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.

DEMONSTRATION OF THE FEASIBILITY OF REAL TIME APPLICATION OF MACHINE LEARNING TO PRODUCTION SCHEDULING

Amir Ghasemi

Department of IT & Logistics
Amsterdam School of International Business
Amsterdam University of Applied Sciences
Amsterdam, 1102CV, THE NETHERLANDS

Kamil Erkan Kabak

Dept. of Industrial Engineering
Izmir University of Economics
Balcova
Izmir, 35330, TURKEY

Cathal Heavey

CONFIRM Research Centre
School of Engineering
University of Limerick
Limerick, V94 T9PX, IRELAND

ABSTRACT

Industry 4.0 has placed an emphasis on real-time decision making in the execution of systems, such as semiconductor manufacturing. This article will evaluate a scheduling methodology called Evolutionary Learning Based Simulation Optimization (ELBSO) using data generated by a Manufacturing Execution System (MES) for scheduling a Stochastic Job Shop Scheduling Problem (SJSSP). ELBSO is embedded within Ordinal Optimization (OO), where in the first phase it uses a meta model, which previously was trained by a Discrete Event Simulation model of a SJSSP. The meta model used within ELBSO uses Genetic Programming (GP)-based Machine Learning (ML). Therefore, instead of using the DES model to train and test the meta model, this article uses historical data from a front-end fab to train and test. The results were statistically evaluated for the quality of the fit generated by the meta-model.

1 INTRODUCTION

The emerging Industry 4.0 concept is an umbrella term for a new industrial paradigm that embraces a set of future industrial developments regarding Cyber-Physical Systems (CPS), Internet of Things (IoT), Internet of Services (IoS), Robotics, Big Data, Cloud Manufacturing and Augmented Reality. Together, these generate and leverage on the concept of 'smart factories' to comprise the next industrial revolution in manufacturing, characterised by increased flexibility, productivity, efficiency, and sustainability, ultimately ensuring competitiveness in the global market (Weyer et al. 2015). Gathering together many of these research ideas, these revolutions will threaten to change radically the traditional Planning and Scheduling approaches within systems (Dolgui et al. 2019). Thus, in this article we attempt to advance current decision tools, with specific focus on scheduling and specifically in the semiconductor manufacturing.

Specifically, this article aims to advance the research work carried out in Ghasemi et al. (2021). In this article, an Evolutionary Learning Based Simulation Optimization (ELBSO) method within Ordinal Optimization was presented for a Stochastic Job Shop Scheduling Problem (SJSSP). Furthermore, in this article, ELBSO was compared with several published SJSSP methods, specifically Horng et al. (2012), Yang et al. (2014), and Shen and Zhu (2016). A novel aspect of the ELBSO method was the use of a meta model (Genetic Programming (GP)) as part of the solution procedure that resulted in fast execution, which could allow this solution method to be used as a real-time decision support tool. In this article, we evaluate whether it is possible to have the GP meta-model learn directly from the Manufacturing Execution System (MES). To test this, we used collected MES data that have been manually cleaned (documented later) and used them to schedule a photolithography tool, which is essentially an SJJSP.

Figure 1 details the key objective of this article, where MES data feed the GP metamodel continuously on a rolling horizon (RH) and then is used to schedule the set of photolithography tools. Note that we do not demonstrate this in a real-live setting, but from collected data, which has been cleaned (which is documented later). To evaluate how well this approach works, we statistically compare the quality of the results.

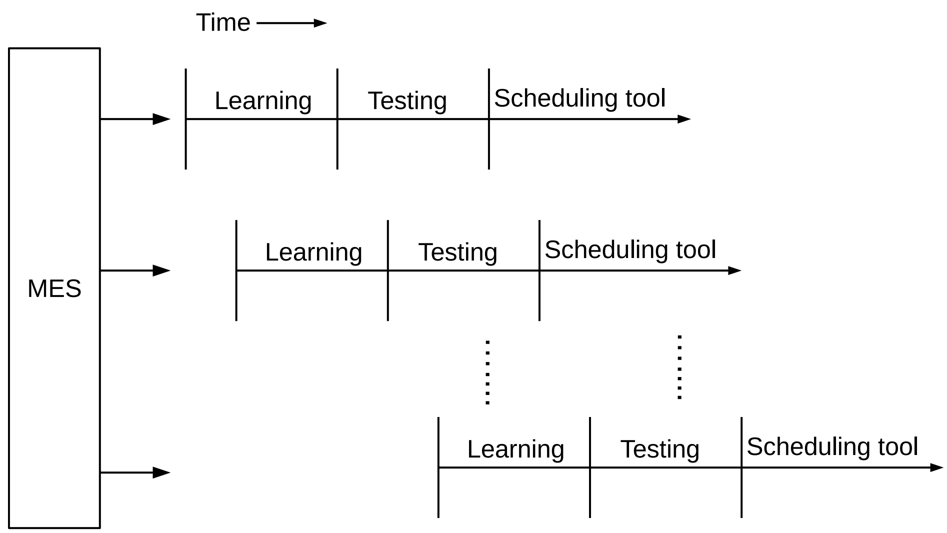

Figure 1: Within this research, the key objective is to evaluate if an ML tool can learn from from an MES. Moreover, it can be used as a Decision Support Tool (DST), which in this example a DST to schedule a photolithography tool.

In the next section, a brief literature review is carried out on scheduling of manufacturing facilities with a focus on Machine Learning (ML) techniques and real-time support. Then in Section 3 details of the problem addressed and the environment are presented, followed by a description of GP. Sections 5 and 6 present the design of the experiment and its results, and finally, the last two sections are discussions and conclusions.

2 LITERATURE REVIEW

ML techniques, which provide tasks such as data classification, clustering, reduction, regression, and anomaly detection by learning autonomously from input data (see, Kang et al. (2020), pg.3; Bertolini et al. (2021)), have prevailed and applied in various disciplines. In particular, a recent systematic literature review by Bertolini et al. (2021) they report that Supervised Learning (SL) and its methods such as Neural Networks (NNs), Support Vector Machine (SVM) and Tree-Based (TB) techniques - Decision Trees (DTs), Random Forests (RFs), Gradient Boosting (GB) - are dominant in the operations management literature. However, they highlight the growing trend towards Unsupervised Learning (UL) and Deep Learning (DL) techniques in studies conducted in the last five years. Also, Production Planning and Control (PPC) is one of main areas where the use of ML rises, and scheduling is found as the second biggest area composing about one-third of the PPC studies (Bertolini et al. 2021). In another study of ML on production lines, Kang et al. (2020) find that wafer fabrication is the process that most uses ML in the manufacturing domain. Similarly to the findings of Bertolini et al. (2021), this study also reports that SL is the main type of ML used, and Artificial Neural Networks (ANN) are the most preferred ML algorithm. Regarding algorithms, SVMs and RF are applied as second and third frequently used algorithms. Other algorithms used in production lines are DTs, LSTM (Long Short-Term Memory), KNN (K-Nearest Neighbors), GBDT (Gradient Boosted Decision Tree), and CNN (Convolutional Neural Network).

The semiconductor manufacturing environment has different complexities regarding scheduling. These are stochastic and dynamic job arrivals, unrelated machines, re-entrant visits, varying machine capabilities, varying processing times at each layer operation, sequence-dependent setups, precedence constraints, varying due dates, and machine failures. These are the reasons that scheduling problems in semiconductor manufacturing are regarded an extended version of the Flexible Job-Shop Scheduling Problem (FJSP) (Chien and Lan 2021) or Complex Flexible Job-Shop scheduling problems (Mönch, Fowler, Dauzère-Pérès, Mason, and Rose 2011). Deterministic scheduling in wafer fabs according to machine environments, process restrictions and objectives are categorized by Mönch et al. (2011). To solve this type of problem, varying Operations Research (OR) approaches including dynamic programming (DIP), mixed integer programming (MIP), constraint programming (CP) (see Bixby et al. (2006); also the review by Mönch et al. (2011)), heuristic methods (i.e., dispatching rules), and metaheuristic approaches such as Genetic Algorithms (GA) (see also Ghasemi et al. (2018), Madathil et al. (2018)) are frequently applied apart from the application of hybrid (or ensemble) approaches (see Mönch and Roob (2018)). To illustrate, Jong et al. (2020) conducted a mould scheduling problem by first applying GA to obtain better sequences, then applying Ant Colony Optimization (ACO) in order to optimize the sequence by the expert system. They note that the hybrid strategy with GA and ACO gives superior results in three case studies. However, the performance of these approaches has revealed some limitations according to optimization criteria, computational complexity, and system conditions (Chien and Lan 2021). Therefore, the scheduling literature is viewed under two main research approaches that would have the ability to alter dispatching rules with varying system states. The first is the simulation-based approaches by finding the best performance of the alternative rules and selecting the best for each state (see Kutanoglu and Sabuncuoglu (2001) and Priore et al. (2018)). The other one is referred to as Artificial Intelligence (AI) techniques or learning-based scheduling (Priore et al. 2018) or knowledge-based approach (Chien and Lan 2021) in which various ML approaches are applied to the scheduling problem. It is noted that a similar categorization of techniques on FJSP is also given by Zhou and Yang (2019) as heuristics (dispathcing rules), metaheuristics, hyperheuristics, and artificial intelligence (AI). To illustrate, Zhou and Yang (2019) presents four Multi-Objective Genetic Programming based Hyper-Heuristics (MO-GPHH) for FSJP (or MO-DFJSP for multi-objective and Dynamic FSJP (DFSJP)). They consider GP as one of the learning-based approaches for solving FSJP apart from the other approaches such as reinforcement learning, ensemble learning, ANNs and Gaussian processes. The mean weighted tardiness, maximum tardiness and mean flow time are the objectives chosen for the problem. The test bed for the application is performed by a DES model of the DFJSP. That is, simulation-based fitness analysis is performed for the Scheduling Policies (SPs) generated by MO-GPHH. Then, decoding into decision rules is employed for a GP and embedded in the DES model for decisions. They concluded that evolved SPs chosen by MO-GPHH have better performances than manual chosen SPs for any objective. Furthermore, Priore et al. (2018) analyze the effectiveness of the most widely used ensemble methods, such as bagging, boosting, and stacking, by mean tardiness and mean flow time. They use SVMs, inductive learning-based DTs, Backpropagation Neural Networks (BPNs), and Case Based-Reasoning (CBR) for a dynamic scheduling problem. Their proposed framework is to evaluate the use of ML algorithms. Accordingly, training and test examples are collected from a Flexible Manufacturing System (FMS) simulation model running under various scenarios, and the best dispatching rule is obtained for each state of the system. With knowledge of the output received from different ML algorithms, managers periodically make real-time performance decisions within a control system (Priore et al. 2018). Chien and Lan (2021) analyze an agent training framework to minimize the makespan. The problem in this study is related to parallel machine scheduling with sequence-dependent setups. The framework applies a Deep Q Network (DQN) and a hybrid GA. The structure of the proposed agent-based framework involves four parts. The first part defines a production environment by generating product families, number of machines, and jobs, as well as defining states, actions, and rewards. The second part generates training data, and the third part includes agent training where the hyper-parameter settings and DQN training occurs. The last part analyzes the performance and results (Chien and Lan 2021). In another study, Deep Reinforcement Learning (RL) is used in dynamic multi-objective scheduling by Luo et al. (2021) for an FSJP. Their study proposes a Two-Hierarchy Deep Q Network (THDQN). That is, it includes two DQN based agents. They compare four different objectives with four different reward functions. Their results prove to be better than those of the proposed THDQN on various dispatching rules. Müller et al. (2022) evaluated five constraint programming solvers and used two technologies for algorithm selection approaches. The first one are DTs that are more prevailing, the other one is Deep NNs. Their study found that the Constraint Programming (CP) solver by CPLEX outperforms Google OR-Tools if only the solution quality is considered. However, with regard to algorithm selection approaches, they observe that the DT technique based on RF gives better results. Further, Bixby et al. (2006) present Real-Time scheduling applications using STARTS (i.e., Space-Time Allocation for Real-Time Scheduling) algorithm that encapsulates MIP and CP. The algorithm works together with the fab manufacturing execution system (MES) by receiving the lot and equipment status data. The algorithm compares the feasible solutions to obtain the optimum solution using MIP and CP approaches.

With regard to stochastic job shop scheduling problem (SJSP), Tavakkoli-Moghaddam et al. (2005) find feasible points by a hyrid method of NN and SA. In addition, Gu et al. (2009) presents a parallel quantum genetic algorithm (QGA) for SJSP. QGA is selected to have a better global search approach with fast convergence. In their study, expected makespan is minimized by a stochastic expected value model. Further on multi-objective SJSP, Lei et al. (2007) propose an evolutionary approach. However, little research has been done on stochastic job-shop scheduling in semiconductor manufacturing. Furthermore, it is noted that the stochastic unrelated machine scheduling problem is reported to be strongly NP-hard (Zhou and Yang 2019; Mati and Xie 2004).

Metamodeling provides a mathematical relationship between a set of input variables to estimate output variables and a design strategy is employed to predict the response more effectively (Tunali and Batmaz 2000). Graph-Based Machine Learning (GBML) is applied mainly in metamodeling techniques. This type of learning includes nodes and edges to define the features and the interactions among the features in the system. The other recent metamodeling approaches include the use of NNs (Dunke and Nickel (2020)) and gene regulatory fuzzy cognitive maps (Liu et al. (2019)). With regard to GP and metamodelling, learning techniques applied are different in comparison to the other above mentioned studies. More specifically, by using the evolutionary algorithms, GP tries to learn a mathematical function through approximation of the relationship between the input and the output data, i.e., symbolic regression (Koza 1994).

3 PROBLEM DESCRIPTION

In this section, the considered problem within this article is detailed. Consider a photolithography workstation at time point t (assume t is always within the production time). From the production scheduling point of view, there are two sets of wafers within the photolithography workstation at time point t: a) in process wafers; b) queued wafers waiting (on a predefined queue) for production. Note that these wafers can be from different lots (Within the front-end fab, wafers usually flow within batches called lots). In addition to the above, consider the scenario that, at time point t, lot l arrives to the workstation. Accordingly, the Processing Time (PT) of wafers within lot l can be calculated as follows:

$$PT(l) = f(R_t, S_l, \varepsilon) \tag{1}$$

where R_t refers to the number of wafers remaining inside the workstation (both categories mentioned above), S_l is the size of lot l (number of wafers in the lot), and ε is the error value. Note that, from the production scheduling literature point of view, each wafer can be considered as an operation or job.

In this article, we present an ML-based metamodeling framework to estimate PT for each lot, at the arrival moment t, based on the MES data obtained from a semiconductor manufacturing fab. In other words, we attempt to replace function f in Equation 1 with f' obtained from a trained ML-based metamodel as follows:

$$PT'(l) = f'(R_t, S_l, \varepsilon) \tag{2}$$

where $PT'(l)$ refers to the estimated processing time of lot l in the photolithography workstation at time point t.

To illustrate, Figure 2 shows a photolithograpy workstation at time point t. At this time point, Lot l arrives to the workstation including n wafers. Each wafer is shown by xxx, l, w referring to an arbitrary wafer id, the lot number, and the wafer number, respectively. In time point t, both first and second wafers of lot l are in process (In Process), while there is a group of wafers waiting on the queue to be processed (On Queue). The final goal is to estimate the processing time of wafers on Lot l according to the current state of the workstation. Note that the processing time of all wafers in each lot are considered equal, as they all flow together within the production line and batch changing is not allowed, which is a critical assumption in most semiconductor front-end fabs.

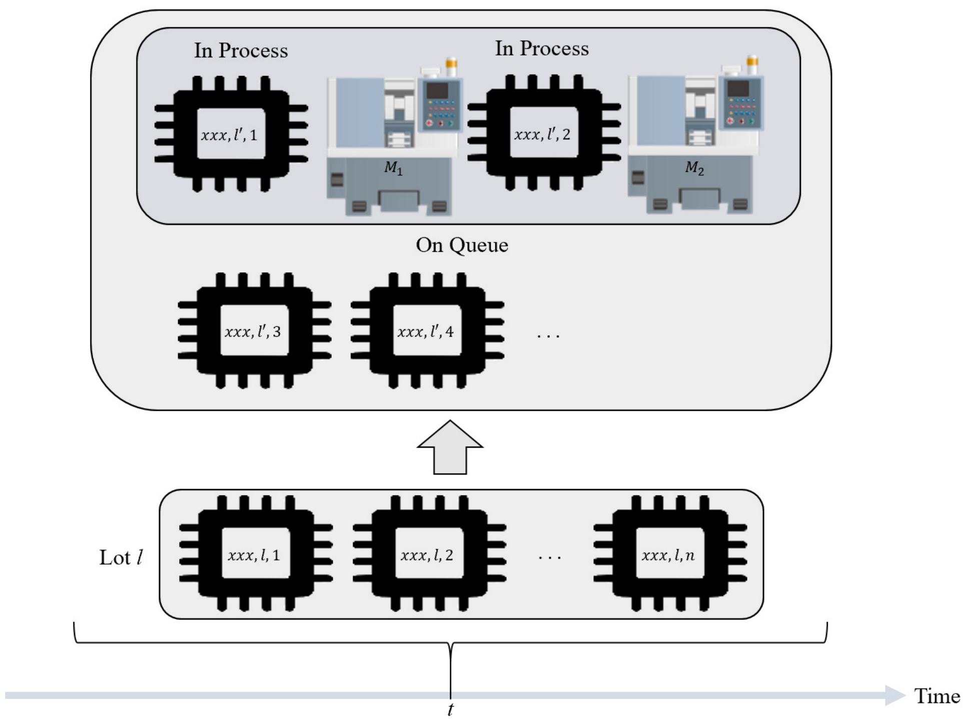

Figure 2: An example of the problem environment.

To transform Equation 1 into Equation 2, GP is selected as the ML-based metamodeling tool. It is worth mentioning that this selection is made based on the results of previous successful GP applications to production planning and scheduling problems such as Ghasemi and Heavey (2021), Ghasemi et al. (2021), Can and Heavey (2016). The following section details the architecture of the implemented GP.

4 GENETIC PROGRAMMING

With regard to Genetic Programming (GP), it is one of the most used approaches for generating Dispatching Rules (DRs) for scheduling problems (Ghasemi and Heavey 2021). Generally, GP is defined as an evolutionary algorithm and one of the hyper-heuristic approaches that are applied for obtaining new heuristics for different types of combinatorial optimization problems ((Durasević and Jakobović 2022); (Zhang et al. 2021)). GP is mainly preferred approach for its flexibility for representation and its promising results, also as a hyper-heuristic it works on a heuristic search space with the help of other heuristics (Zhang et al. 2021). With regard to scheduling problems, GP allows obtaining superior Dispatching Rules (DRs) as well as automation of having new DRs (Durasević and Jakobović 2022).

In this research, using the HeuristicLab tool version 3.3, ((HeuristicLab 2022)) we designed a GP metamodel. Note that a full GP architectural information with the same software package is provided in Ghasemi et al. (2021). In this research, we follow the GP parameter design strategy provided by Ghasemi et al. (2021). Consequently, five parameters are used to calibrate the proposed model. These parameters are Population Size (PS), Mutation Probability (MP), Maximum Generations (MG), Tree Length (TL), and Tree Depth (TD). The values of these parameters for metamodel generation were found by following Taguchi's method and preliminary test runs in Heuristic Lab. The values 1000, 0.20, 500, 150 and 25 are selected for PS, MP, MG, TL and TD, respectively, after finding the results of main effects (**?**).

5 EXPERIMENTS DESIGN

As mentioned above, in this research, the obtained data sets are from a semiconductor manufacturing front-end fab. Accordingly, data was collected for 3 months from a front-end fab with a data set consisting of 929,178 rows of data. To clarify, these data sets include the size of lots (TRACKINMAINQTY), photolithography entrance times in each route (TRACKINTIME), and photolithography exit times in each route (TRACKOUTTIME) (see Figure 3). Using track in and track out times of each lot, the processing time for each lot could be easily calculated. Moreover, they enabled us to detect the number of lots inside the photolithography area in each time point. It is worth mentioning here that all data sets are sampled from the MES and verified by experts before analysis. Since some lots do not require the photolithography in each route, their processing time in the photolithography area equals to zero. Therefore, all jobs with processing times equal to zero are deleted.

The information from 1000 lots in sequence was extracted at their arrival to the workstation. Next, Equation 1 is built for each lot. Finally, 700 (70%) rows of data (700 lots data) and 300 rows of data (30%) are used for training and testing the GP model. Note that all experiments in this article are executed on a PC with 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz CPU and 16 GB RAM. In the following section, the results of both training and testing are provided.

6 EXPERIMENTS RESULTS

In this research, using the HeuristicLab tool version 3.3, (HeuristicLab 2022) we designed a GP metamodel with the discussed parameters. To evaluate GP solutions, we used the Mean Relative Error (MRE), which is known as one of the most practical metrics for evaluating estimation techniques (Park and Stefanski 1998). The general formulation of MRE is shown in Equation 3 as follows:

$$MRE = \frac{1}{ss} \times \sum_{b=1}^{ss} \frac{|Y_b' - Y_b|}{Y_b}, \tag{3}$$

where Y_b' defines the estimated value (here $PT'(l)$) of the factor Y_b (here $PT(l)$) in the sample b, and total number of samples equals ss. Table 1, shows the results of GP implementation in 10 different replications.

To demonstrate the accuracy of the metamodel, Figure 4 shows example results for the training and testing of the the GP metamodel. Figure 4 shows an extracted portion that demonstrates the quality of the metamodel. As mentioned above, in the training of the GP metamodel, 700 rows of data (samples) where

TRACKINTIME	TRACKOUTTIME	TRACKINMAINQTY
Datetime	Datetime	Number
TRACKINTIME	TRACKOUTTIME	TRACKINMAINQTY
12/06/2015 19:50:43	12/06/2015 19:50:43	1
12/06/2015 19:50:44	12/06/2015 20:12:52	1
10/06/2015 17:19:23	10/06/2015 17:19:23	1
10/06/2015 17:19:25	10/06/2015 17:55:32	1
01/07/2015 21:56:56	01/07/2015 21:56:56	1
01/07/2015 21:56:57	01/07/2015 22:31:13	1
20/10/2017 14:17:50	20/10/2017 14:17:50	1
20/10/2017 14:27:33	20/10/2017 14:27:33	1
10/04/2015 01:35:26	10/04/2015 01:35:27	7
10/04/2015 01:35:28	10/04/2015 02:02:25	7
18/04/2015 21:38:17	18/04/2015 21:38:17	7
18/04/2015 21:38:18	18/04/2015 22:11:29	7
30/07/2015 14:49:32	30/07/2015 14:49:32	1
30/07/2015 14:49:33	30/07/2015 15:13:27	1
19/05/2015 19:15:50	19/05/2015 19:15:51	25
19/05/2015 19:15:52	19/05/2015 19:58:14	25
06/05/2015 13:52:56	06/05/2015 13:52:56	25
06/05/2015 13:52:58	06/05/2015 14:24:18	25
25/02/2016 20:17:54	25/02/2016 20:17:55	13
25/02/2016 20:17:55	25/02/2016 20:51:35	13
01/07/2016 03:07:33	01/07/2016 03:42:22	3
22/04/2015 14:27:19	22/04/2015 14:27:20	13
22/04/2015 14:27:21	22/04/2015 15:00:28	13
24/06/2016 15:25:37	24/06/2016 15:25:37	3
24/06/2016 15:25:38	24/06/2016 15:55:26	3
25/06/2016 22:50:09	25/06/2016 22:50:09	3
25/06/2016 22:50:10	25/06/2016 23:26:43	3

Figure 3: Photolithography workstation data set sample.

used for training and 300 rows of data where used for testing. It is quite clear that GP has a good quality not only in predicting solution fluctuations, but also their objective values.

To provide more insights on the metamodeling quality in this research, Figure 5 shows the scatter plot for results obtained for the first GP replication. Considering Figure 5, the correlation between $PT(l)$ values their estimated values ($PT'(l)$, which is shown by 'Estimated Value') is clearly demonstrated.

7 DISCUSSION

This section discusses the experiment results demonstrated in Section 6. First, the MRE values for the test data are shown in Table 1. By analyzing Table 1 results, it can be concluded that the trained GP-based metamodel has an average accuracy of about 75% (between 25% to 28% errors) in predicting PT values in different test replications. It is worth mentioning that, within provided experiments, the whole training and testing process takes less than four minutes (ranging between 214 to 226 seconds) making the proposed

Table 1: GP replications MRE and execution time (CPU Time).

Replication	1	2	3	4	5	6	7	8	9	10
MRE	0.26	0.28	0.26	0.25	0.26	0.27	0.28	0.26	0.26	0.26
CPU Time (s)	221	216	223	226	222	214	216	224	221	226

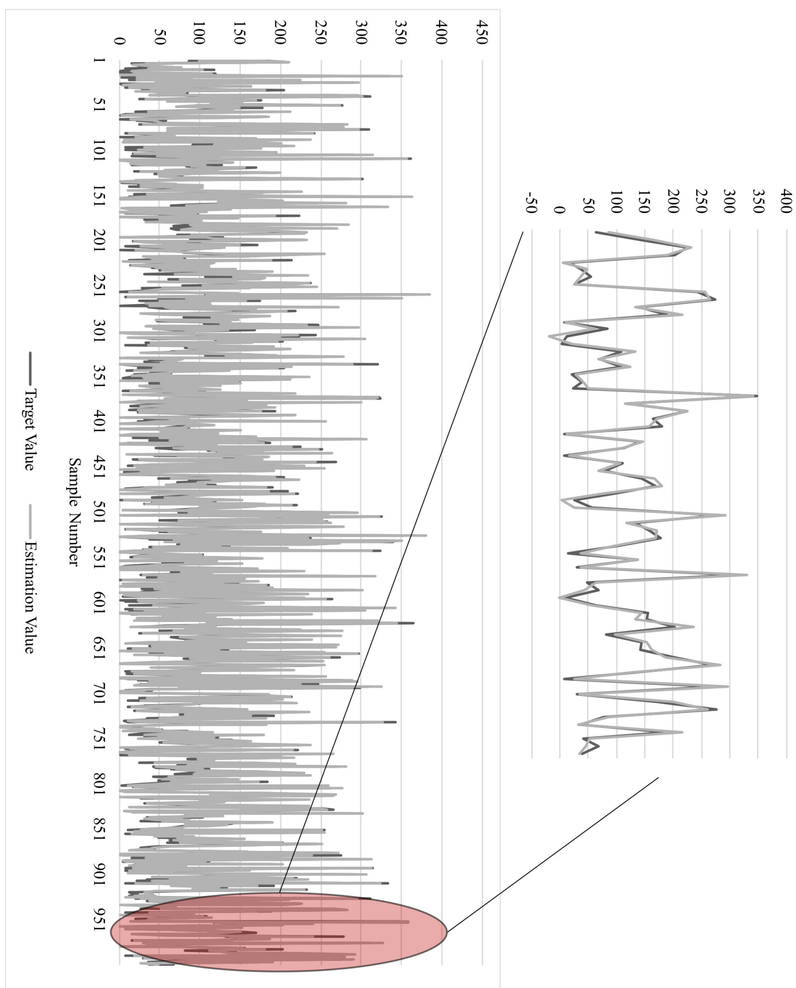

Figure 4: GP line chart result for the first test replication.

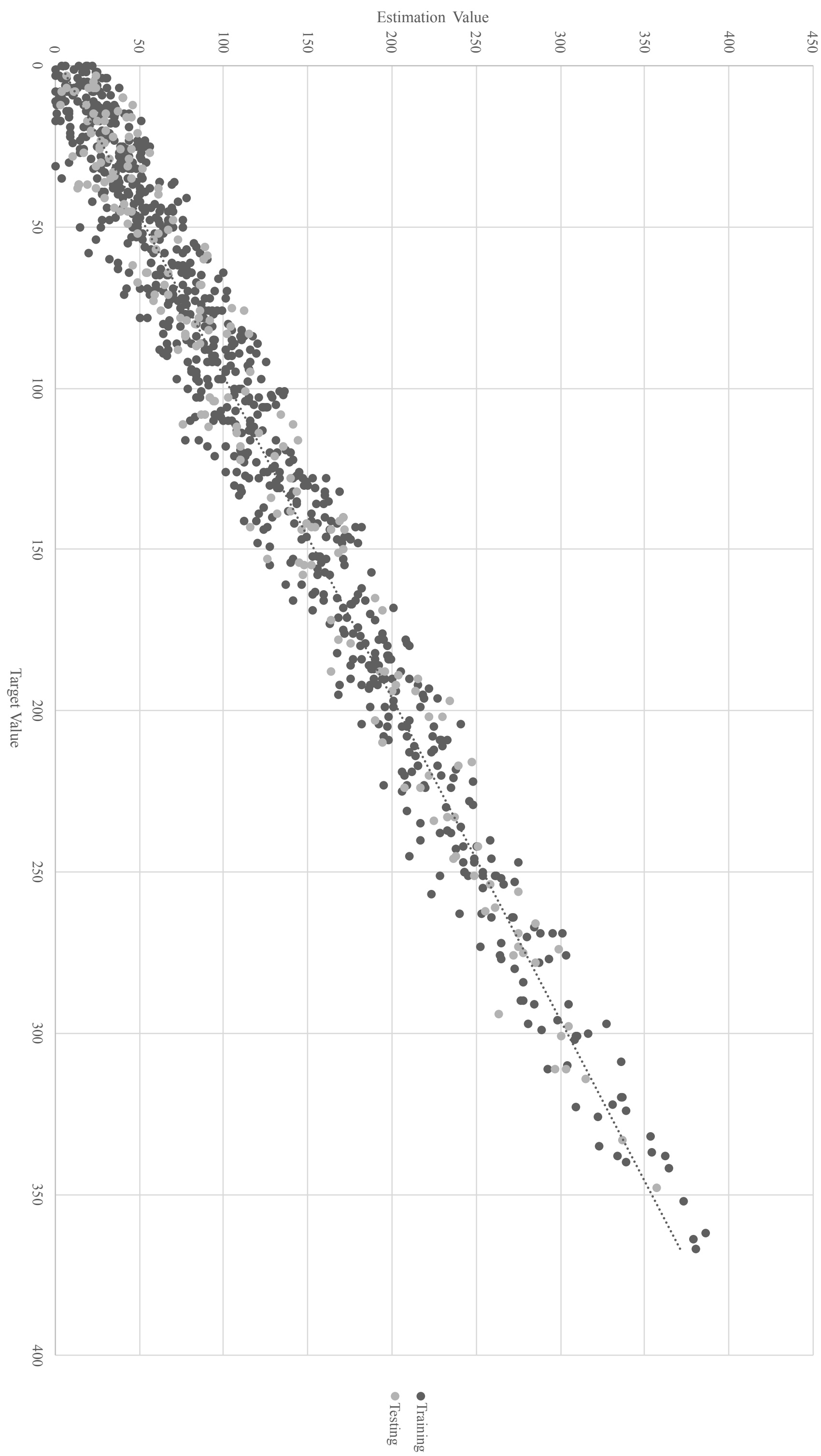

Figure 5: GP scatter plot result for the first test replication.

metamodel capable of being used in real-time applications. To assess the quality of the proposed metamodel GP, in this investigation, graphically, Figures 4 and 5 demonstrate the difference between the values of $PT(l)$ (Target Value) and $PT'(l)$ (Estimation Value) using the line graph and the correlation scatter plot. From Figure 4, especially the test portion extracted, it can be clearly concluded that the proposed metamodel follows trends within the data and the estimated $PT(l)$ values are reasonably accurate. Lastly, Figure 5 shows a significant correlation between the values $PT(l)$ and $PT'(l)$.

8 CONCLUSIONS

Recently, notable digital breakthroughs have taken place in production systems worldwide, referred to as Industry 4.0. According to Industry 4.0, it is required to be able to monitor and predict the performance of manufacturing systems in a real time, which has mainly fostered the emergent of concepts such as Digital Twins (DTs). With regard to this, the article attempted to provide a real-time application of ML-based metamodeling to train from MES data and estimate the processing time of operations using GP. In this research, the MES data from a semiconductor manufacturer (as semiconductor industry is one of the leading industries in implementing Industry 4.0), is used to train the ML-based metamodel. The accuracy of the proposed methodology was validated using both statistical and graphical analysis.

Although our research showed promising results especially in real-time MES data metamodeling, it has some limitations that we are willing to address in future research. We used only 1000 rows of data for training and testing, which can be increased in future research. We only used GP as the ML tool, thus we are interested to assess the quality of other regression-based ML methods within the same problem context. Finally, towards designing DTs for decision making within the Industry 4.0 era, we are interested in integrating the designed metamodel, in this article, with optimizers to provide real-time DT-based decision support tools for manufacturing systems.

ACKNOWLEDGMENTS

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI 16/RC/3918, co-funded by the European Regional Development Fund.

REFERENCES

Bertolini, M., D. Mezzogori, M. Neroni, and F. Zammori. 2021. “Machine Learning for Industrial Applications: A Comprehensive Literature Review”. *Expert Systems with Applications* 175:114820.

Bixby, R., R. Burda, and D. Miller. 2006. “Short-Interval Detailed Production Scheduling in 300mm Semiconductor Manufacturing using Mixed Integer and Constraint Programming”. In *2006 IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, 148–154. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Can, B., and C. Heavey. 2016. “A Demonstration of Machine Learning for Explicit Functions for Cycle Time Prediction using MES Data”. In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 2500–2511. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Chien, C.-F., and Y.-B. Lan. 2021. “Agent-based approach integrating deep reinforcement learning and hybrid genetic algorithm for dynamic scheduling for Industry 3.5 smart production”. *Computers & Industrial Engineering* 162:107782.

Dolgui, A., D. Ivanov, S. P. Sethi, and B. Sokolov. 2019, January. “Scheduling in Production, Supply Chain and Industry 4.0 Systems by Optimal Control: Fundamentals, State-Of-The-Art and Applications”. *International Journal of Production Research* 57(2):411–432.

Dunke, F., and S. Nickel. 2020. “Neural networks for the metamodeling of simulation models with online decision making”. *Simulation Modelling Practice and Theory* 99:102016.

Durasević, M., and D. Jakobović. 2022. “Selection of Dispatching Rules Evolved by Genetic Programming in Dynamic Unrelated Machines Scheduling based on Problem Characteristics”. *Journal of Computational Science* 61:101649.

Ghasemi, A., A. Ashoori, and C. Heavey. 2021. “Evolutionary Learning Based Simulation Optimization for Stochastic Job Shop Scheduling Problems”. *Applied Soft Computing* 106:107309.

Ghasemi, A., and C. Heavey. 2021, December. "An Evaluation of Strategies for Job Mix Selection in Job Shop Production Environments - Case: A Photolithography Workstation". In *Proceedings of the 2021 Winter Simulation Conference*, edited by S. Kim, B. Feng, K. Smith, S. Masoud, Z. Zheng, C. Szabo, and M. Loper, 1–12. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Ghasemi, A., C. Heavey, and K. E. Kabak. 2018. "Implementing a New Genetic Algorithm to Solve the Capacity Allocation Problem in the Photolithography Area". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 3696–3707. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Gu, J., X. Gu, and M. Gu. 2009. "A Novel Parallel Quantum Genetic Algorithm for Stochastic Job Shop Scheduling". *Journal of Mathematical Analysis and Applications* 355(1):63–81.

HeuristicLab 2022. "A Paradigm-Independent and Extensible Environment for Heuristic Optimization". Last Accessed September 30, 2022. https://dev.heuristiclab.com/trac.fcgi/.

Horng, S.-C., S.-S. Lin, and F.-Y. Yang. 2012, February. "Evolutionary Algorithm for Stochastic Job Shop Scheduling with Random Processing Time". *Expert Systems with Applications* 39(3):3603–3610.

Jong, W.-R., H.-T. Chen, Y.-H. Lin, Y.-W. Chen, and T.-C. Li. 2020. "The Multi-Layered Job-Shop Automatic Scheduling System of Mould Manufacturing for Industry 3.5". *Computers & Industrial Engineering* 149:106797.

Kang, Z., C. Catal, and B. Tekinerdogan. 2020. "Machine Learning Applications in Production Lines: A Systematic Literature Review". *Computers & Industrial Engineering* 149:106773.

Koza, J. R. 1994, June. "Genetic Programming as a Means for Programming Computers by Natural Selection". *Statistics and Computing* 4(2):87–112.

Kutanoglu, E., and I. Sabuncuoglu. 2001. "Experimental Investigation of Iterative Simulation-Based Scheduling in a Dynamic and Stochastic Job Shop". *Journal of Manufacturing Systems* 20(4):264–279.

Liu, J., Y. Chi, Z. Liu, and S. He. 2019. "Ensemble Multi-Objective Evolutionary Algorithm for Gene Regulatory Network Reconstruction based on Fuzzy Cognitive Maps". *CAAI Transactions on Intelligence Technology* 4(1):24–36.

Luo, S., L. Zhang, and Y. Fan. 2021. "Dynamic Multi-Objective Scheduling for Flexible Job Shop by Deep Reinforcement Learning". *Computers & Industrial Engineering* 159:107489.

Madathil, S. C., S. Nambiar, S. J. Mason, and M. E. Kurz. 2018. "On Scheduling a Photolithography Area Containing Cluster Tools". *Computers & Industrial Engineering* 121:177–188.

Mati, Y., and X. Xie. 2004, January. "The Complexity of Two-Job Shop Problems with Multi-Purpose Unrelated Machines". *European Journal of Operational Research* 152(1):159–169.

Mönch, L., J. W. Fowler, S. Dauzère-Pérès, S. J. Mason, and O. Rose. 2011, December. "A Survey of Problems, Solution Techniques, and Future Challenges in Scheduling Semiconductor Manufacturing Operations". *Journal of Scheduling* 14(6):583–599.

Mönch, L., and S. Roob. 2018. "A Matheuristic Framework for Batch Machine Scheduling Problems with Incompatible Job Families and Regular Sum Objective". *Applied Soft Computing* 68:835–846.

Müller, D., M. G. Müller, D. Kress, and E. Pesch. 2022, January. "An Algorithm Selection Approach for the Flexible Job Shop Scheduling Problem: Choosing Constraint Programming Solvers Through Machine Learning". *European Journal of Operational Research* 302:874–891.

Park, H., and L. A. Stefanski. 1998. "Relative-Error Prediction". *Statistics & Probability Letters* 40(3):227–236.

Priore, P., B. Ponte, J. Puente, and A. Gómez. 2018. "Learning-Based Scheduling of Flexible Manufacturing Systems Using Ensemble Methods". *Computers & Industrial Engineering* 126:282–291.

Shen, J., and Y. Zhu. 2016, June. "Chance-Constrained Model for Uncertain Job Shop Scheduling Problem". *Soft Computing* 20(6):2383–2391.

Tavakkoli-Moghaddam, R., F. Jolai, F. Vaziri, P. Ahmed, and A. Azaron. 2005, November. "A Hybrid Method for Solving Stochastic Job Shop Scheduling Problems". *Applied Mathematics and Computation* 170(1):185–206.

Tunali, S., and I. Batmaz. 2000. "Dealing with the Least Squares Regression Assumptions in Simulation Metamodeling". *Computers & Industrial Engineering* 38(2):307–320.

Weyer, S., M. Schmitt, M. Ohmer, and D. Gorecky. 2015, January. "Towards Industry 4.0 - Standardization as the Crucial Challenge for Highly Modular, Multi-Vendor Production Systems". *IFAC-PapersOnLine* 48(3):579–584.

Yang, H.-a., Y. Lv, C. Xia, S. Sun, and H. Wang. 2014. "Optimal Computing Budget Allocation for Ordinal Optimization in Solving Stochastic Job Shop Scheduling Problems". *Mathematical Problems in Engineering* 2014:1–10.

Zhang, F., Y. Mei, S. Nguyen, K. C. Tan, and M. Zhang. 2021. "Multitask Genetic Programming-Based Generative Hyperheuristics: A Case Study in Dynamic Scheduling". *IEEE Transactions on Cybernetics*:1–14.

Zhou, Y., and J.-j. Yang. 2019. "Automatic Design of Scheduling Policies for Dynamic Flexible Job Shop Scheduling by Multi-Objective Genetic Programming based Hyper-Heuristic". *Procedia CIRP* 79:439–444.

AUTHOR BIOGRAPHIES

AMIR GHASEMI is an Assistant Professor in the department of IT & Logistics at Amsterdam School of International Business (AMSIB). He published papers in the field of Simulation, Optimization, and Machine Learning-based Decision Support Tools for operations, transportation, and supply chain planning. His research interests include designing Simulation, Optimization, and Machine Learning-based Smart Agents in order to replace and/or support the human in decision making with a wide range of application domains including Digitalization, Sustainability, and Circular Economy. His email address is: a.ghasemi2@hva.nl.

KAMIL ERKAN KABAK is an Assistant Professor in the Department of Industrial Engineering, Izmir University of Economics. He received the Bachelor's degree from the Middle East Technical University in Ankara, Turkey, the Master's degree from the Department of Industrial Engineering, Dokuz Eylul University, Izmir, Turkey, and the Ph.D. degree from the Department of Design and Manufacturing Technology, University of Limerick, Limerick, Ireland. His research interests include combinatorial optimization, data analytics, simulation modeling and decision support systems. His email address is: erkan.Kabak@ieu.edu.tr.

CATHAL HEAVEY is an Associate Professor in the School of Engineering at the University of Limerick. He is an Industrial Engineering graduate of the National University of Ireland (University College Galway) and holds an M. Eng.Sc. and Ph.D. from the same University. He has published in the areas of queuing and simulation modeling. His research interests include simulation modeling of discrete-event systems; modeling and analysis of supply chains and manufacturing systems; process modeling; and decision support systems. His email address is Cathal.Heavey@ul.ie.