Proceedings of the 2022 Winter Simulation Conference
B. Feng, G. Pedrielli, Y. Peng, S. Shashaani, E. Song, C.G. Corlu, L.H. Lee, E.P. Chew, T. Roeder, and P. Lendermann, eds.

A BIASED-RANDOMIZED SIMHEURISTIC FOR A HYBRID FLOW SHOP WITH STOCHASTIC PROCESSING TIMES IN THE SEMICONDUCTOR INDUSTRY

Majsa Ammouriova
Javier Panadero

Computer Science Department
Universitat Oberta de Catalunya
Euncet Business School
Barcelona, 08018, SPAIN

Madlene Leißau
Christoph Laroque

Institute for Management and Information
University of Applied Sciences Zwickau
Kornmarkt 1
Zwickau, 08056, GERMANY

Christin Schumacher

Informatik 4 – Modeling and Simulation
Department of Computer Science
TU Dortmund University
Dortmund, D-44221, GERMANY

Angel A. Juan

Dept. of Applied Statistics and Operations Research
Universitat Politècnica de València
Plaza Ferrandiz-Carbonell, s/n
Alcoy, 03801, SPAIN

ABSTRACT

Compared to other industries, production systems in semiconductor manufacturing have an above-average level of complexity. Developments in recent decades document increasing product diversity, smaller batch sizes, and a rapidly changing product range. At the same time, the interconnections between equipment groups increase due to rising automation, thus making production planning and control more difficult. This paper discusses a hybrid flow shop problem with realistic constraints, such as stochastic processing times and priority constraints. The primary goal of this paper is to find a solution set (permutation of jobs) that minimizes the production makespan. The proposed algorithm extends our previous work by combining biased-randomization techniques with a discrete-event simulation heuristic. This simulation-optimization approach allows us to efficiently model dependencies caused by batching and by the existence of different flow paths. As shown in a series of numerical experiments, our methodology can achieve promising results even when stochastic processing times are considered.

1 INTRODUCTION

Manufacturing companies are experiencing many challenges regarding customer orientation and on-time production, especially in an intensified global business and a digital transformation. In the past, generally, the primary objective was to increase the utilization of production chains towards an economic optimum. However, this goal has increasingly shifted more towards customer-oriented production in recent years. Accordingly, the focus of most companies is now on-time feasibility and adherence to promised delivery dates. This confronts operational production planning with questions about certain batches' latest possible release, so that the product can still be manufactured and delivered on time. A constantly increasing complexity of production systems, in conjunction with a high degree of automation and random events –such as equipment failures and dependencies of immediately upstream or downstream equipment groups– repeatedly pose challenges for many production companies. As a result, there is an increasing need

978-1-6654-7661-4/22/$31.00 ©2022 IEEE

to use optimization algorithms to handle the complexity of planning problems. According to Amaran et al. (2016), the use of simulation-optimization approaches could provide a differentiated answer here. Simulation-optimization methods are well established in planning, design, and ramp-up but are currently rarely used for operational decision support.

In many production systems, jobs can be processed through different equipment and transport routes. For example, in semiconductor manufacturing, individual jobs take different routes through the production system depending on various specifications and parameters set in advance. Different paths through the production system might cause time dependencies, so the order in which jobs leave the system might differ from the order in which jobs enter it. Consequently, this can lead to difficulties in planning batch-related insertion dates, especially if batch processes are also found within the production system. Therefore, the use of discrete-event simulation is needed to model the system and these time dependencies. In this paper, we analyze a stochastic real-world scheduling use case in the form of a multi-path version of a hybrid flow shop scheduling problem (Tosun et al. 2020). Flow shops describe production scenarios, in which every job follows the same direction through the production system. A flow shop is hybrid, if there exists more than one machine per stage. Assembly flow shops describe independent parallel flow shops, which all end up in one or more same stages. This version considers two batch processes, a global priority rule and a so-called "same setup" rule. The model is based on the specifics of a real production process managed by a German manufacturing industrial partner. The aforementioned elements make the computation of the makespan a non-trivial task: due to the existence of time dependencies and batches, the makespan cannot always be computed using a closed analytical expression. Furthermore, stochastic processing times pose an additional challenge, which has now been considered in this real-world scheduling use case.

Accordingly, the main contributions of this paper are: *(i)* a simulation model of a hybrid flow shop problem with pre-determined paths, which are predefined by individual parameters, specifications, and batching; *(ii)* a fast discrete-event heuristic that is able to deal with the complexity of the modeled system and compute the makespan associated with a proposed solution; *(iii)* the extension of the previous heuristic to a biased-randomized algorithm (Juan et al. 2020), which introduces an extended degree of randomness into the heuristic constructive process; and *(iv)* the evaluation of computational experiments, which show the performance of the proposed methodology for solving this scheduling problem. Concepts of discrete-event simulation and heuristic algorithms can be combined into discrete-event driven heuristics. Some typical applications include scenarios in which synchronization issues are relevant (Fikar et al. 2016), or rare events have to be modeled (Gholami et al. 2009; Allaoui and Artiba 2004). Furthermore, biased-randomized algorithms may generate alternative solutions based on heuristic dispatching rules. By employing Monte Carlo simulation and skewed probability distributions, biased-randomized techniques introduce a non-uniform random behavior into a dispatching rule. Employing parallelization techniques, these heuristics can be applied in the same computational time as the original dispatching rule, thus making these algorithms much more powerful than the original heuristics they build upon. Applications of these algorithms can be found in the scientific literature for flow shop problems (Ferrer et al. 2016). Similarly, applications of simheuristics in flow shop and other logistics problems are also available (Hatami et al. 2018; Gruler et al. 2017; Onggo et al. 2019). Despite that, a flow shop problem in the context of semiconductor manufacturing, as the one described here, has never been solved in related literature.

The paper is organized as follows. Section 2 provides a more detailed description of the scheduling problem studied in this paper. Section 3 reviews related work on similar flow shop problems. Section 4 provides more details on the proposed algorithms, so that other researchers or practitioners can reproduce them. Section 5 carries out a series of numerical experiments to test our methodology, and analyzes the obtained results. Finally, Section 6 summarizes the main findings of this paper and points out some future research lines.

2 DESCRIBING THE HYBRID FLOW SHOP PROBLEM

Our problem is based on the characteristics and restrictions of a pre-assembly production process in the semiconductor manufacturing industry. Figure 1 shows the considered problem of this paper. This problem can be categorized with the scheme first introduced by Graham et al. (1979) as a hybrid flow shop with several realistic constraints. In this scheme, scheduling problems are described with parameter tuples $(\alpha|\beta|\gamma)$. Since production systems complexity is increasing, Pinedo (2012) among others extends the classical notation with additional production environments (in α), constraints (in β), and objective functions (in γ).

The production scenario (Figure 1) contains 10 production processes, where one machine processes one job at a time, and 2 production processes that gather products into batches. To support a uniform notation in scheduling literature, the variables are chosen wherever possible according to Ruiz and Vázquez-Rodríguez (2010) and Ruiz et al. (2008). Machines $i \in M$ process jobs $j \in N$ of different types or families $t_j \in T$ (*fmls*). For our production process (Figure 1) M is defined as $M = \{P1, P2, \ldots, P10, G1, G2\}$. Four possible product types t_j appearing in the production scenario are $T = \{T1, T2, S1, S2\}$.

According to their type, jobs have to be produced by a sequence of machines on a predefined route. Every job j is processed on machine $P1$ as a first step. After that, production paths are divided according to the job's type. Jobs $j \in \{S1, S2\}$ follow the upper path of Figure 1 and are processed next by machine $P2$. Jobs of product types $T1$ and $T2$ ($j \in \{T1, T2\}$) are processed at the lower path of Figure 1 on machine $P3$, $P6$, $P8$, and, subsequently, by machine $P9$ in the case of product type $T2$. Jobs of job type $T1$ skip machine $P9$ in the lower path. Accordingly, the model contains skipping stages (*skip*) referred to the categorization scheme of Pinedo (2012).

In the upper path, jobs are handled after $P2$ in the case of type $S1$ by machine $P4$, and in the case of type $S2$ by $P5$ and $P7$. So, the model includes machine qualifications (M_j) according to the categorization scheme of Pinedo (2012). In addition, a priority rule regarding the jobs is applied in the system. Note that, because of their different paths, it is possible that jobs are processed at different speeds through the production system. Jobs of product type $T2$ are prioritized, so if there are jobs of type $T2$ waiting in a queue in front of a machine, they are processed first on the machine (*prec*).

Apart from the batching processes, a 'same setup' priority rule is also applied in the production. In the semiconductor industry, numerous different product types are handled by the systems, in which each requires a special setup on the machines. In order to reduce these setup times, jobs of the same product type are given priority if possible (same setup). This method helps to sort the jobs in the queue depending on the setup status of the machines to minimize setup times and their influence to the makespan.

At the end of the production process in Figure 1, machines $i \in \{G1, G2\}$ gather several jobs and process them simultaneously. Hence, they are referred to as batching machines, where the machines can deal with several product types in one batch. Six jobs are processed simultaneously on machine $G1$, and the processing of jobs begins as soon as six jobs of product types $S1$ and $S2$ are waiting in front of the machine. As a last step, all jobs have to pass machine $P10$. The objective for the problem of our paper is then to find an input permutation of jobs (a solution) that reduces the makespan C_{max}. The deterministic problem can be formalized as:

$$FHm \mid M_j,\ batch,\ prec,\ skip,\ fmls \mid C_{max}.$$

Additionally, we consider uncertainty in process parameters to reflect real-world use case. To the best of our knowledge, such a realistic formulation of a hybrid flow shop from practice has not been analyzed in literature. Also hybrid flow shop scenarios with five β-components are rarely found in literature. The consideration of uncertainty for scheduling problems is often neglected. Even though solutions to our problem are missing in literature, related problems and related literature is presented in the following section.

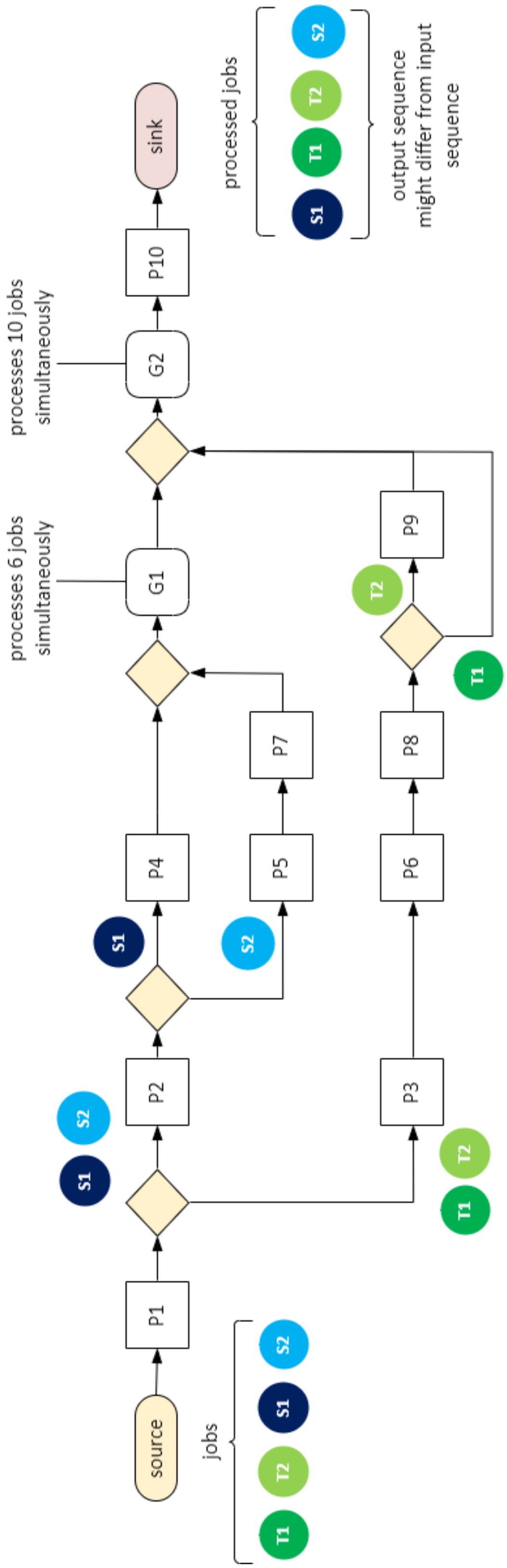

Figure 1: A hybrid flow shop problem with random processing times, flow paths, and batching.

3 RELATED WORK

In several research papers, mixed integer programs, heuristics, metaheuristics and discrete-event simulation experiments are developed for hybrid flow shop problems for different application cases (Allaoui and Artiba 2004; Gholami et al. 2009; Fan et al. 2018; Lee and Loong 2019). The hybrid flow shop problem is NP-hard, even for a scenario with two stages and two identical parallel machines on one of both stages, and one machine on the other stage (Gupta 1988).

Several literature reviews of hybrid flow shops and assembly flow shops can be found (Komaki et al. 2019; Ruiz and Vázquez-Rodríguez 2010; Nikzad et al. 2015). A review of hybrid flow shops with the integration of batching components is provided in Morais et al. (2013). The optimization criteria makespan and time-based objectives dominate in the literature for flow shop environments (Komaki et al. 2019; Lee and Loong 2019). For hybrid flow shops, two possible formulations of priority rules can be found. First, there is a fixed defined ranking of priority groups, such as $Prio_1 = \{j_4\}$, $Prio_2 = \{j_1, j_3\}$, and $Prio_3 = \{j_2\}$. Secondly, for each job, a set of preceding jobs $Prio_j$ might be defined (Ruiz et al. 2008). A few papers study hybrid flow shop problems with priorities, batching, and makespan objectives.

For a hybrid flow shop with setup times, delayed machines, lag times between stages, machine qualifications, and makespan objective, Zandieh et al. (2010) compare Nawaz-Enscore-Ham (NEH), 'shortest processing time', and 'longest processing time'. The authors conclude that NEH provides the best results for this type of problems.

A genetic algorithm for the hybrid flow shop problem with batching is proposed by Wilson et al. (2004). The initial solution for the genetic algorithm is generated by a 'longest processing time' dispatching rule, followed by the 'earliest completion time' heuristic for the following stages. A tabu search for a hybrid flow shop has been introduced by Logendran et al. (2006). In each iteration, the algorithm swaps jobs in their positions. To avoid loops, a tabu list memorizes pairs of jobs exchanged in previous iterations. The following dispatching rules are compared to construct the initial solution: Two algorithms based on'longest processing time', applied with the sum of the processing times over the stages per job, and an alphanumeric order. The paper shows that neither algorithm is better evaluated than the other.

A 'shortest processing time' heuristic for a hybrid flow shop with two stages is introduced by He et al. (2007). In the application case, jobs can only be processed by one qualified machine in the first stage. In the following stage, all jobs are processed on one machine, which uses batching.

The NEH heuristic gives good results in the area of flow shops (FM) in many cases. NEH was first formulated by Nawaz et al. (1983) for minimizing the makespan for flow shops ($FM \,||\, C_{max}$). For instance, a complex and realistic hybrid flow shop problem with changeover times, machine qualifications, and priorities is analyzed by Ruiz et al. (2008). To solve this problem, the NEH heuristic (Nawaz et al. 1983) is modified to consider M_j, *skip* and *prec* of the categorization scheme by Pinedo (2012).

Schumacher and Buchholz (2020) and Schumacher et al. (2020) present approaches to combine clustering, heuristics, metaheuristics, and discrete-event simulation to predict uncertainties in demands and schedule jobs in a realistic production. The problem is a hybrid flow shop with two stages, machine qualification, skipping stages, and uncertainties in demands, and the objective is to minimize the makespan. Distributions of scrap rates are estimated, and quantiles of the resulting distribution are used to increase production volume to avoid costly rescheduling. The second step applies 'shortest processing time', a tabu search algorithm, and local search algorithms. The best-evaluated plans are selected and evaluated in the third step by a detailed simulation model.

4 A BIASED-RANDOMIZED STOCHASTIC DISCRETE-EVENT SIMULATION ALGORITHM

The hybrid flow shop problem described in Section 2 is a complex problem considering setup times, stochastic random processing times, and job types' priority. In the studied problem, the values of processing times follow a stochastic distribution (Section 5) and are not deterministic. Laroque et al. (2022) propose an approach to solve a deterministic version of a similar problem. Their approach utilizes a multi-start biased-randomized version of the popular NEH heuristic.

The NEH heuristic starts by ranking the jobs by total processing time in descending order for each job using the average processing time of job j in stage i. The first and second jobs in the list are selected and scheduled into the permutation in both possible orders. Ordering remains the same for each processing stage. The permutation with lower C_{max} is chosen. Afterwards, each not yet scheduled job is successively tested in all possible positions of the sequences generated and selected in the previous steps. Each time, the sequence with the lowest C_{max} is chosen.

In Laroque et al. (2022), the makespan of each proposed solution is computed using a deterministic discrete-event simulation. Our work extends the previous one by considering stochastic processing times, which obviously represent an additional challenge for any optimization solver. In our approach, an additional simulation layer is introduced to evaluate the effect of the stochastic elements on the makespan. Also, the deterministic discrete-event simulation is transformed into a stochastic discrete-event simulation. Figure 2 illustrates the logic behind our algorithm. A biased-randomized heuristic component takes care of generating new solutions (permutations of jobs) as explained in Laroque et al. (2022). Then, a stochastic discrete-event simulation component is responsible for executing the proposed solution in an environment with random processing times, which allows us to get a random observation of the makespan associated with the solution proposed by the heuristic component. In our approach, we run one hundred replications to estimate the makespan of a proposed solution. In the simulation component, events are defined, e.g., finishing processing of job j in stage i. The event list is updated once an event is executed. The output of a simulation run is C_{max} that varies from one replication to another because of the introduced uncertainty of processing times, and the recorded C_{max} values are used in further analysis by the algorithm.

After several runs, the simulation component can offer an accurate estimate of the expected makespan associated with the proposed permutation of jobs. Notice that confidence intervals can also be computed with the same computational effort, which might be very useful to classify different solutions according to their quality and generate a list of 'elite' job permutation plans. At the end, the solution with the lowest expected makespan that the algorithm has found (best stochastic solution) is returned to the manager.

5 COMPUTATIONAL EXPERIMENTS

The approach in Section 4 has been implemented in Python 3.9 and tested on a generated set of instances for the problem described in Section 2. The deterministic version of the instances can be found at https://www.researchgate.net/publication/356874077_instances_flowShop. Twenty problem instances are defined in this set. The nomenclature of the instances follows the format n_m_y, where n represents the number of jobs, m represents the number of machines, and y is an identifier number for instances with the same number of jobs and machines. Four types of jobs are being processed on the machines in these instances. Each of these types has its path across the machines. The setup and processing times on the machines vary based on the job type. In our iased-randomized heuristic component, the α parameter of the geometric probability distribution was randomly chosen in the interval $(0.3, 0.4)$. Experiments were run on a computer with an Intel Xeon E5-2650 v4 with 32GB of RAM. For each instance, the makespan of its best-found solution is shown in Table 1. These solutions correspond to the deterministic version and stochastic version of the problem. In addition, the proposed permutations of jobs of the deterministic solution is evaluated under stochastic conditions.

Our best-found solutions for the deterministic version of the problem (OBD) are recorded in the second column of Table 1. For the stochastic version, random processing times have been considered. These random processing times have been modeled using the log-normal probability distribution with two parameters, μ and σ. The estimated value of the processing time of job j on machine i, $E(P_{ij})$, is the deterministic value of the processing time, p_{ij}. For these numerical experiments, the variability of processing times is assumed to be $Var(P_{ij}) = c \times p_{ij}$, where $c > 0$ is a design parameter. Hence, according to the log-normal properties:

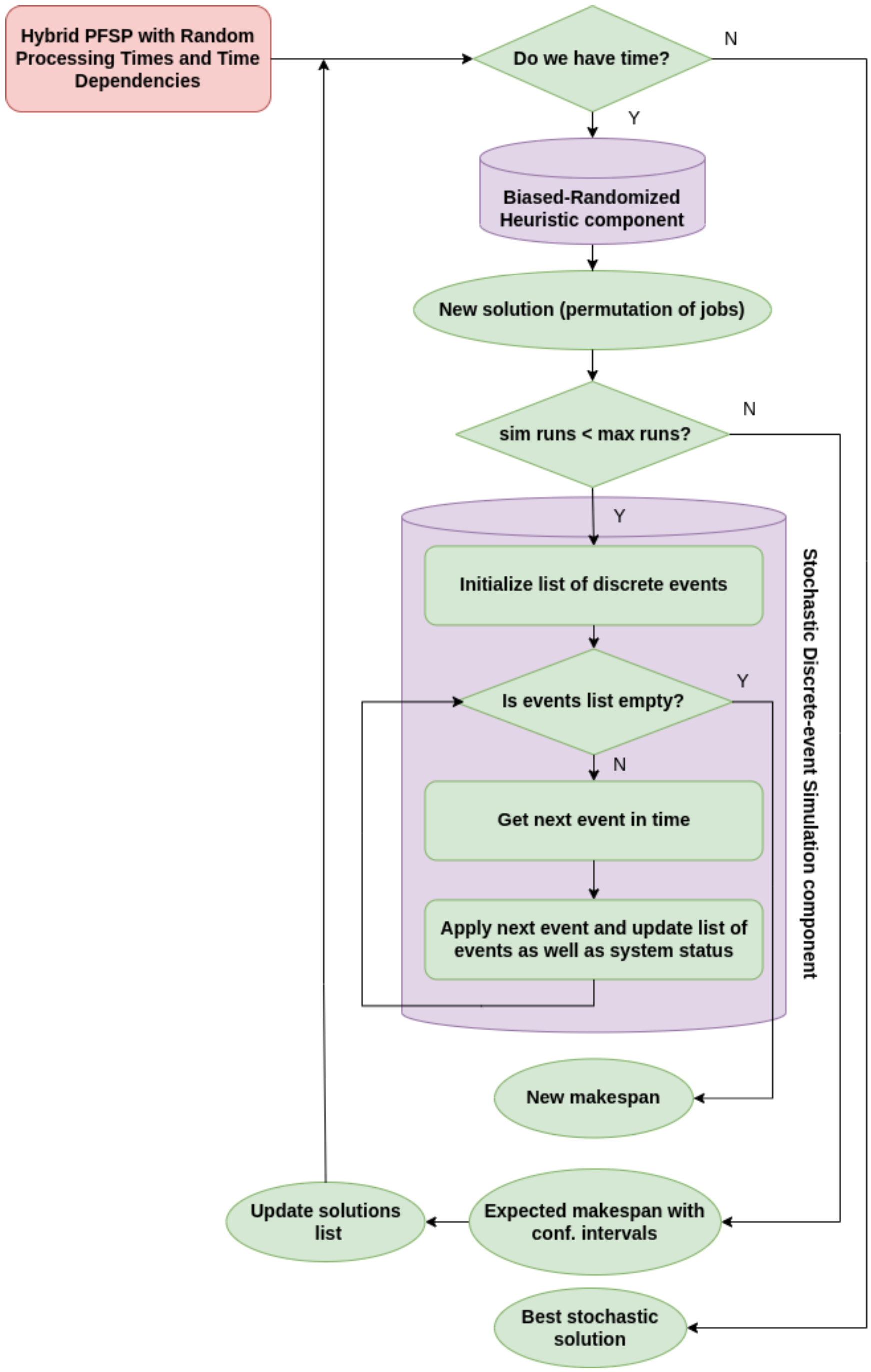

Figure 2: The logic behind our simheuristic BR-DES approach.

Table 1: Experimental results, where OBD and OBS are our best deterministic and stochastic solutions, respectively. The -L, -M, and -H refer to the level of stochasticity introduced.

		$c = 10$ (low level)		$c = 15$ (medium level)		$c = 20$ (high level)		Gaps (%)					
instance	OBD	OBD-L	OBS-L	OBD-M	OBS-M	OBD-H	OBS-H						
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(2)-(1)	(3)-(1)	(4)-(1)	(5)-(1)	(6)-(1)	(7)-(1)
30_12_4	549.50	604.57	603.46	617.51	615.20	631.26	627.29	10.02	9.82	12.38	11.96	14.88	14.16
30_12_3	541.50	585.79	583.82	608.94	601.45	618.84	604.91	8.18	7.81	12.45	11.07	14.28	11.71
30_12_5	637.50	713.87	707.87	719.94	716.48	748.29	744.12	11.98	11.04	12.93	12.39	17.38	16.72
30_12_6	690.50	748.29	744.12	757.33	742.38	774.93	767.76	8.37	7.76	9.68	7.51	12.23	11.19
30_12_7	517.50	557.71	555.48	571.21	565.00	586.91	582.08	7.77	7.34	10.38	9.18	13.41	12.48
30_12_8	438.00	479.72	473.93	488.35	474.58	502.80	500.32	9.52	8.20	11.50	8.35	14.80	14.23
30_12_10	784.50	857.31	854.07	874.77	866.65	888.90	879.87	9.28	8.87	11.51	10.47	13.31	12.16
200_12_1	3092.00	3167.17	3159.22	3190.56	3186.74	3211.10	3194.95	2.43	2.17	3.19	3.06	3.85	3.33
200_12_2	2938.20	2956.92	2952.53	2986.95	2939.34	2999.87	2992.31	0.64	0.49	1.66	0.04	2.10	1.84
200_12_3	3086.40	3137.45	3123.10	3156.24	3150.80	3187.09	3177.69	1.65	1.19	2.26	2.09	3.26	2.96
400_12_1	7592.75	7613.06	7608.02	7624.10	7602.24	7672.09	7617.28	0.27	0.20	0.41	0.13	1.04	0.32
400_12_2	6351.50	6382.38	6377.44	6389.79	6381.73	6432.86	6394.95	0.49	0.41	0.60	0.48	1.28	0.68
400_12_3	8168.10	8179.02	8172.38	8195.28	8174.77	8212.73	8184.04	0.13	0.05	0.33	0.08	0.55	0.20
Average	**2722.15**	**2767.94**	**2762.73**	**2783.15**	**2770.56**	**2805.21**	**2789.81**	**5.44**	**5.03**	**6.87**	**5.91**	**8.64**	**7.84**

$$\mu = \ln E(P_{ij}) - \frac{1}{2}\ln\left(1 + \frac{Var(P_{ij})}{E(P_{ij})^2}\right)$$

$$\sigma = \left|\sqrt{\ln\left(1 + \frac{Var(P_{ij})}{E(P_{ij})^2}\right)}\right|$$

Three stochastic scenarios were defined by varying the value of c: low, medium, and high. For each scenario, the best deterministic solutions were evaluated under the stochastic conditions, and our best stochastic solutions (OBS) were found and recorded. Thus, for instance 30_12_4, the deterministic solution under medium stochastic level (OBD-M) has a makespan of 617.51, and our best stochastic solution (OBS-M) has a makespan of 615.20 (Columns OBD-M and OBS-M in Table 1). The last columns in Table 1 displays gaps with respect to OBD, and Figure 3 displays these gaps.

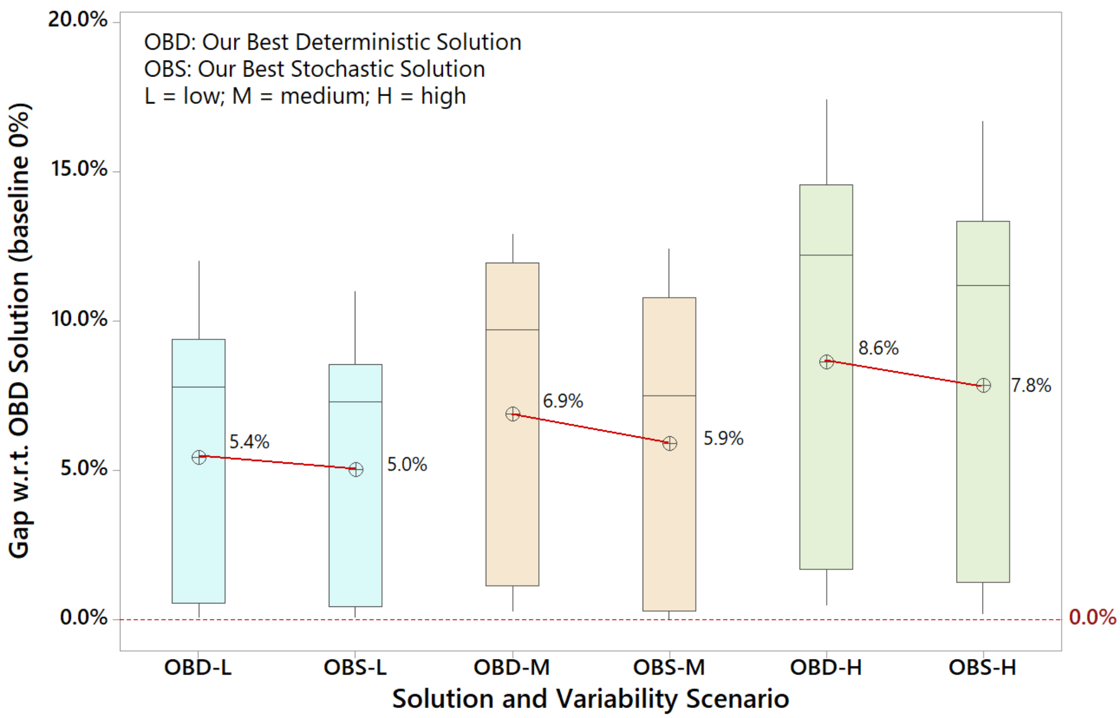

Figure 3: Gaps of the different approaches with respect to the best deterministic solution.

6 CONCLUSIONS

This paper considers a hybrid flow shop problem with random processing times, different flow paths depending on the type of job, time dependencies, and batching requirements. The model analyzed is based on a real-life system in the semiconductor industry. Due to the existence of time dependencies, it is not feasible to compute the makespan associated with a proposed solution by simply using an analytical expression. Hence, a discrete event simulation needs to be carried out each time a new solution is proposed. Apart from being time-consuming, using a pure simulation approach does not allow us, in general, to generate high-quality solutions. Moreover, the existence of random processing times makes the problem even more challenging. In order to properly address this stochastic optimization problem, which aims at minimizing the expected makespan, a simheuristic algorithm is proposed. It integrates a biased-randomized heuristic with a stochastic discrete-event simulation. The biased-randomized heuristic is in charge of proposing promising solutions, which are then submitted to the stochastic DES to assess its expected makespan. According to our numerical experiments, the resulting simheuristic is able to outperform the best solutions found for the deterministic version of the problem (one using expected processing times) when these are employed in a scenario under uncertainty. This illustrates the fact that optimal / near-optimal solutions to a deterministic version of an optimization problem become sub-optimal solutions when utilized in the stochastic version. Also, the results show that the higher the variability, the more sub-optimal the deterministic solution is.

For future work, we plan to enrich the system model using FlexSim and then connect our biased-randomized heuristic component with it by employing the Python connectivity recently provided by the aforementioned commercial simulator. Also, we plan to integrate a machine learning component that can act as a surrogate model to estimate the outcome of the simulation. This will be useful to filter out non-promising solutions and save computational time in large-sized systems.

ACKNOWLEDGMENTS

This work has been partially funded by the Spanish Ministry of Science (PID2019-111100RB-C21 /AEI/ 10.13039/501100011033), as well as by the Barcelona City Council and Fundació "la Caixa" under the framework of the Barcelona Science Plan 2020-2023 (grant 21S09355-001).

REFERENCES

Allaoui, H., and A. Artiba. 2004. "Integrating Simulation and Optimization to Schedule a Hybrid Flow Shop with Maintenance Constraints". *Computers & Industrial Engineering* 47(4):431–450.

Amaran, S., N. V. Sahinidis, B. Sharda, and S. J. Bury. 2016. "Simulation Optimization: A Review of Algorithms and Applications". *Annals of Operations Research* 240(1):351–380.

Fan, K., Y. Zhai, X. Li, and M. Wang. 2018. "Review and Classification of Hybrid Shop Scheduling". *Production Engineering* 12(5):597–609.

Ferrer, A., D. Guimarans, H. Ramalhinho, and A. A. Juan. 2016. "A BRILS Metaheuristic for Non-smooth Flow-Shop Problems with Failure-Risk Costs". *Expert Systems with Applications* 44:177–186.

Fikar, C., A. A. Juan, E. Martinez, and P. Hirsch. 2016. "A Discrete-Event Driven Metaheuristic for Dynamic Home Service Routing with Synchronised Trip Sharing". *European Journal of Industrial Engineering* 10(3):323–340.

Gholami, M., M. Zandieh, and A. Alem-Tabriz. 2009. "Scheduling Hybrid Flow Shop with Sequence-Dependent Setup Times and Machines with Random Breakdowns". *The International Journal of Advanced Manufacturing Technology* 42(1-2):189–201.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. 1979. "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey". 5:287–326.

Gruler, A., C. L. Quintero-Araújo, L. Calvet, and A. A. Juan. 2017. "Waste Collection under Uncertainty: A Simheuristic based on Variable Neighbourhood Search". *European Journal of Industrial Engineering* 11(2):228–255.

Gupta, J. N. 1988. "Two-stage, Hybrid Flowshop Scheduling Problem". *Journal of the Operational Research Society* 39(4):359–364.

Hatami, S., L. Calvet, V. Fernandez-Viagas, J. M. Framinan, and A. A. Juan. 2018. "A Simheuristic Algorithm to Set Up Starting Times in the Stochastic Parallel Flowshop Problem". *Simulation Modelling Practice and Theory* 86:55–71.

He, L., S. Sun, and R. Luo. 2007. "A Hybrid Two-Stage Flowshop Scheduling Problem". *Asia-Pacific Journal of Operational Research* 24(1):45–56.

Juan, A. A., C. G. Corlu, R. D. Tordecilla, R. de la Torre, and A. Ferrer. 2020. "On the Use of Biased-Randomized Algorithms for Solving Non-smooth Optimization Problems". *Algorithms* 13(1):8.

Komaki, G. M., S. Sheikh, and B. Malakooti. 2019. "Flow Shop Scheduling Problems with Assembly Operations: A Review and New Trends". *International Journal of Production Research* 57(10):2926–2955.

Laroque, C., M. Leißau, P. Copado, C. Schumacher, J. Panadero, and A. A. Juan. 2022. "A Biased-Randomized Discrete-Event Algorithm for the Hybrid Flow Shop Problem with Time Dependencies and Priority Constraints". *Algorithms* 15(2):54.

Lee, T.-S., and Y.-T. Loong. 2019. "A Review of Scheduling Problem and Resolution Methods in Flexible Flow Shop". *International Journal of Industrial Engineering Computations* 10(1):67–88.

Logendran, R., P. de Szoeke, and F. Barnard. 2006. "Sequence-Dependent Group Scheduling Problems in Flexible Flow Shops". *International Journal of Production Economics* 102(1):66–86.

Morais, M. d. F., M. G. Filho, and T. J. Perassoli Boiko. 2013. "Hybrid Flow Shop Scheduling Problems Involving Setup Considerations: A Literature Review and Analysis". *International Journal of Industrial Engineering* 20(11/12):614–630.

Nawaz, M., E. E. Enscore Jr, and I. Ham. 1983. "A Heuristic Algorithm for the m-Machine, n-Job Flow-Shop Sequencing Problem". *Omega* 11(1):91–95.

Nikzad, F., J. Rezaeian, I. Mahdavi, and I. Rastgar. 2015. "Scheduling of Multi-Component Products in a Two-Stage Flexible Flow Shop". *Applied Soft Computing* 32:132–143.

Onggo, B. S., J. Panadero, C. G. Corlu, and A. A. Juan. 2019. "Agri-Food Supply Chains with Stochastic Demands: A Multi-Period Inventory Routing Problem with Perishable Products". *Simulation Modelling Practice and Theory* 97:101970.

Pinedo, M. 2012. *Scheduling: Theory, Algorithms, and Systems*. 4th ed. New York: Springer.

Ruiz, R., F. S. Şerifoğlub, and T. Urlings. 2008. "Modeling Realistic Hybrid Flexible Flowshop Scheduling Problems". *Computers and Operations Research* 35(4):1151–1175.

Ruiz, R., and J. A. Vázquez-Rodríguez. 2010. "The Hybrid Flow Shop Scheduling Problem". *European Journal of Operational Research* 205(1):1–18.

Schumacher, C., and P. Buchholz. 2020. "Scheduling Algorithms for a Hybrid Flow Shop under Uncertainty". 13(11):277.

Schumacher, C., P. Buchholz, K. Fiedler, and N. Gorecki. 2020. "Local Search and Tabu Search Algorithms for Machine Scheduling of a Hybrid Flow Shop under Uncertainty". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 1456–1467. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tosun, Ö., M. Marichelvam, and N. Tosun. 2020. "A Literature Review on Hybrid Flow Shop Scheduling". *International Journal of Advanced Operations Management* 12(2):156–194.

Wilson, A. D., R. E. King, and T. J. Hodgson. 2004. "Scheduling Non-similar Groups on a Flow Line: Multiple Group Setups". *Robotics and Computer-Integrated Manufacturing* 20(6):505–515.

Zandieh, M., E. Mozaffari, and M. Gholami. 2010. "A Robust Genetic Algorithm for Scheduling Realistic Hybrid Flexible Flow Line Problems". *Journal of Intelligent Manufacturing* 21(6):731–743.

AUTHOR BIOGRAPHIES

MAJSA AMMOURIOVA is a researcher at a postdoctoral position in the Computer Science Department at the Universitat Oberta de Catalunya. She earned her PhD degree from TU Dortmund University in 2021. Her research interests are x-heuristics which includes simheuristics. Her email address is mammouriova@uoc.edu.

MADLENE LEIẞAU is a research assistant at the Institute of Management and Information at the University of Applied Sciences Zwickau, Germany and works in the research group Industry Analytics (www.industry-analytics.de) in the project KI-StudiUm. She holds a M.Sc, in management. Her main interest is the application of simulation and machine learning methods in relation to operational decision support for production. Her email address is Madlene.Leissau.Gel@fh-zwickau.de.

JAVIER PANADERO is an Associate Professor of Simulation and High Performance Computing in the Computer Science Dept. at the Universitat Oberta de Catalunya. He is also a Lecturer at the Euncet Business School. He holds a Ph.D. and a M.S. in Computer Science. His major research areas are high performance computing and simheuristics. He has co-authored more than 40 scientific articles. His website address is http://www.javierpanadero.com and his email address is jpanaderom@uoc.edu.

CHRISTIN SCHUMACHER is a researcher in the department of Computer Science at TU Dortmund University. She holds a M.Sc. in business mathematics. Her research interests include mathematical optimization, heuristics and discrete event simulation for machine scheduling and real-world optimization problems. Her e-mail address is christin.schumacher@tu-dortmund.de. Her website is https://ls4-www.cs.tu-dortmund.de/cms/de/home/schumacher/.

CHRISTOPH LAROQUE studied business computing at the University of Paderborn, Germany. Since 2013 he has been Professor of Business Computing at the University of Applied Sciences Zwickau, Germany. He is mainly interested in the application of simulation-based decision support techniques for operational production and project management. His email address is Christoph.Laroque@fh-zwickau.de.

ANGEL A. JUAN is a Full Professor in the Dept. of Applied Statistics and Operations Research at the Universitat Politècnica de València (Spain), as well as Invited Professor at University College Dublin (Ireland) and Universidade Aberta (Portugal). Dr. Juan holds a Ph.D. in Industrial Engineering and an M.Sc. in Mathematics. He completed a predoctoral internship at Harvard University and postdoctoral internships at the Massachusetts Institute of Technology and the Georgia Institute of Technology. His main research interests include applications of simheuristics and learnheuristics in computational logistics and finance. He has published over 125 articles in JCR-indexed journals and more than 300 papers indexed in Scopus. His website address is http://ajuanp.wordpress.com and his email address is ajuanp@upv.es.