SPS2022
A.H.C. Ng et al. (Eds.)

doi:10.3233/ATDE220192

Bottleneck Detection Through Data Integration, Process Mining and Factory Physics-Based Analytics

Mahesh Kumbhar [a,1], Amos H.C. Ng [a,b] and Sunith Bandaru [a]
[a] *School of Engineering Science, University of Skövde, Sweden*
[b] *Division of Industrial Engineering and Management, Uppsala University, Sweden*

Abstract. Production systems are evolving rapidly, thanks to key Industry 4.0 technologies such as production simulation, digital twins, internet-of-things, artificial intelligence, and big data analytics. The combination of these technologies can be used to meet the long-term enterprise goals of profitability, sustainability, and stability by increasing the throughput and reducing production costs. Owing to digitization, manufacturing companies can now explore operational level data to track the performance of systems making processes more transparent and efficient. This untapped granular data can be leveraged to better understand the system and identify constraining activities or resources that determine the system's throughput. In this paper, we propose a data-driven methodology that exploits the technique of data integration, process mining, and analytics based on factory physics to identify constrained resources, also known as bottlenecks. To test the proposed methodology, a case study was performed on an industrial scenario were a discrete event simulation model is built and validated to run future what-if analyses and optimization scenarios. The proposed methodology is easy to implement and can be generalized to any other organization that captures event data.

Keywords. Process Mining, Factory Physics, Data Analytics, Manufacturing, Bottleneck Detection

1. Introduction

The efficiency of any organization is dependent on the rate of output under various resource constraints. In production environments, efficiency is generally measured in terms of the throughput of the production system [1]. Since improving throughput is also one of the direct means for increasing profitability, manufacturing companies are always exploring efficient ways of getting the most out of their production systems. To get higher throughput, performance improvements need to be made on a constraint activity that is restricting throughput. Such an activity is called a bottleneck [2]. The activity in question can refer to any resource, such as a machine, a manufacturing cell, a human operator, a conveyor, etc. Finding bottleneck activity in a system is not a trivial task. In the age of digitization, manufacturing companies capture tons of data regarding process activities.

[1]Corresponding Author: Mahesh Kumbhar, School of Engineering Science, University of Skövde, Skövde, Sweden; E-mail: mahesh.kumbhar@his.se

The concept of process mapping was developed as a way to better understand processes within a system and their inter-relationships. Due to the static nature of process mapping, eventually process mining came to be, which uses process data to analyze operational activities dynamically [3]. Process event data extracted from activities can be used to create dynamic process flows. By analyzing different events captured by the data, improvement activities can be performed to increase the throughput of the system. A review of the literature reveals that the use of process mining for improving the productivity of production systems has not been adequately explored [4].

The term "bottleneck" has different definitions based on various methods in the literature and methods proven using simulation and validation with industrial cases. For example, the utilization factory physics-based method was simulated on a serial line considering constant arrival rate of the products [5], [6] but not explored and validated when the arrival rate is not constant. Hence, considering general process data captured by modern manufacturing companies, a generalized methodology has not yet been developed to identify bottleneck activities and improve system throughput.

In this paper, we introduce a three-step methodology for transforming process data into metrics that can be used to identify and quantify bottlenecks. The three steps are: (i) *data integration*, where process event-logs are integrated with other organization order data, (ii) *process mining*, in which the actual process model is derived using event-log, and (iii) *factory physics-based analytics*, in which a simulation model is developed by abstract process map derived from actual process map and utilization-based statistics are derived to quantify *busyness* of various activities. The novelty of this research is fourfold. First, it proposes a data-driven procedure for exploiting existing process data to create an actual process model. Second, it introduces the use of abstracted process models to create a simulation model. Third, using the concept of factory physics, utilization measures of various activities are calculated to detect and quantify bottlenecks of the system. And finally, a methodology that can be easily implemented using process mining tools and basic utilization calculations performed dynamically to analyze bottlenecks within any organization that maintains event logs. The calculated descriptive statistics of the utilization of all activities are explored to detect probable bottleneck activity restricting the throughput of the system. In a future study, exploration of root-cause of a particular activity, and subsequent improvement is expected to improve system throughput. This cycle can be continuously repeated when other system activities become a bottleneck.

This paper is structured as follows. Section 2 gives an overview of the literature on process mining, bottleneck detection and factory physics fundamentals. Section 3 proposes a three-step methodology to detect and quantify bottlenecks in any organization which logs operational event data. Section 4 reports the application of the methodology to a real-world case study. Finally, Section 5 concludes the paper and mentions our future research directions.

2. Literature Review

Van der Aalst [3] explains the importance of process mining for extracting knowledge from event-logs to discover, monitor and improve processes. An event-log constitutes various timestamped events for different activities involved in a system over a specified time. Process mining is a valuable tool to understand processes dynamically by exploiting such event level operational data. As operational data contains case-wise timestamps and events, time-perspective process discovery can be carried out, visualising the bottleneck activity. Lorenz et al. [4] propose a data-driven procedure using process mining. An event-log is used to generate process models, and the bottlenecks are identified as machines having the highest cycle time. Evaluation of model is performed using *replay fitness* of the as-is process model, process conformance is obtained for all observed traces, and improvement areas are suggested to enhance the productivity of assembly process line. The main limitation of bottleneck detection through process mining is that it can only represent bottlenecks visually but cannot quantify them. Therefore, next we review the available literature on bottleneck detection methods, which are classified into simulation, data-driven and analytical methods.

To analyze the complex nature of real-time processes, Law et al. [7] proposed a simulation-based method that aims to emulate real-world processes with computer models which can be experimented. The authors propose a simulation-based utilization approach, in which a server that is busy for the highest duration during the simulation time is considered a bottleneck. Roser et al. [8] proposed active and inactive time periods as a way to identify bottlenecks. The activity having the longest average active time is considered a bottleneck. Active state refers to working, waiting, in-repair, and set-up. Inactive state refers to waiting for the arrival of a part. This general approach of an active period can be applied to any activities of manufacturing like machines, buffers, AGV. Further, the active period concept was extended by Roser et al. [9] who differentiate bottlenecks as sole and shifting bottlenecks. Shifting bottleneck occurs when an overlap of the active period of the current bottleneck (active period) exists with previous and subsequent bottleneck (active period) at any particular time t. In the period of shifting bottleneck, no machine is the sole bottleneck, and in the non-overlap active period, there is the sole bottleneck. Such an approach has been simulated and tested on job shop and flow shop production scenarios. To improve production systems, the dynamics of factories was analyzed by Kasemset and Voratas [5] who proposed a bottleneck identification method that uses the theory of constraints. A bottleneck machine is identified if all three requirements (high utilization of machine, high utilization factor and low bottleneck rate) are satisfied by any of the machines. The approach is simulated on single and multiple bottleneck test scenarios.

Sengupta et al. [10] proposed a method by considering the inter-departure time of each resource. Resource state is broken down into cycle time (busy), idle time (blocked-up), blocked-down and failure state. Failure state time is removed from inter-departure time, blocked-up and blocked-down time added, and the resource with the lowest blocked-up and blocked-down time is considered as a bottleneck. Leopris and Zdenka [11] try to combine bottleneck detection methods based on the criteria of utilization, starvation, blocking, and waiting for labor. A machine is identified as the bottleneck if it has the minimum difference of system average rate with individual rate. Yu and Andrea [12] attempted to combine all bottleneck detection methods together and compare bottlenecks identified with different methods j on machine i to find a reliable metric. Statistical testing is conducted with the null hypothesis of a bottleneck metric mean X_{ij} being equal to another bottleneck metric mean Y_{ij}, and alternate hypothesis of X_{ij} being less than Y_{ij}. If the computed p-value from the test is less than the critical value, bottleneck condition of X_{ij} is significant. This analysis is conducted for all bottleneck conditions and identified bottleneck machines.

Simulation-based methods are described based on deterministic statistical parameters of the system. Therefore, they typically cannot capture variations from real data. Data-driven bottleneck detection methods aim to use such existing data to detect bottlenecks. Chiang et al. [13] proposed a data-driven algorithm based on real-time starvation ST_i and blockage BT_i probabilities on serial production systems. Arrows based on blockage and starvation probabilities are placed on each machine, an arrow drawn to a machine m_{i+1} from machine m_i if ST_{i+1} is greater than BT_i. To strengthen the method of starvation and blockage, Li et al. [14] proposed a method by exploring characteristics of bottleneck machines in terms of blockage and starvation. In a serial line, for each machine, characteristics like blockage and starvation time are measured. Based on this information on shop floor bottleneck machine identified when machine having lowest blockage and starvation time. Concerning the bottleneck machine, the upstream machine has the highest blockage time, and the downstream machine has the highest starvation time observed. Change in state of blocking to starvation on bottleneck machine defined as turning point method. The concept of an active period can be extended with diagnostic capability utilizing bottleneck type, determined based on elapsed time in each active state of the machine, and necessary action is taken to eliminate that particular bottleneck type. Several real-time case studies were performed on serial production lines by Subramaniyan et al. [15]. Lai et al. [16] extended the turning point method to complex systems where complex systems converted into serial lines using pseudo station replacing with the main line and detecting bottleneck using turning point method. The method has been applied to a complex industrial case study.

Apart from simulation-based and data-driven approaches, Wu [17] developed an analytical method. The author quantifies the throughput bottleneck of a machine or a factory using the utilization and variability of the system. The variability of the system is measured in terms of flow variability and process variability. The throughput bottleneck machine has the highest utilization in the system, and if all machines utilization is below 100%, there is no throughput bottleneck in the system. It depends on machine utilization rate, effective processing time, variation of arrival rate and variation of service rate. Here, the effective processing time is measured based on cycle time except waiting for another batch. Machine utilization is difficult to measure; hence inventory information is used to measure machine utilization. The variability of the machine is extended to the variability of the factory using utilization, arrival rate, variability and effective process time. The dynamics of factory predicted using utilization and factory personnel can able to know how variability will affect the production performance.

Simulation-based approaches are most commonly applied for bottleneck detection, but they are still not well-developed to directly process real-time shop floor data for dynamic bottlenecks that may change on a weekly and even daily basis. Thus, there is a need to explore hybrid methods which can handle complex (non-serial) production systems and generate digital twins (i.e., simulation models with connections to real-time data) for dynamic bottleneck detection. This research is a step towards this direction in which the factory physics-based bottleneck theory is implemented into a digital twin to process arrival and processing rate data from a real production system for the purpose of detecting the bottleneck machine on a weekly basis.

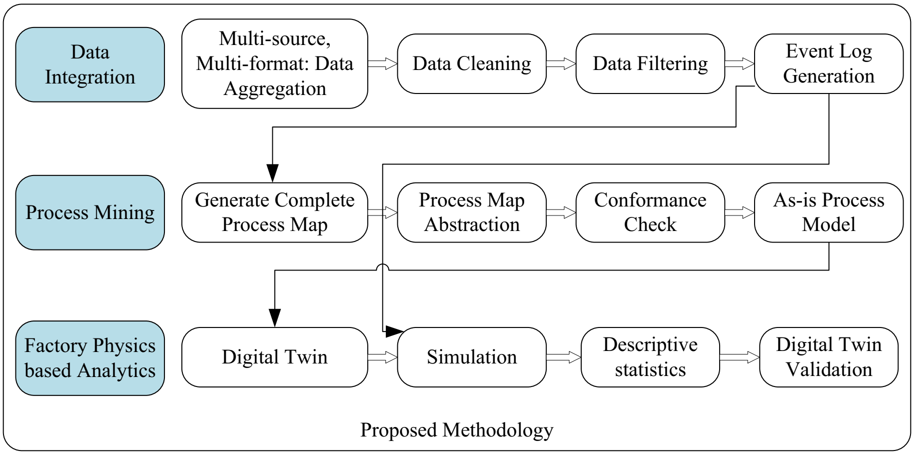

Figure 1. The proposed three-step methodology for detecting and quantifying bottlenecks.

3. Proposed Methodology

The proposed methodology, shown in Figure 1, consists of three main steps: data integration, process mining, and factory physics-based analytics. The methodology utilizes event-logs to discover the as-is process flow dynamically. Knowledge obtained from process discovery is used to build an abstract digital model of a system that can simulate industrial data with factory physics-based utilization measures to detect and quantify bottlenecks. In this way, the bottlenecks are identified and quantified using a hybrid approach of process mining, simulation and factory physics-based analytics.

3.1. Data Integration

To satisfy customer requirements, manufacturing companies perform a sequence of operations to deliver value-added products or services and while performing operations, transaction data is captured. Process execution is performed on several activities with a series of events performed at an instance level recorded into the system. Multi-source and multi-format data is aggregated to track operational activities as events. Data cleaning is performed to remove default process execution steps like initialization of events and validate all activities names of the system. Data filtering is applied to refine data which consists of an actual process with a subset of events in each activity. Data can be shown in terms of `CaseId`, `Activity`, timestamps, and other case attributes. `CaseId` refers to a unique instance identity. `Activity` represents the execution of a process performed on a particular station (machine or counter). Activity performed on a `CaseId` is recorded with start and end timestamp. Event-log is arranged by `CaseId` with respective events and timestamps.

3.2. Process Mining

In any system, processes play a pivotal role in generating output using limited resources. To achieve the highest outcome with constrained resources, manufacturing companies

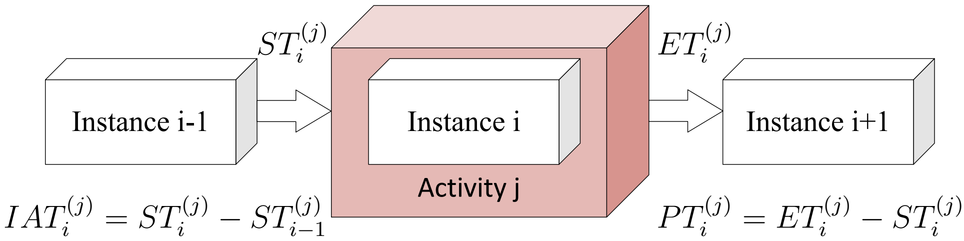

Figure 2. Calculation of inter-arrival time and processing time on activity j.

try to make processes very efficient to deliver output. Initially, process map tools like value stream maps were generated to analyze processes at a given point in time to make efficient processes. To analyze processes in the current digitization age, process mining tools can be used to discover processes dynamically using operational event data. Process Mining generates the actual process model using event logs depicting various activities and paths followed by all `CaseIds`. Activity inclusion criteria perform an abstraction of the process map to target a particular part of the system under consideration. Conformance checking of as-is processes with a designed model gives insights about variations of actual processes.

3.3. *Factory Physics-based Analytics*

The abstract process map is used to develop a *digital twin*, which is a replica of a physical system consisting of a collection of activities that act and interact together toward the accomplishment of the logical goal. The digital twin is developed to experiment with an actual system using a mathematical model which imitates the real system by changing inputs to the system to see how it affects output performance. System performance is measured in terms of the utilization of the system to achieve desired goal [16]. Utilization of activity is defined as $u = r/c$, where r is input rate to activity and c is capacity of activity [17]. This definition of utilization applies to any manufacturing company having complex events routing where we can determine possible bottleneck activity. Input rate is calculated based on an inter-arrival rate of case ID on a particular activity, and the capacity of activity is obtained in terms of processing time. Here, processing time refers to actual process time, including setup, failure and other possible detractors.

Figure 2 shows on activity j, instance i start process at ST_i start time and complete process at ET_i having processing time of $PT_i = ET_i - ST_i$ and inter-arrival time $IAT_i = ST_i - ST_{i-1}$. The utilization of an activity is calculated as the busy ratio BR, given by:

$$BR = median\left(\frac{InputRate}{OutputRate}\right) = median\left(\frac{\frac{1}{IAT_{i+1}}}{\frac{1}{PT_i}}\right) = median\left(\frac{ET_i - ST_i}{ST_{i+1} - ST_i}\right) \quad (1)$$

The busy ratio can be calculated for all activities in the system to measure their utilization over a given time period. An activity with significantly high utilization can be thought of as constraining the output rate of the system and hence identified as a bottleneck. The results are validated with digital twin simulation to identify bottleneck activity.

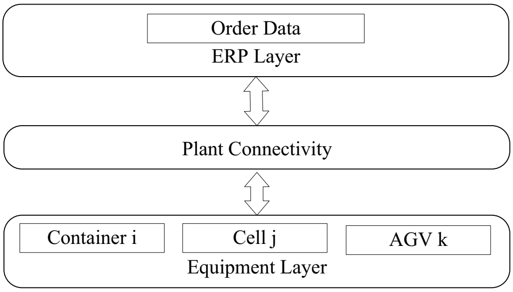

Figure 3. Data integration from ERP and MES systems.

4. Results and Discussion on an Industrial Case Study

In this section, we apply the proposed methodology to a real industrial case study from a highly automated cellular manufacturing facility in Gothenburg, producing multiple variants of assembled products with complex routing. The facility performs machining operations on some of the components, while others are available directly for the assembly operation from a different facility which is not considered in the current study. The components are moved in containers between all the machining and assembly cells using AGVs. Each cell is preceded by one or more buffers that are known never to reach their full capacity. In the following sections, we describe the application of each step of the proposed methodology to this use case.

4.1. Data Integration

The data, collected during a period of four months, comes from multiple sources and in multiple formats as shown in Figure 3. Data concerning various orders and their quantities are captured into the ERP layer. At the equipment layer, each order is split into multiple batches. Any given batch can be associated with a corresponding container i. In addition, each container is associated with a cell or buffer location j, and an AGV k. The three main sources of data referred to as (i) order data, (ii) container data, and (iii) transport data, are described below:

1. **Order Data:** Table 1 shows an extract of the order data for a randomly selected batch ‘2751542’. It depicts the sequence of activities performed on the components. `EventType` describes a series of process execution steps for a particular component being processed on a given cell. `OrderId` represents a series of operations performed on a cell for a particular quantity, while `Material` represents the name of a component. `ContainerId` defines a container assigned to a particular component for a given order with a respective timestamp.
2. **Container Data:** Table 2 shows the events associated with the movement of a particular container ‘1051864’ carrying 15 units of ‘Compo. 1’ from location ‘PSC 1 ut’ to ‘ORC 2 in’ through an intermediate buffer.
3. **Transport Data:** The movement of the containers is executed by AGVs which generate corresponding transport data. For example, in Table 3 a container ‘1051864’ is being transported with `TransportId` ‘1627656’ using an AGV from

Table 1. Order data captured at the ERP layer.

EventDateTime	EventType	OrderId	Material	BatchId	Qty	ContainerId	Location
2021/04/17 11:55:10	100	61799	Comp. 1	2751542	2	1051864	ORC 2 in
2021/04/17 12:10:48	110	61799	Comp. 1	2751542	2	1053083	ORC 2 ut
2021/04/17 14:26:39	100	61824	Comp. 1	2751542	3	1052671	ORC 2 in
2021/04/17 14:27:41	110	61824	Comp. 1	2751542	3	1053196	ORC 2 ut

Table 2. Container data captured at the equipment layer.

EventDateTime	EventType	ContainerId	Location	Material	Quantity
2021/04/16 07:15:35	20	1051864	PSC 1 ut	Comp. 1	15
2021/04/16 08:08:55	21	1051864	PSC 1 ut	Comp. 1	15
2021/04/16 08:12:12	20	1051864	Buffer ORC 2	Comp. 1	15
2021/04/17 10:27:32	21	1051864	Buffer ORC 2	Comp. 1	15
2021/04/16 10:29:00	20	1051864	ORC 2 in	Comp. 1	15
2021/04/16 11:55:08	21	1051864	ORC 2 in	Comp. 1	15

Table 3. AGV transport data captured at the equipment layer.

EventDateTime	TransportId	FromLocation	ToLocation	Carrier	ContainerId
2021/04/16 08:03:32	1627656	PSC 1	Buffer ORC 2	AGV	1051864
2021/04/16 08:03:33	1627656	Buffer ORC 2	ORC 2	AGV	1051864

Table 4. Event-log generated by integrating order data, container data and transport data.

Batch	ContainerId	Quantity	Location	Start Time	End Time
2751542	1051724	15	Buffer ORC 2	2021/04/16 05:54	2021/04/21 00:50
2751535	1049070	15	PSC 1 in	2021/04/16 05:56	2021/04/16 07:11
2751535	1051825	15	PSC 2 ut	2021/04/16 07:00	2021/04/16 07:03
2751535	1051825	15	Buffer ORC 2 in	2021/04/16 07:06	2021/04/17 10:06

cell 'PSC 1' to 'ORC 2'. The transport data provides a higher level of detail than the container data.

For process mining to work, the event-log needs to have fields for `CaseId`, `EventId`, `Activity`, and their corresponding `StartTime` and `EndTime` stamps. In the current study, `CaseId` refers to batches consisting of processed components and assembled products. Each batch comprises a `ContainerId` that can be treated as the `EventId` for process mining. Each `ContainerId` is processed at a location or cell, which becomes the `Activity`. The raw data provided by the company was missing identifiers for the mapping of components to assembly operations. Common identifiers were derived to track all components to the assembly operation. The complete event-log over four months is obtained by combining all batches. Table 4 shows an extract of this event-log for the batch '2751535'.

4.2. *Process Mining*

In the current case study, all events are recorded into ERP and MES systems with timestamp, time perspective process discovery used to generate models without using a-priori

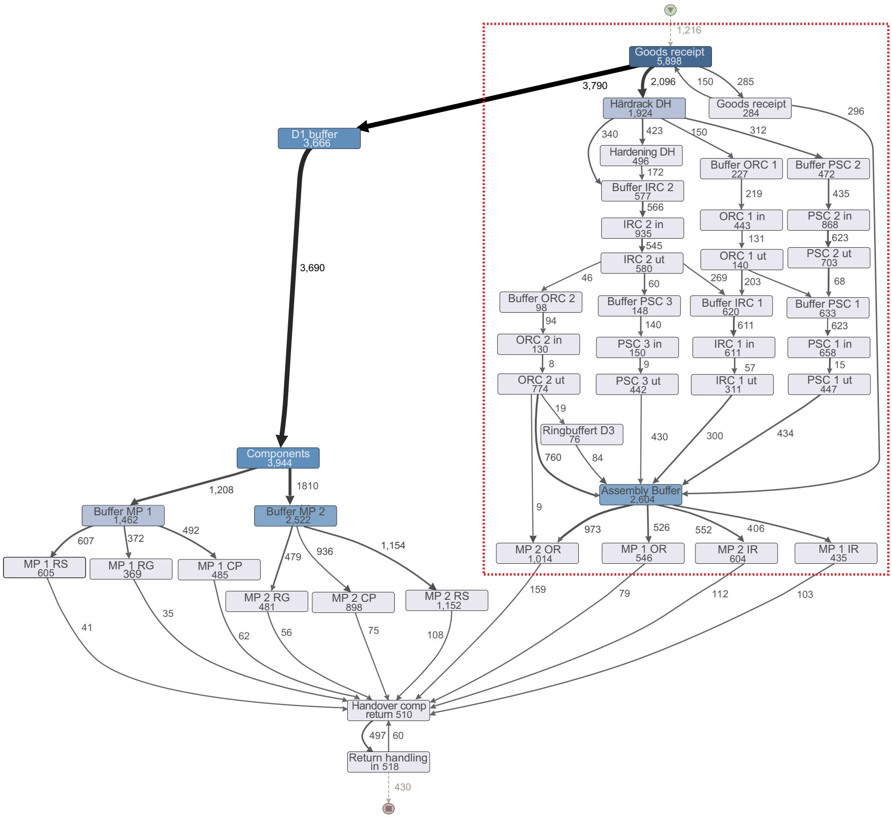

Figure 4. An abstraction of the full-scale process map generated by DISCO. The rectangular region encloses the part of the process map considered in this study.

information [3]. In addition, conformance checking is used to compare the as-is process model with the real system.

Figure 4 shows the abstracted process map obtained using the process mining software DISCO [18]. The process map shows material inventory location 'Goods Receipt' separating components into produced components and procured components that are used directly in the assembly operation. Each cell is preceded by a unique or combined buffer. 'Components' indicates multiple procured components that go directly for assembly operation (MP1 and MP2). 'Assembly Buffer' indicates buffer prior to assembly operation for in-house process components. Components from 'Assembly Buffer', and 'Buffer MP' combine together to form an assembled product. 'Handover comp return' represents the aggregation of assembled products coming from MP1 and MP2.

4.3. Factory Physics-based Analytics

The digital twin shown in Figure 5 is modeled in FACTS Analyzer 3.1 [19] using the part of the abstracted process map bounded by the rectangle in Figure 4. It consists of two assembly machines, seven machining cells and multiple buffers. In the abstract process model, for each machining cell, two activities labeled 'in' and 'ut' were generated be-

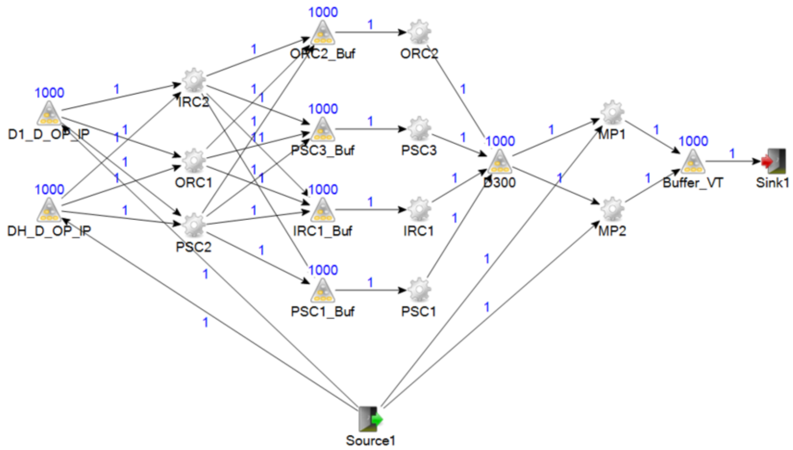

Figure 5. Digital twin of a part of the abstracted process flow shown in the rectangle in Figure 4.

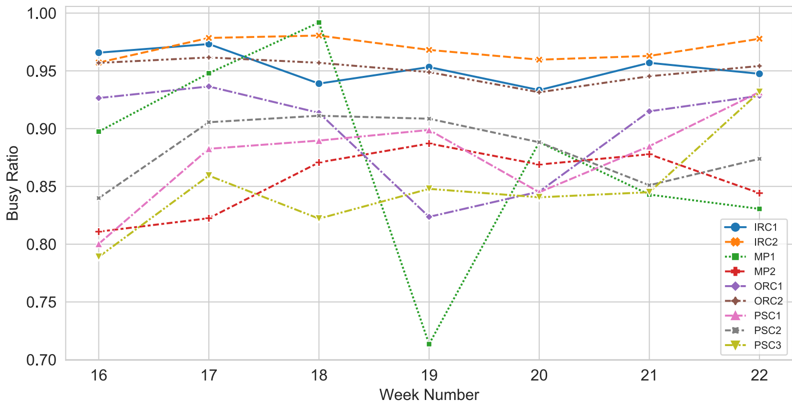

Figure 6. Weekly utilization of the cells in terms of busy ratio.

cause of different timestamps recorded for input and output from the cell. These activities are merged in the digital twin to represent one cell in the digital twin.

For each cell, the utilization is calculated week-wise from Equation (1) and plotted in Figure 6. It can be seen that IRC2 is consistently the bottleneck in almost all weeks during the given time period. The event-log is also given as an input to the FACTS simulation engine to calculate utilization statistics for each cell in different weeks. Busy ratio box-plot statistics for weeks 19 and 20 generated by FACTS are shown in Figure 7. The box-plot shows the variation of the busy ratio for cells in a particular week. For weeks 19 and 20, cell IRC2 shows the least variation of busy ratio statistics. It signifies that this cell is constraining the system and negatively impacts the throughput of whole the facility.

The proposed methodology bridges the gap between data-driven and simulation-based bottleneck detection methods. As digitization is at every corner of the process, there is a growing need to develop a general methodology that can be utilized to improve

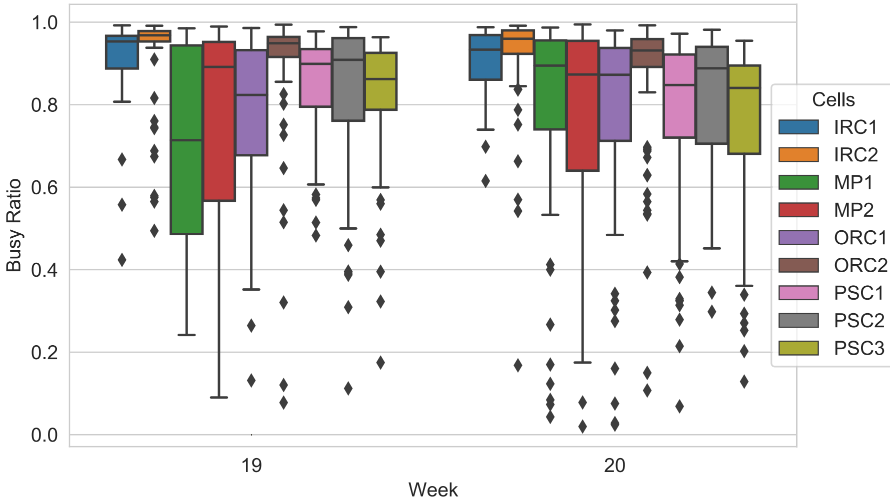

Figure 7. Boxplots for the busy ratio of cells obtained using FACTS for weeks 19 and 20.

the throughput of a system. After analyzing the utilization of activities, the next step is to investigate improvement opportunities for throughput by taking various measures such as reducing cycle time, increasing buffer capacity, increasing availability by implementing preventive maintenance, and reducing downtime.

5. Conclusions and Future Work

Digitalization in the manufacturing sector has opened up huge opportunities for utilizing the vast amounts of data that modern production systems generate at every level of production. Analysis of such granular data can not only help in identifying and diagnosing problems, but can also suggest what kind of improvements have to be implemented to make the systems more efficient. In this paper, we have presented such a data-driven methodology for bottleneck detection. The methodology consists of three steps: (i) data integration, (ii) process mining, and (iii) factory physics-based analytics. Starting with process-level event data coming from multiple sources and in multiple formats, the first step involves cleaning, filtering, and integrating the data into a common event log. Next, the event log is used to generate a full-scale process map using available process mining tools. An abstraction of the full-scale map is used to build a discrete event simulation model. Finally, based on the factory physics definition of utilization, we calculate the "busyness" of various resources to quantify the bottlenecks. The calculated metrics are validated against simulations, and the resource with the highest busy ratio is reported as the bottleneck. The methodology has been demonstrated in a real industrial case study from a highly automated cellular manufacturing facility with a complex routing, producing multiple variants of assembled products. The proposed methodology is not specific to this industrial case, and can be applied to any system containing activities/resources where event level data is captured.

As part of future work, we intend to embedded the assembly logic into the simulation model for performing what-if analysis with respect to various improvements that can be made to the industrial case. Optimal levels of improvement at each resource for a specified throughput can also be obtained through simulation-based multi-objective optimization. Without accurate knowledge of the location of the bottleneck, any efforts and investments in improving the capacity of the system might be sub-optimal at best or in vain at worst. Therefore, developing effective methods to identify the true bottleneck is paramount to any system improvement activities in the organization.

References

[1] W. J. Hopp and M. L. Spearman, *Factory physics*. Waveland Press, 2011.
[2] E. M. Goldratt and J. Cox, *The goal: A process of ongoing improvement*. Routledge, 2016.
[3] W. van der Aalst, *Process mining: Data science in action*. Springer, 2016.
[4] R. Lorenz, J. Senoner, W. Sihn, and T. Netland, “Using process mining to improve productivity in make-to-stock manufacturing,” *International Journal of Production Research*, vol. 59, no. 16, pp. 4869–4880, 2021.
[5] C. Kasemset and V. Kachitvichyanukul, “Simulation-based procedure for bottleneck identification,” in *Asian Simulation Conference*. Springer, 2007, pp. 46–55.
[6] C. Betterton and S. Silver, “Detecting bottlenecks in serial production lines–a focus on interdeparture time variance,” *International Journal of Production Research*, vol. 50, no. 15, pp. 4158–4174, 2012.
[7] A. M. Law and W. D. Kelton, *Simulation modeling and analysis*. Mc GrawHill, 2000.
[8] C. Roser, M. Nakano, and M. Tanaka, “A practical bottleneck detection method,” in *Proceeding of the 2001 Winter Simulation Conference*, vol. 2. IEEE, 2001, pp. 949–953.
[9] ——, “Shifting bottleneck detection,” in *Proceedings of the 2002 Winter Simulation Conference*, vol. 2. IEEE, 2002, pp. 1079–1086.
[10] S. Sengupta, K. Das, and R. P. Vantil, “A new method for bottleneck detection,” in *Proceedings of the 2008 Winter Simulation Conference*. IEEE, 2008, pp. 1741–1745.
[11] M. Leporis and Z. Králová, “A simulation approach to production line bottleneck analysis,” in *International Conference Cybernetics and Informatics*, 2010, pp. 13–22.
[12] C. Yu and A. Matta, “A statistical framework of data-driven bottleneck identification in manufacturing systems,” *International Journal of Production Research*, vol. 54, no. 21, pp. 6317–6332, 2016.
[13] S.-Y. Chiang, C.-T. Kuo, and S. M. Meerkov, “DT-bottlenecks in serial production lines: Theory and application,” *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 567–580, 2000.
[14] L. Li, Q. Chang, J. Ni, G. Xiao, and S. Biller, “Bottleneck detection of manufacturing systems using data driven method,” in *2007 IEEE International Symposium on Assembly and Manufacturing*. IEEE, 2007, pp. 76–81.
[15] M. Subramaniyan, A. Skoogh, H. Salomonsson, P. Bangalore, M. Gopalakrishnan, and A. Sheikh Muhammad, “Data-driven algorithm for throughput bottleneck analysis of production systems,” *Production & Manufacturing Research*, vol. 6, no. 1, pp. 225–246, 2018.
[16] X. Lai, H. Shui, D. Ding, and J. Ni, “Data-driven dynamic bottleneck detection in complex manufacturing systems,” *Journal of Manufacturing Systems*, vol. 60, pp. 662–675, 2021.
[17] K. Wu, “An examination of variability and its basic properties for a factory,” *IEEE Transactions on Semiconductor Manufacturing*, vol. 18, no. 1, pp. 214–221, 2005.
[18] Fluxicon BV. (2022) Process mining for professionals. [Online]. Available: https://fluxicon.com/disco/
[19] A. H. C. Ng, M. Urenda Moris, and J. Svensson, “Multi-objective simulation optimization for production systems design using facts analyser,” in *Swedish Production Symposium*, 2008, pp. 101–109.