environment.

Job id	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Due Date	290	310	440	300	440	470	530	560	640	610	550	700	760	740	820

Table 9
Constructed problems based on Ghasemi et al. (2021)'s test environment.

Problem id	1	2	3	4	5	6	7	8
Size	$(10 \cdot 10 \cdot 10)$	$(10 \cdot 10 \cdot 10)$	$(10 \cdot 10 \cdot 10)$	$(10 \cdot 10 \cdot 10)$	$(15 \cdot 15 \cdot 15)$	$(15 \cdot 15 \cdot 15)$	$(15 \cdot 15 \cdot 15)$	$(15 \cdot 15 \cdot 15)$
CT	1	1	5	5	1	1	5	5
CE	1	5	1	5	1	5	1	5

Table 10
Comparative experiments in an stochastic mode using Ghasemi et al. (2021)'s test environment.

Problem id	1	2	3	4	5	6	7	8
ELBSO Mean	9106.61	9119.52	44 258.7	45 606.82	35 761.3	35 900.3	180 486.46	181 767.82
ELBSO STD	219.41	218.49	1396.38	1589.31	1380.49	1604.22	5604.32	7655.91
ESOO Mean	9443.05	9473.31	47 005.7	47 840.51	37 858.27	37 952.31	184 344.22	185 448.52
ESOO STD	221.65	163.78	1165.9	1226.3	1087.81	2284.29	3667.85	7248.7
SO Mean	6840.68	6254.71	33 611.49	32 786.62	25 864.45	25 697.12	128 811.84	120 653.81
SO STD	320.06	307.48	1584.56	1566.17	697.70	700.36	3482.07	3413.05

machine. We used Eq. (20) to calculate this factor. This factor is related to the machine process capability constraint.

$$k^f = \frac{\sum_{o=1}^{NO_j} Fl_{oj} \quad j \in \{1, \dots, N\}}{NO} \tag{20}$$

- *Machine Dedication Ratio*: This factor indicates the number of critical operations compared to the total number of operations. We used Eq. (21) to calculate this factor. This factor is related to the machine dedication constraints.

$$k^c = \frac{\sum_{o=1}^{NO_j} Cr_{oj} \quad j \in \{1, \dots, N\}}{NO} \tag{21}$$

- *Sequence-Dependent Setup Time Occurrence Ratio*: This factor indicates the number of operations that processing one after the other needs sequence-dependent setup time compared to the total number of operations. We used Eq. (22) to calculate this factor. This factor is related to the maximum reticles (masks) sharing constraints.

$$k^s = \frac{\sum_{i=1}^{NO} \sum_{i'=1}^{NO} St_{ii'}}{NO} \tag{22}$$

Using the data provided from a real fab (Ghasemi et al., 2020) a test environment consisting of five jobs, with five operations in each and five machines is developed. For each operation, there is a set of

Operations Environment (Inputs)

Job ID	Operation ID	Stochastic processing times		
		Machine1	*Machine2*	*Machine3*
Job1	***Op1***	(7, 3)	(8, 2)	(9, 1)
	Op2	(8, 2)	(7, 3)	(5, 3)
	Op3	(5, 1)	(7, 3)	(8, 2)
Job2	***Op4***	(6, 1)	(5, 1)	(6, 1)
	Op5	(5, 1)	(5, 1)	(7, 3)
	Op6	(6, 1)	(8, 2)	(9, 1)
Job3	***Op7***	(8, 1)	(6, 2)	(9, 1)

Fig. 7. Proposed SFJS environment without special constraints of the photolithography work area.

Table 11
Comparative experiments on the presence of the three special constraints of the photolithography work area.

Probability distribution of the processing and Sequence-dependent Setup Times	Problem	
	SFJSSP-Photolithography	SFJSSP
Normal	28.61 (3.66)	22.76 (3.27)
Uniform	29.08 (4.17)	25.37 (5.63)
Exponential	34.14 (14.30)	30.07 (13.31)

alternative machines capable of processing it. As proposed by Ghasemi et al. (2020), the processing times follow a *Gamma* distribution with parameters 16.98 and 2.448. Furthermore, we considered stochastic sequence-dependent setup times between the operations of different jobs. According to the experts opinion the sequence-dependents setup times also follow a *Gamma* distribution with parameters set at 0.1 relative to the processing times. Eight different SFJSSPs are constructed based on the developed environment. The amounts of three key factors (*Flexibility Ratio*, *Machine Dedication Ratio*, and *Sequence-Dependent Setup Time Occurrence Ratio*) are what differentiate these eight scheduling problems from one another. These scheduling problems are solved using the proposed SO model with the objective function of minimizing makespan. Due to the stochasticity of the manufacturing environment presented in this section, each SO replication can report different results. Thus, 20 replications of SO are used, similar to Section 6.1.1, and the results are obtained in the same manner as explained in Section 6.1.1, terminating the SO in each replication after 600 s or after reaching 100 iterations of the optimization algorithm. It is worth noting that the 600 s time limitation ended the execution of the SO.

As Table 13 shows, different levels of k^f, k^c, and k^s affect the performance of the manufacturing system and result in various values for makespan. Fig. 9 presents a set of heat maps to analyze the effect of the factors on the performance level of the photolithography work area. Each row of the heat maps illustrates the relationship between two factors and the amount of makespan. According to the first row, increasing the level of flexibility (k^f) appears to prompt performance improvement by decreasing the amount of makespan (at both levels of k^c). In this regard, raising the level of flexibility gives more options to the SO model to find an optimal solution, which can positively affect the manufacturing system's performance. However, a reverse relationship between k^c and makespan seems to exist. In fact, both of the maps placed in the first row indicate that makespan worsens when k^c grows from 0.4 to 0.6 (in both levels of k^f). By increasing the number of critical operations of each job, the level of flexibility decreases. Thus, the SO model is limited in opting for the best solution with the minimum makespan. The second row illustrates the impact of k^f and k^s on the amount of makespan. Considering the amount of the makespan in each level of k^s, increasing the level of k^s has a negative effect on the performance of the manufacturing system. To be more detailed, when $k^f = 0.2$, increasing the level of k^s appears to increase the amount of makespan. A similar observation also holds when the

Table 12
Constructed problems based on the proposed test environment.

Problem id	1	2	3	4	5	6	7	8
k^f	0.2	0.2	0.2	0.2	0.6	0.6	0.6	0.6
k^c	0.4	0.4	0.6	0.6	0.4	0.4	0.6	0.6
k^s	0.1	0.2	0.1	0.2	0.1	0.2	0.1	0.2

amount of k^f is set to 0.6. The last row also demonstrates the same previously elaborated relationships (see Table 12).

To sum up, higher levels of k^f improve the performance of the manufacturing system while the opposite stands for k^s and k^c.

7. Conclusion

The ever-increasing development of technology and digital breakthroughs in the era of I4.0 has resulted in the advent of smart manufacturing systems. As one of the early adopters of I4.0, semiconductor manufacturing is an example of these smart manufacturing systems. On the one hand, semiconductor production is highly cost-intensive. On the other hand, the growth in using ICs in industrial, commercial, and military products renders the semiconductor industry of a rapidly increasing global demand. Thus, operational excellence has gained substantial importance in this industry. From the supply chain point of view, semiconductor manufacturing consists of front-end and back-end fabs. Most production processes are occurred within the front-end fab, while assemblies are mainly executed within the back-end fab. Within the front-end fab, the photolithography workstation is the well-known bottleneck process. Therefore, developing a competent schedule for the photolithography work area can elevate the performance level of the whole semiconductor production line. From the manufacturing system point of view, the scheduling problem of the photolithography workstation can be considered an SFJSSP. Due to the stochastic nature of the processing and sequence-dependent setup times and three special constraints (machine process capability constraints, machine dedication constraints, maximum reticles (masks) sharing constraints), the scheduling problem of this work area is among the most complex optimization problems. Furthermore, simpler versions of this problem are NP-hard (e.g., Low and Fang (2005)). Thus, conventional methods are inefficient in addressing this scheduling problem in a reasonable time. In this regard and in the era of I4.0, hybrid methods such as SO approaches have proven their applicability in addressing complex production scheduling problems. In other words, combining simulation models with optimization methods could develop a promising tool for solving various complex and stochastic industrial problems, such as scheduling the photolithography work area.

Thus, in this paper, we proposed an SO method, consisting of a simulation model and a tailored GA algorithm to solve the SFJSSP of the photolithography work area. In this regard, a mathematical model that respects all the specifications of the photolithography workstation (the uncertain nature and three special constraints) was presented. In the next step, we transformed the presented mathematical model into an SO model. Subsequently, a detailed simulation model of the photolithography area was proposed considering the stochasticity and special constraints of this work center. The simulation model develops an initial schedule based on LWR dispatching rule, satisfies the stochastic constraints of the scheduling problem, and calculates the objective function's value. Further on, a tailored GA was proposed to optimize the initially developed schedule iteratively integrated with the simulation model forming an SO structure. Then, we conducted a set of numerical experiments to validate the proposed SO and assess its performance compared to other algorithms in the literature. In this regard, first, SO was compared against three SO-based algorithms in the literature in addressing three SJSSPs with stochastic processing times for operations which varied in case of the probability distribution of the processing times. Considerable improvements can be achieved in cost

Operations Environment (Inputs)

Job ID (j)	Operation ID (i)	IsCritical (Cr_i)	Stochastic processing times ($\emptyset_{im}$)			Stochastic sequence-dependent setup times ($\emptyset'_{ii'}$)						
			Machine1	*Machine2*	*Machine3*	*Op1*	*Op2*	*Op3*	*Op4*	*Op5*	*Op6*	*Op7*
Job1	***Op1***	0	(7, 3)	(8, 2)	(9, 1)	-	-	-	(2, 0.01)	0	(1, 0.01)	0
	Op2	0	(8, 2)	(7, 3)	(5, 3)	0	-	-	(1, 0.01)	0	0	(3, 0.01)
	Op3	0	(5, 1)	(7, 3)	(8, 2)	0	0	-	(1, 0.01)	(2, 0.01)	0	(1, 0.01)
Job2	***Op4***	1	(6, 1)	(5, 1)	(6, 1)	(2, 0.01)	(1, 0.01)	(1, 0.01)	-	-	-	0
	Op5	0	(5, 1)	(5, 1)	(7, 3)	0	0	(2, 0.01)	0	-	-	(2, 0.01)
	Op6	1	(6, 1)	(8, 2)	(9, 1)	(1, 0.01)	0	0	0	0	-	(3, 0.01)
Job3	***Op7***	0	(8, 1)	(6, 2)	(9, 1)	0	(3, 0.01)	(1, 0.01)	0	(2, 0.01)	(3, 0.01)	-

Fig. 8. Proposed SFJS environment of the photolithography work area.

Table 13
Results of solving the problems.

Problem id	1	2	3	4	5	6	7	8
Mean	263.26	263.72	279.29	280.29	244.79	245.35	249.59	251.29
STD	17.31	17.31	18.82	18.88	18.48	18.40	18.29	18.29

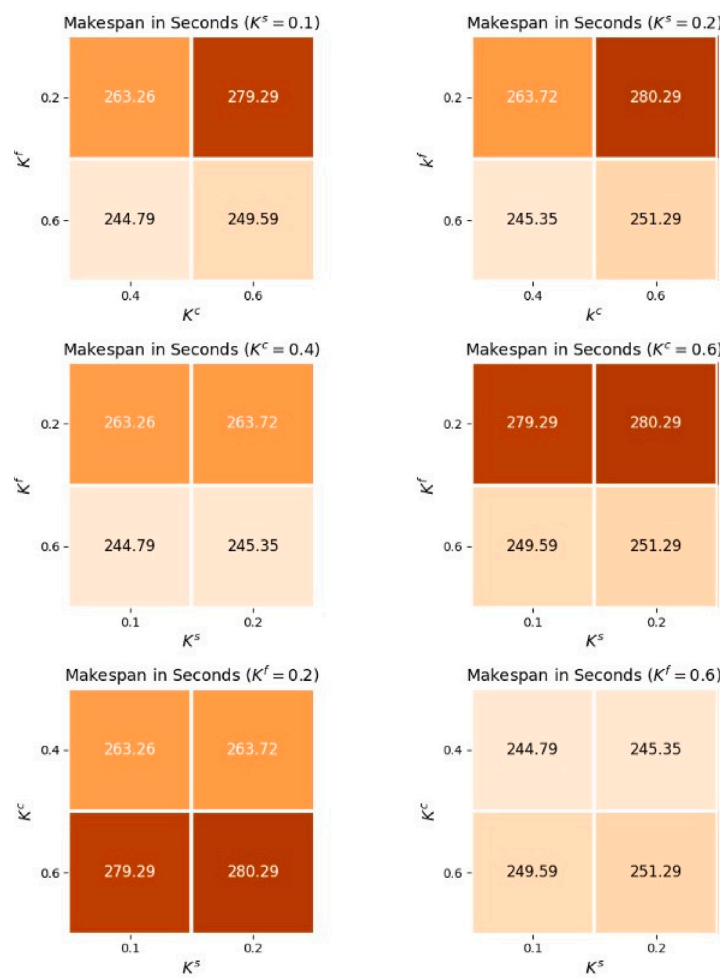

Fig. 9. Sensitivity analysis.

resulting from tardiness and earliness of the manufacturing system by using SO compared to other algorithms. Additionally, to further prove the superiority of SO in solving JSSPs under stochastic uncertainty of processing times, eight different SJSSPs were addressed employing the proposed SO in comparison with two similarly designed algorithms. The results of the comparative experiments proved that the proposed SO has the competency to solve a wide range of stochastic and complex scheduling problems in a reasonable time while resulting in higher performance levels for the system. Finally, using real fab data, we focused on the three special constraints of the photolithography work area. In this regard, first, we conducted a comparison to assess the effect of the presence of the special constraints on the performance level of the manufacturing system. Based on the results, enforcing the special constraints of the photolithography work area in an SFJS manufacturing environment affects the performance of the manufacturing system by resulting in higher values for the makespan compared to

Table 14
Results of solving the problems of Section 6.1.2.

Problem id	1	2	3	4	5	6	7	8
Approach one mean	7512.90	8679.30	40 360.14	40 331.74	30 305.62	31 370.19	138 970.36	142 768.70
Approach two mean	6840.68	6254.71	33 611.49	32 786.62	25 864.45	25 697.12	128 811.84	120 653.81

Table 15
Results of t-test for the problems of Section 6.1.2.

Sample	N	Median	P-value
Approach one mean - Approach two mean	8	4163.55	0.014

Table 16
Results of solving the problems of Section 6.2.2.

Problem id	1	2	3	4	5	6	7	8
Approach one mean	271.1	268.67	282.87	275.90	254.10	249.25	251.74	251.51
Approach two mean	263.26	263.72	279.29	280.29	244.79	245.35	249.59	251.29

Table 17
Results of t-test for the problems of Section 6.1.2.

Sample	N	Median	P-value
Approach one mean - Approach two mean	8	3.66	0.08

an SFJS environment that does not enforce these constraints. Subsequently, we introduced three factors namely, *Flexibility Ratio*, *Machine Dedication Ratio*, and *Sequence-Dependent Setup Time Occurrence Ratio* as representatives of machine process capability constraints, machine dedication constraints, and maximum reticles (masks) sharing constraints, respectively. The effect of different levels of the aforementioned factors on minimizing the makespan in the scheduling problem of the photolithography work area was analyzed. It was concluded that higher levels of flexibility can improve the performance of the manufacturing system while the opposite stands for the *Machine Dedication Ratio* and the *Sequence-Dependent Setup Time Occurrence Ratio*.

The scheduling problems encountered in manufacturing systems, especially those derived from real-world scenarios, are known to be highly challenging. Complex production systems, such as semiconductor manufacturing, require efficient scheduling methods that can effectively address their JSSPs within reasonable time constraints. Therefore, the proposed SO model holds significant promise as a practical and valuable decision-making tool, specifically within the context of semiconductor production lines. Furthermore, the versatility of the proposed SO model in handling diverse JSSPs, as demonstrated in Section 6, makes the reported methodology applicable to other manufacturing systems characterized by complex JSSPs. By relaxing the special constraints associated with the photolithography workstation and introducing similar constraints tailored to the specifications of the target manufacturing system, the proposed methodology can be readily extended to address scheduling problems in various manufacturing contexts.

This paper suggests three directions for future work. Firstly, there are myriad sources of uncertainty within a manufacturing environment among which stochastic processing and sequence-dependent setup times are addressed in this paper. However, other forms of uncertainty (such as machine breakdowns and emergency jobs) could also be taken into consideration in future works. Secondly, though simulation has proven its superiority in modeling, analyzing, and assessing real systems, it has an undeniable weak point. Simulation models of problems as complex as the SFJSSP of the photolithography work area can be highly time-intensive, causing SO implementations to be unacceptable for real large-sized industrial problems. Thus, using methods such as machine learning (ML) tools to replace the extensive simulation replications with a learning-based method can be another direction of future research. Last but not least, as mentioned earlier, FJSSPs belong to the NP-Hard class as complex combinatorial optimization problems (Geyik and Dosdoğru, 2013). Due to such complexity, several swarm intelligence and evolutionary algorithms, other than GA, have been developed within the literature to address these problems, including Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Tabu Search (TS), and Artificial Bee Colony (ABC) (Xiong et al., 2022). Though the proposed GA yields promising results in addressing the SFJSSP of the photolithography work area, employing other algorithms and comparing their results with those of our GA might unravel more efficient approaches to address such complex scheduling problems.

CRediT authorship contribution statement

Ensieh Ghaedy-Heidary: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Erfan Nejati:** Software, Investigation. **Amir Ghasemi:** Conceptualization, Writing – review & editing, Supervision, Project administration. **S. Ali Torabi:** Supervision, Project administration, Funding acquisition.

Data availability

Data will be made available on request.

Appendix

The literature on SO approaches in stochastic environments often employs the average of simulation replications for selecting chromosomes. To demonstrate the effectiveness of the selection approach proposed in this article, particularly in comparison to the average criterion, a comparison was conducted on the problems outlined in Sections 6.1.2 and 6.2.2, each characterized by distinct objective functions. The problems were solved twice — once using the average as the criterion (Approach One) for forming the *Selection Population* and the *Next Population*, and once using the approach introduced by our GA (Approach Two). The results for the problems outlined in Section 6.1.2 are presented in Table 14. Following this, a Wilcoxon Signed Rank Test, a non-parametric method designed for comparing two paired samples, was executed utilizing Minitab 19 to assess the efficacy of the two approaches. The test utilized the differences between the outcomes obtained from Approach One and Approach Two as the sample for analysis. As depicted in Table 15, the test yielded a P-Value of 0.014 and a Median of 4163.55. Indicating that, at a confidence level of 95%, the sorting approach of our GA outperforms the alternative approach. Similarly, Table 16 displays the results for solving the problems in Section 6.2.2 using both approaches. The Wilcoxon Signed Rank Test was used again and the same approach was used to generate the sample data for analysis (the differences between Approach One and Approach Two) resulting in a P-Value of 0.08 and a Median of 3.66, as shown in Table 17. Consistent with the previous experiment, our approach demonstrated better results compared to the alternative approach at a confidence level of 90%.

References

Akcali, E., Uzsoy, R., 2000. A sequential solution methodology for capacity allocation and lot scheduling problems for photolithography. In: Twenty Sixth IEEE/CPMT International Electronics Manufacturing Technology Symposium (Cat. No. 00CH37146). IEEE, pp. 374–381.

Akcalt, E., Nemoto, K., Uzsoy, R., 2001. Cycle-time improvements for photolithography process in semiconductor manufacturing. IEEE Trans. Semicond. Manuf. 14 (1), 48–56.

Apornak, A., Raissi, S., Pourhassan, M.R., 2021. Solving flexible flow-shop problem using a hybrid multi criteria taguchi based computer simulation model and DEA approach. J. Ind. Syst. Eng. 13 (2), 264–276.

Arakawa, M., Fuyuki, M., Inoue, I., 2003. An optimization-oriented method for simulation-based job shop scheduling incorporating capacity adjustment function. Int. J. Prod. Econ. 85 (3), 359–369.

Aydt, H., Turner, S.J., Cai, W., Low, M.Y.H., 2009. Research issues in symbiotic simulation. In: Proceedings of the 2009 Winter Simulation Conference (WSC). IEEE, pp. 1213–1222.

Bang, J.-Y., Kim, Y.-D., 2011. Scheduling algorithms for a semiconductor probing facility. Comput. Oper. Res. 38 (3), 666–673.

Bitar, A., Dauzère-Pérès, S., Yugma, C., 2014. On the importance of optimizing in scheduling: The photolithography workstation. In: Proceedings of the Winter Simulation Conference 2014. IEEE, pp. 2561–2570.

Bitar, A., Dauzère-Pérès, S., Yugma, C., Roussel, R., 2016. A memetic algorithm to solve an unrelated parallel machine scheduling problem with auxiliary resources in semiconductor manufacturing. J. Sched. 19 (4), 367–376.

Brindle, A., 1980. Genetic Algorithms for Function Optimization (Ph.D. thesis). University of Alberta.

Cakici, E., Mason, S., 2007. Parallel machine scheduling subject to auxiliary resource constraints. Prod. Plan. Control 18 (3), 217–225.

Cao, Z., Lin, C., Zhou, M., Huang, R., 2018. Scheduling semiconductor testing facility by using cuckoo search algorithm with reinforcement learning and surrogate modeling. IEEE Trans. Autom. Sci. Eng. 16 (2), 825–837.

Cemernek, D., Gursch, H., Kern, R., 2017. Big data as a promoter of industry 4.0: Lessons of the semiconductor industry. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). IEEE, pp. 239–244.

Chang, H.-C., Liu, T.-K., 2017. Optimisation of distributed manufacturing flexible job shop scheduling by using hybrid genetic algorithms. J. Intell. Manuf. 28 (8), 1973–1986.

Chen, T.-R., Chang, T.-S., Chen, C.-W., Kao, J., 1995. Scheduling for IC sort and test with preemptiveness via Lagrangian relaxation. IEEE Trans. Syst. Man Cybern. 25 (8), 1249–1256.

Chen, J.C., Chen, Y.-Y., Liang, Y., 2016. Application of a genetic algorithm in solving the capacity allocation problem with machine dedication in the photolithography area. J. Manuf. Syst. 41, 165–177.

Chen, J.C., Wu, C.-C., Chen, C.-W., Chen, K.-H., 2012. Flexible job shop scheduling with parallel machines using genetic algorithm and grouping genetic algorithm. Expert Syst. Appl. 39 (11), 10016–10021.

Chen, R., Yang, B., Li, S., Wang, S., 2020. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. Comput. Ind. Eng. 149, 106778.

Choi, I.-C., Choi, D.-S., 2002. A local search algorithm for jobshop scheduling problems with alternative operations and sequence-dependent setups. Comput. Ind. Eng. 42 (1), 43–58. http://dx.doi.org/10.1016/S0360-8352(02)00002-5, URL: https://linkinghub.elsevier.com/retrieve/pii/S0360835202000025.

Chung, S.-H., Huang, C.-Y., Lee, A.H., 2008. Heuristic algorithms to solve the capacity allocation problem in photolithography area (CAPPA). Or Spectrum 30 (3), 431–452.

Dabbas, R.M., Chen, H.-N., Fowler, J.W., Shunk, D., 2001. A combined dispatching criteria approach to scheduling semiconductor manufacturing systems. Comput. Ind. Eng. 39 (3–4), 307–324.

Defersha, F.M., Chen, M., 2009. A coarse-grain parallel genetic algorithm for flexible job-shop scheduling with lot streaming. In: 2009 International Conference on Computational Science and Engineering, Vol. 1. IEEE, pp. 201–208.

Djerid, L., Portmann, M.-C., 1996. Genetic algorithm operators restricted to precedent constraint sets: genetic algorithm designs with or without branch and bound approach for solving scheduling problems with disjunctive constraints. In: 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929), Vol. 4. IEEE, pp. 2922–2927.

Dosdoğru, A.T., Göçken, M., Geyik, F., 2015. Integration of genetic algorithm and Monte Carlo to analyze the effect of routing flexibility. Int. J. Adv. Manuf. Technol. 81, 1379–1389.

Driss, I., Mouss, K.N., Laggoun, A., 2015. A new genetic algorithm for flexible job-shop scheduling problems. J. Mech. Sci. Technol. 29 (3), 1273–1281.

Ellis, K., Lu, Y., Bish, E., 2004. Scheduling of wafer test processes in semiconductor manufacturing. Int. J. Prod. Res. 42 (2), 215–242.

Figueira, G., Almada-Lobo, B., 2014. Hybrid simulation–optimization methods: A taxonomy and discussion. Simul. Model. Pract. Theory 46, 118–134. http://dx.doi.org/10.1016/j.simpat.2014.03.007, URL: https://linkinghub.elsevier.com/retrieve/pii/S1569190X14000458.

Geyik, F., Cedimoglu, I.H., 2004. The strategies and parameters of tabu search for job-shop scheduling. J. Intell. Manuf. 15 (4), 439–448.

Geyik, F., Dosdoğru, A.T., 2013. Process plan and part routing optimization in a dynamic flexible job shop scheduling environment: an optimization via simulation approach. Neural Comput. Appl. 23 (6), 1631–1641. http://dx.doi.org/10.1007/s00521-012-1119-7, URL: http://link.springer.com/10.1007/s00521-012-1119-7.

Ghasemi, A., Ashoori, A., Heavey, C., 2021. Evolutionary learning based simulation optimization for stochastic job shop scheduling problems. Appl. Soft Comput. 106, 107309. http://dx.doi.org/10.1016/j.asoc.2021.107309, URL: https://linkinghub.elsevier.com/retrieve/pii/S1568494621002325.

Ghasemi, A., Azzouz, R., Laipple, G., Kabak, K.E., Heavey, C., 2020. Optimizing capacity allocation in semiconductor manufacturing photolithography area – case study: Robert bosch. J. Manuf. Syst. 54, 123–137. http://dx.doi.org/10.1016/j.jmsy.2019.11.012, URL: https://linkinghub.elsevier.com/retrieve/pii/S0278612519301153.

Ghasemi, A., Heavey, C., Kabak, K.E., 2018. Implementing a new genetic algorithm to solve the capacity allocation problem in the photolithography area. In: Proceedings of the 2018 Winter Simulation Conference. WSC '18, IEEE Press, pp. 3696–3707, URL: http://dl.acm.org/citation.cfm?id=3320516.3320956. event-place: Gothenburg, Sweden.

Gong, H., Chen, D., Xu, K., 2014. Parallel-batch scheduling and transportation coordination with waiting time constraint. Sci. World J. 2014.

Gu, J., Gu, M., Cao, C., Gu, X., 2010. A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. Comput. Oper. Res. 37 (5), 927–937.

Gupta, A., Sivakumar, A., 2002. Simulation based multiobjective schedule optimization in semiconductor manufacturing. In: Proceedings of the Winter Simulation Conference, Vol. 2. IEEE, San Diego, CA, USA, pp. 1862–1870. http://dx.doi.org/10.1109/WSC.2002.1166480, URL: http://ieeexplore.ieee.org/document/1166480/.

Gupta, A.K., Sivakumar, A.I., 2006. Job shop scheduling techniques in semiconductor manufacturing. Int. J. Adv. Manuf. Technol. 27 (11), 1163–1169.

Ham, M., 2012. Integer programming-based real-time dispatching (i-RTD) heuristic for wet-etch station at wafer fabrication. Int. J. Prod. Res. 50 (10), 2809–2822.

Ham, A.M., Cho, M., 2015. A practical two-phase approach to scheduling of photolithography production. IEEE Trans. Semicond. Manuf. 28 (3), 367–373.

Herding, R., Mönch, L., 2022. An agent-based infrastructure for assessing the performance of planning approaches for semiconductor supply chains. Expert Syst. Appl. 202, 117001. http://dx.doi.org/10.1016/j.eswa.2022.117001, URL: https://www.sciencedirect.com/science/article/pii/S0957417422004201.

Horng, S.-C., Lin, S.-S., Yang, F.-Y., 2012. Evolutionary algorithm for stochastic job shop scheduling with random processing time. Expert Syst. Appl. 39 (3), 3603–3610. http://dx.doi.org/10.1016/j.eswa.2011.09.050, URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417411013595.

Jamrus, T., Chien, C.-F., Gen, M., Sethanan, K., 2018. Hybrid particle swarm optimization combined with genetic operators for flexible job-shop scheduling under uncertain processing time for semiconductor manufacturing. IEEE Trans. Semicond. Manuf. 31 (1), 32–41. http://dx.doi.org/10.1109/TSM.2017.2758380, URL: http://ieeexplore.ieee.org/document/8054748/.

Jiang, T., Zhang, C., 2018. Application of grey wolf optimization for solving combinatorial problems: job shop and flexible job shop scheduling cases. IEEE Access 6, 26231–26240.

Johnzén, C., Vialletelle, P., Dauzere-Péres, S., Yugma, C., Derreumaux, A., 2008. Impact of qualification management on scheduling in semiconductor manufacturing. In: 2008 Winter Simulation Conference. IEEE, pp. 2059–2066.

Kawanaka, H., Yamamoto, K., Yoshikawa, T., Shinogi, T., Tsuruoka, S., 2001. Genetic algorithm with the constraints for nurse scheduling problem. In: Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No. 01TH8546), Vol. 2. IEEE, pp. 1123–1130.

Kim, S., Yea, S.-H., Kim, B., 2002. Shift scheduling for steppers in the semiconductor wafer fabrication process. IIE Trans. 34 (2), 167–177.

Klemmt, A., Lange, J., Weigert, G., Lehmann, F., Seyfert, J., 2010. A multistage mathematical programming based scheduling approach for the photolithography area in semiconductor manufacturing. In: Proceedings of the 2010 Winter Simulation Conference. IEEE, pp. 2474–2485.

Krug, W., Wiedemann, T., Liebelt, J., Baumbach, B., Verbraeck, A., 2002. Simulation and optimization in manufacturing, organization and logistics. In: Proceedings 14th European Simulation Symposium. p. 7.

Kuck, M., Ehm, J., Hildebrandt, T., Freitag, M., Frazzon, E.M., 2016. Potential of data-driven simulation-based optimization for adaptive scheduling and control of dynamic manufacturing systems. In: 2016 Winter Simulation Conference (WSC). IEEE, Washington, DC, USA, pp. 2820–2831. http://dx.doi.org/10.1109/WSC.2016.7822318, URL: http://ieeexplore.ieee.org/document/7822318/.

Lee, Y.F., Jiang, Z.B., Liu, H.R., 2009. Multiple-objective scheduling and real-time dispatching for the semiconductor manufacturing system. Comput. Oper. Res. 36 (3), 866–884. http://dx.doi.org/10.1016/j.cor.2007.11.006, URL: https://www.sciencedirect.com/science/article/pii/S0305054807002316.

Lee, Y.H., Lee, B., 2003. Push-pull production planning of the re-entrant process. Int. J. Adv. Manuf. Technol. 22 (11), 922–931.

Lee, Y.H., Lee, S., 2022. Deep reinforcement learning based scheduling within production plan in semiconductor fabrication. Expert Syst. Appl. 191, 116222. http://dx.doi.org/10.1016/j.eswa.2021.116222, URL: https://www.sciencedirect.com/science/article/pii/S0957417421015359.

Lee, Y.H., Park, J., Kim, S., 2002. Experimental study on input and bottleneck scheduling for a semiconductor fabrication line. IIE Trans. 34 (2), 179–190.

Lee, K.-M., Yamakawa, T., Lee, K.-M., 1998. A genetic algorithm for general machine scheduling problems. In: 1998 Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No. 98EX111), Vol. 2. IEEE, pp. 60–66.

Li, S., Tang, T., Collins, D.W., 1996. Minimum inventory variability schedule with applications in semiconductor fabrication. IEEE Trans. Semicond. Manuf. 9 (1), 145–149.

Linnéusson, G., Ng, A.H., Aslam, T., 2020. A hybrid simulation-based optimization framework supporting strategic maintenance development to improve production performance. European J. Oper. Res. 281 (2), 402–414.

Liu, Z., Wang, J., Zhang, C., Chu, H., Ding, G., Zhang, L., 2021. A hybrid genetic-particle swarm algorithm based on multilevel neighbourhood structure for flexible job shop scheduling problem. Comput. Oper. Res. 135, 105431.

Longo, F., 2010. Emergency simulation: state of the art and future research guidelines. SCS M&S Mag. 1 (4), 1–8.

Low, C.P., Fang, C., 2005. On the load-balanced demand points assignment problem in large-scale wireless LANs. In: International Conference on Information Networking. Springer, pp. 21–30.

Lu, S.C., Ramaswamy, D., Kumar, P., 1994. Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants. IEEE Trans. Semicond. Manuf. 7 (3), 374–388.

Malekpour, H., Hafezalkotob, A., Khalili-Damghani, K., 2021. Product processing prioritization in hybrid flow shop systems supported on Nash bargaining model and simulation-optimization. Expert Syst. Appl. 180, 115066. http://dx.doi.org/10.1016/j.eswa.2021.115066, URL: https://www.sciencedirect.com/science/article/pii/S0957417421005078.

März, L., Krug, W., Marz, L., Rose, O., Weigert, G., 2011. Simulation und Optimierung in Produktion und Logistik. Springer.

Mokhtari, H., Dadgar, M., 2015. Scheduling optimization of a stochastic flexible job-shop system with time-varying machine failure rate. Comput. Oper. Res. 61, 31–45.

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

Mönch, L., Uzsoy, R., Fowler, J.W., 2018a. A survey of semiconductor supply chain models part I: semiconductor supply chains, strategic network design, and supply chain simulation. Int. J. Prod. Res. 56 (13), 4524–4545. http://dx.doi.org/10.1080/00207543.2017.1401233, URL: https://www.tandfonline.com/doi/full/10.1080/00207543.2017.1401233.

Mönch, L., Uzsoy, R., Fowler, J.W., 2018b. A survey of semiconductor supply chain models part III: master planning, production planning, and demand fulfilment. Int. J. Prod. Res. 56 (13), 4565–4584.

Nishi, Y., Doering, R., 2000. Handbook of Semiconductor Manufacturing Technology. CRC Press.

Ólafsson, S., 2006. Metaheuristics. Handb. Oper. Res. Manag. Sci. 13, 633–654.

Park, H., Stefanski, L.A., 1998. Relative-error prediction. Statist. Probab. Lett. 40 (3), 227–236. http://dx.doi.org/10.1016/S0167-7152(98)00088-1, URL: http://www.sciencedirect.com/science/article/pii/S0167715298000881.

Peng, B., Lü, Z., Cheng, T.C.E., 2015. A tabu search/path relinking algorithm to solve the job shop scheduling problem. Comput. Oper. Res. 53, 154–164.

Pezzella, F., Morganti, G., Ciaschetti, G., 2008. A genetic algorithm for the flexible job-shop scheduling problem. Comput. Oper. Res. 35 (10), 3202–3212. http://dx.doi.org/10.1016/j.cor.2007.02.014, URL: https://linkinghub.elsevier.com/retrieve/pii/S0305054807000524.

Pfund, M.E., Mason, S.J., Fowler, J.W., 2006. Semiconductor manufacturing scheduling and dispatching. In: Handbook of Production Scheduling. In: International Series in Operations Research & Management Science, Springer, Boston, MA, pp. 213–241. http://dx.doi.org/10.1007/0-387-33117-4_9, URL: https://link.springer.com/chapter/10.1007/0-387-33117-4_9.

Shao, G., Shangguan, Y., Tao, J., Zheng, J., Liu, T., Wen, Y., 2018. An improved genetic algorithm for structural optimization of Au–Ag bimetallic nanoparticles. Appl. Soft Comput. 73, 39–49.

Sivakumar, A.I., 2001. Multiobjective dynamic scheduling using discrete event simulation. Int. J. Comput. Integr. Manuf. 14 (2), 154–167. http://dx.doi.org/10.1080/09511920150216279, URL: http://www.tandfonline.com/doi/abs/10.1080/09511920150216279.

Spier, J., Kempf, K., 1995. Simulation of emergent behavior in manufacturing systems. In: Proceedings of SEMI Advanced Semiconductor Manufacturing Conference and Workshop. IEEE, pp. 90–94.

Tay, J.C., Wibowo, D., 2004. An effective chromosome representation for evolving flexible job shop schedules. In: Genetic and Evolutionary Computation Conference. Springer, pp. 210–221.

Toktay, L.B., Uzsoy, R., 1998. A capacity allocation problem with integer side constraints. European J. Oper. Res. 109 (1), 170–182.

Uzsoy, R., Lee, C.-Y., Martin-Vega, L.A., 1992. A review of production planning and scheduling models in the semiconductor industry part I: system characteristics, performance evaluation and production planning. IIE Trans. 24 (4), 47–60.

Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., Kyek, A., 2018. Optimization of global production scheduling with deep reinforcement learning. Procedia CIRP 72, 1264–1269. http://dx.doi.org/10.1016/j.procir.2018.03.212, URL: https://linkinghub.elsevier.com/retrieve/pii/S221282711830372X.

Wein, L.M., 1988. Scheduling semiconductor wafer fabrication. IEEE Trans. Semicond. Manuf. 1 (3), 115–130.

Wu, J.-Z., Chien, C.-F., 2008. Modeling semiconductor testing job scheduling and dynamic testing machine configuration. Expert Syst. Appl. 35 (1–2), 485–496.

Xiong, H., Shi, S., Ren, D., Hu, J., 2022. A survey of job shop scheduling problem: The types and models. Comput. Oper. Res. 142, 105731. http://dx.doi.org/10.1016/j.cor.2022.105731, URL: https://www.sciencedirect.com/science/article/pii/S0305054822000338.

Xiong, J., Tan, X., Yang, K.-w., Xing, L.-n., Chen, Y.-w., 2012. A hybrid multiobjective evolutionary approach for flexible job-shop scheduling problems. Math. Probl. Eng. 2012, 1–27. http://dx.doi.org/10.1155/2012/478981, URL: https://www.hindawi.com/journals/mpe/2012/478981/.

Xiong, H.H., Zhou, M., 1998. Scheduling of semiconductor test facility via Petri nets and hybrid heuristic search. IEEE Trans. Semicond. Manuf. 11 (3), 384–393.

Yan, B., Chen, H.Y., Luh, P.B., Wang, S., Chang, J., 2012. Optimization-based litho machine scheduling with load balancing and reticle expiration. In: 2012 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, pp. 575–580.

Yang, H., Lv, Y., Xia, C., Sun, S., Wang, H., 2014. Optimal computing budget allocation for ordinal optimization in solving stochastic job shop scheduling problems. Math. Probl. Eng. 2014, 1–10. http://dx.doi.org/10.1155/2014/619254, URL: http://www.hindawi.com/journals/mpe/2014/619254/.

Yugma, C., Riffart, R., Dauzere-Peres, S., Vialletelle, P., Buttin, F., 2007. A dispatcher simulator for a photolithography workshop. In: 2007 IEEE/SEMI Advanced Semiconductor Manufacturing Conference. IEEE, pp. 100–104.

Zhang, G., Gao, L., Shi, Y., 2011. An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Syst. Appl. 38 (4), 3563–3573.

Zhang, H., Jiang, Z., Guo, C., 2009. Simulation-based optimization of dispatching rules for semiconductor wafer fabrication system scheduling by the response surface methodology. Int. J. Adv. Manuf. Technol. 41 (1–2), 110–121. http://dx.doi.org/10.1007/s00170-008-1462-0, URL: http://link.springer.com/10.1007/s00170-008-1462-0.

Zhang, P., Lv, Y., Zhang, J., 2018. An improved imperialist competitive algorithm based photolithography machines scheduling. Int. J. Prod. Res. 56 (3), 1017–1029.

Zhang, Q., Manier, H., Manier, M.-A., 2012. A genetic algorithm with tabu search procedure for flexible job shop scheduling with transportation constraints and bounded processing times. Comput. Oper. Res. 39 (7), 1713–1723.