

Article

Using Worker Position Data for Human-Driven Decision Support in Labour-Intensive Manufacturing

Ayse Aslan ^{1,*} , Hanane El-Raoui ² , Jack Hanson ³ , Gokula Vasantha ^{1,*} , John Quigley ² , Jonathan Corney ³  and Andrew Sherlock ⁴

¹ The School of Computing, Engineering and The Built Environment, Edinburgh Napier University, Edinburgh EH10 5DT, UK

² Strathclyde Business School, University of Strathclyde, Glasgow G1 1XQ, UK

³ School of Engineering, The University of Edinburgh, Edinburgh, EH8 9YL, UK

⁴ National Manufacturing Institute Scotland, Glasgow PA3 2EF, UK

* Correspondence: a.aslan@napier.ac.uk (A.A.); g.vasantha@napier.ac.uk (G.V.)

Abstract: This paper provides a novel methodology for human-driven decision support for capacity allocation in labour-intensive manufacturing systems. In such systems (where output depends solely on human labour) it is essential that any changes aimed at improving productivity are informed by the workers' actual working practices, rather than attempting to implement strategies based on an idealised representation of a theoretical production process. This paper reports how worker position data (obtained by localisation sensors) can be used as input to process mining algorithms to generate a data-driven process model to understand how manufacturing tasks are actually performed and how this model can then be used to build a discrete event simulation to investigate the performance of capacity allocation adjustments made to the original working practice observed in the data. The proposed methodology is demonstrated using a real-world dataset generated by a manual assembly line involving six workers performing six manufacturing tasks. It is found that, with small capacity adjustments, one can reduce the completion time by 7% (i.e., without requiring any additional workers), and with an additional worker a 16% reduction in completion time can be achieved by increasing the capacity of the bottleneck tasks which take relatively longer time than others.

Keywords: industrial productivity; process mining; discrete event simulation; indoor positioning systems; completion time; flexible capacity allocation



Citation: Aslan, A.; El-Raoui, H.; Hanson, J.; Vasantha, G.; Quigley, J.; Corney, J.; Sherlock, A. Using Worker Position Data for Human-Driven Decision Support in Labour-Intensive Manufacturing. *Sensors* **2023**, *23*, 4928. <https://doi.org/10.3390/s23104928>

Academic Editor: Miguel Delgado-Prieto

Received: 14 April 2023

Revised: 13 May 2023

Accepted: 19 May 2023

Published: 20 May 2023



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

1. Introduction

Labour-intensive manufacturing systems [1], like workshops, often experience deviations from planned production schedules due to human workers' unpredictable and flexible behaviour. This behaviour differs from automated machines, which perform tasks as prescribed. Deviations can occur even with fixed task assignments due to workers' abilities and conduct. For example, workers can interrupt their own task and help colleagues with tasks at other work stations when they observe that their coworkers are having difficulties. Understanding when and why workers demonstrate flexible working practices can help manufacturers learn and adjust their decision-making processes (e.g., re-planning the task assignment) accordingly, leading to successful and worker-accepted process improvements. Such human-driven strategies are more likely to succeed compared to those based on idealised performance.

In such industrial environments, sensor technologies are key to understanding worker behaviour and how they actually perform manufacturing tasks. Since in labour-intensive manufacturing systems human workers are the backbone of the production process, to understand how specific manufacturing tasks are actually being performed it is necessary to track worker movements using indoor positioning systems, such as Ultra-wideband

tags, which are becoming increasingly common in production facilities [2]. Real-time position data from these sensors can be used to analyse how tasks are executed by workers in labour-intensive manufacturing processes. Our paper demonstrates this using a real-world dataset [3] of six assembly workers with an initial task assignment.

Real-time position data from workers requires analysis and data mining techniques to derive a meaningful representation of how workers perform tasks. This paper proposes a process mining [4] methodology to derive the sequence of assembly operations carried out by human workers performing a manufacturing task and use this representation to define a process model. By comparing the models derived from the human worker data to their nominal (i.e., fixed) task assignment, deviations are identified that workers are sometimes involved in tasks that are not assigned to them as a result of their adaptive and collaborative behaviour. Through this comparison, it is discovered that workers exhibit more flexibility than their assigned tasks suggest, and are capable of performing a variety of tasks and distributing their capacity among them. Using this flexible task assignment, we provide decision support for improving process efficiency and reducing completion time by making minor adjustments to task capacity allocation. Previous studies [5–7] have highlighted manpower allocation as a critical aspect of decision-making in labour-intensive manufacturing. However, most of these studies rely on analytical models that make strict assumptions about how workers should perform tasks. To evaluate the impact of proposed changes to labour allocation, this paper builds a discrete event simulation using the process model generated from worker localisation data, allowing us to assess completion time under new capacity levels. This paper extends the initial results presented in conference paper [8] by demonstrating the potential of worker position data for providing human-driven decision support via data-driven process models and simulation tools.

The rest of this paper is organised as follows. In Section 2, we review the literature and show how this paper is different from the relevant existing studies. Section 3 describes our methodology and outlines its steps. Section 4 presents our main results. Lastly, Section 5 concludes the paper while pointing to several future research directions.

2. Literature Review

Real-time localisation of objects in manufacturing environments, such as people, machines, materials, and products is essential to achieve the objectives of smart manufacturing and Industry 4.0 revolution which depend on the integration of knowledge about the production environment continuously [9,10]. The specific real-time localisation system (RTLS) technologies that enable this in manufacturing are the indoor positioning systems (IPS). Based on the communication technology used, these systems can be grouped into Wi-Fi, Bluetooth, radio frequency identification (RFID), VLC, and ultra-wide band (UWB)-type technologies [11]. The position data supplied by these technologies have the potential to be used for monitoring and control of manufacturing processes in order to improve efficiency, detect faults or anomalies [12]. For example, tagging and tracking unfinished products provide information, such as tact time, which enables efficiency monitoring and production control [13].

Table 1 summarises the previous studies that made use of indoor localisation sensor data for decision support in manufacturing.

Table 1. Comparison to previous studies using indoor localisation sensor data for decision support in manufacturing.

Study	Sensor Technology	Tracked Objects	Purpose	Technique of Analysis
[14]	RFID	products	job scheduling in machines	simulation
[15]	RFID	products	re-design of workflow	simulation
[16]	RFID	products	production scheduling	simulation
[17]	RFID	products	production scheduling	multi objective optimisation
[18]	Fixture	workstations	fault detection	quadratic programming
[19]	UWB	products	lead time prediction	semantic enriching
[20]	Lidar and UWB	workers	safety monitoring	supervised machine learning
[21]	UWB	products	bottleneck identification	process mining and value stream mapping
[22]	UWB	workers	safety monitoring	supervised machine learning
[23]	UWB	products	bottleneck identification	value stream mapping
Our paper	UWB	workers	worker capacity allocation	process mining and simulation

The literature summary in Table 1 suggests that the majority of the studies focus on tracking products to monitor efficiency and provide decision support for process improvement. We note the works by Nwakanma et al. [20] and Islam et al. [22] who also track workers inside the manufacturing environments, as in our methodology. In these papers, authors focus on monitoring safety and detecting dangerous situations. Their approach also involves integrating data on breathing patterns of workers. In contrast, our paper uses worker position data in conjunction with the factory plan to assess and improve process efficiency.

In Table 1, the work of [14–16] should be acknowledged, who, like us, utilise simulations for decision support. However, a key distinction between our approach and theirs pertains to the modelling of worker behaviour during manufacturing tasks in the simulation. Our paper adopts a data-driven approach that leverages worker position data and process mining techniques. In contrast, these studies rely on manual inputs, such as expert opinions.

Process mining algorithms automatically discover, monitor, and improve actual as-is processes (i.e., not assumed processes) from event logs, with applications in process analysis, improvement, compliance, and case management. In manufacturing, it is the third largest application domain and is used to identify process deviations and detect bottlenecks, typically relying on event logs generated by IT tracking systems of activities performed on specific products [4]. For instance, [24] uses an order status tracking system to create event logs for ordered products and the activities performed on them, while [25] relies on code readers placed on machines to automatically scan product IDs and record timestamps. [26] also includes product information. Perhaps, the most relevant study to our paper is by Tran et al. [21], which we note in Table 1, who also uses position data from localisation sensors to extract event logs for process mining. However, in all these studies, the event logs are based on traces of products, whereas our methodology is based on tracking workers.

Product positions can provide a more direct understanding of activities compared to worker positions, but detecting manufacturing activities from worker positions requires activity recognition. Combining process mining with activity recognition is crucial for manufacturing, as noted in [27]. Our methodology uses the facility layout to detect activities related to manufacturing tasks using worker position data, similar to [19,21]. A relevant study in the context of tracking worker movement for process mining to discover process models in manufacturing is [28]. In [28], a video-based system tracks hand movements of workers in a manual assembly line to identify activities and create event logs. Similarly, our paper uses process mining to discover process models and identify deviations in worker behaviour. Additionally, our paper uses the data-driven process model to provide decision support on capacity allocation through discrete event simulation.

Discrete event simulation (DES) models provide decision support for complex systems by allowing what-if analyses and exploration of process redesign alternatives [29]. Traditionally, DES models are manually built using expert knowledge, but data-driven models offer advantages [30]. Process mining, which leverages event logs from actual process exe-

cutions, can be particularly valuable in developing data-driven DES models [31,32]. In this approach, process models and performance information derived from process mining can serve as the input for automating the design of simulation models. While previous studies have focused on healthcare [33,34], applications in manufacturing are emerging [30].

The focus of this paper is to provide human-driven decision support to improve process efficiency. However, we would like to also touch upon the link between the human-centric decision-making and sustainability [35] and resiliency-related objectives [36]. We note that in manufacturing systems in particular these aspects become important to aim through human-centricity in human–robot collaboration systems [37].

3. Materials and Methods

3.1. Data

This paper uses the 2D worker position data provided in [3] which are collected through indoor localisation sensors during a three-hours long work shift. The dataset is generated using both UWB tags worn by workers and motion capture technologies to track workers. Here, we use their UWB data because the motion capture system provides fewer position samples (since it was used for only about two hours). Specifically, these data provide the position measurements taken from each worker in x and y coordinates. These coordinates are measured in meters by considering the anchor position (0, 0) as the reference point on the shop floor. Figure 1 presents a scatter plot of the 2D position data coming from one of the workers for visualisation purposes.

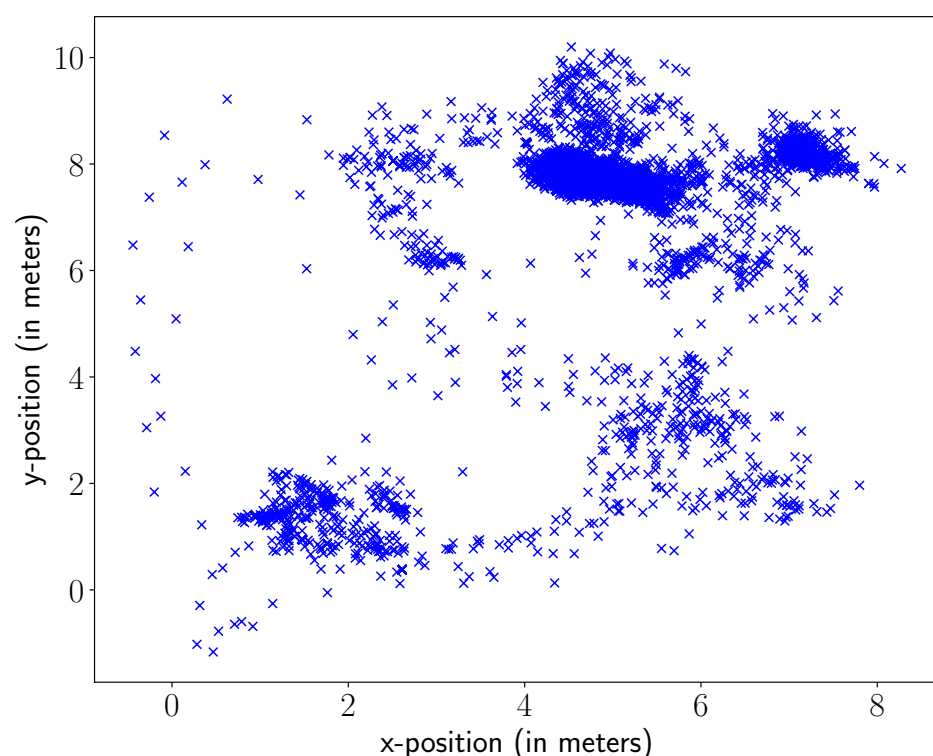


Figure 1. Position Data of Worker 1.

The manufacturing process that workers are involved is an assembly line of tricycles. This line consists of six different manufacturing tasks ($j = 1, 2, \dots, 6$) that are to be performed by six workers ($i = 1, 2, \dots, 6$). Figure 2 presents the tasks and the precedence relations between them (e.g., the axle cannot be assembled without having the lower frame).

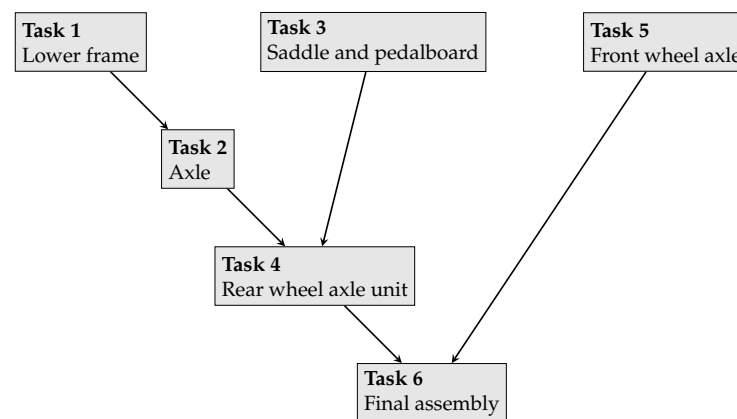


Figure 2. The manufacturing tasks of assembling tricycles.

In this assembly line, each worker is fixed to a single task. So, the fixed task assignment can be described with the following matrix given in Equation (1).

$$A^{\text{fixed}} = [a_{ij}^{\text{fixed}}] = \begin{matrix} & \begin{matrix} \text{Task 1} & \text{Task 2} & \text{Task 3} & \text{Task 4} & \text{Task 5} & \text{Task 6} \end{matrix} \\ \begin{matrix} \text{Worker 1} \\ \text{Worker 2} \\ \text{Worker 3} \\ \text{Worker 4} \\ \text{Worker 5} \\ \text{Worker 6} \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{matrix} \quad (1)$$

The manufacturing floor is divided into regions and each task has its own designated region. The locations of these regions are shown in Figure 3.

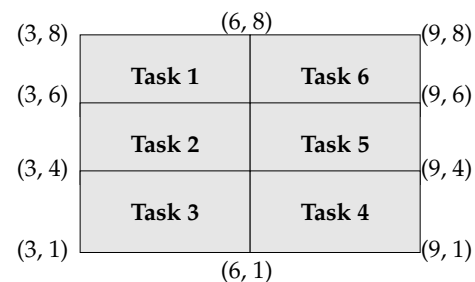


Figure 3. The locations of regions designated for tasks. (Region for Task 1 is defined as $\{(x, y) | 3 \leq x < 6, 6 < y \leq 8\}$, for Task 2 as $\{(x, y) | 3 \leq x < 6, 4 < y \leq 6\}$, for Task 3 as $\{(x, y) | 3 \leq x < 6, 1 < y \leq 4\}$, for Task 4 as $\{(x, y) | 6 \leq x < 9, 1 < y \leq 4\}$, for Task 5 as $\{(x, y) | 6 \leq x < 9, 4 < y \leq 6\}$, and for Task 6 as $\{(x, y) | 6 \leq x < 9, 6 < y \leq 8\}$.)

3.2. Methodology

Given this floor plan and the worker position data collected through UWB sensors, we develop our methodology to extract process models and incorporate them in a discrete event simulation framework to provide decision support. This methodology is summarised in Figure 4.

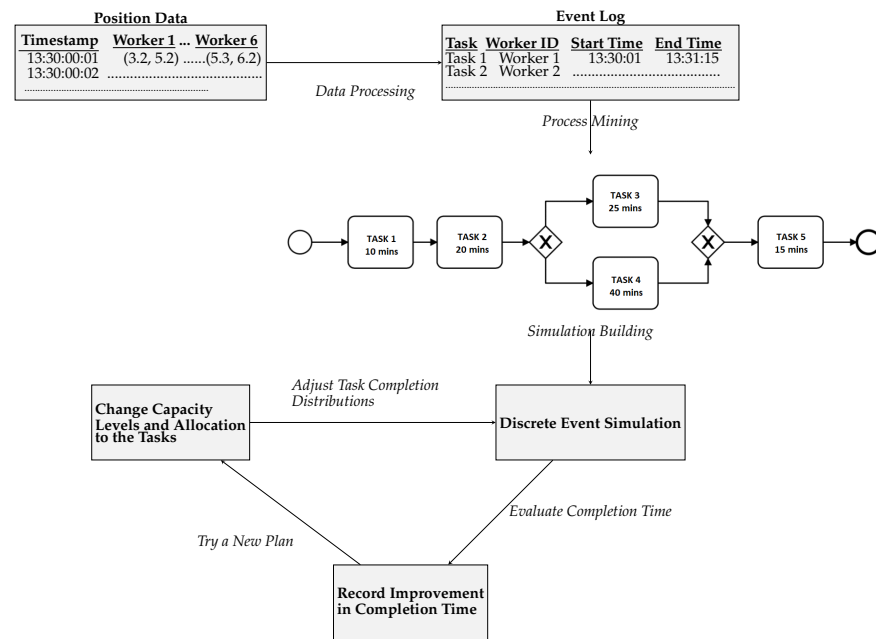


Figure 4. Methodology: using worker position data for human-driven process model discovery and decision support with discrete event simulation.

Our methodology involves a number of steps. These are described in detail below.

3.2.1. Data Processing and The Extraction of an Event Log

UWB tags are measured at an interval of 100 ms. To align the data collected from UWB tags attached to workers, the data interval of 100 ms is converted to seconds as no significant changes in worker positions are expected within a second. The average position data obtained within the same second is used as a smoothing function. If data are missing for a worker, for example, when signals are obstructed, this is substituted with the worker's last known position.

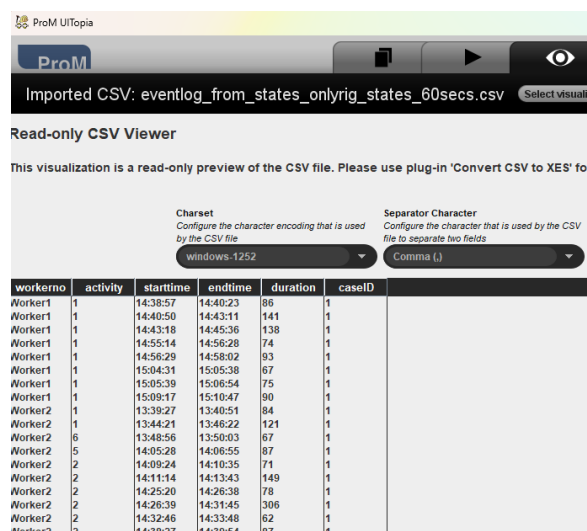
An event log is generated from the synchronised second-by-second position dataset. To assign events relating to the execution of a certain manufacturing task by workers, we examine the proximity of workers to task regions and the duration of their stay there. To identify whether a worker is actually performing a specific task based on their position, the amount of time they spend in a particular region should be considered. For example, if a worker is in one region at second t and then in another region the next second, it could simply be due to walking past, particularly if their time in the initial region was brief. To address this, a parameter $\tau > 0$ is used to determine if a worker is performing a task i at some time t if their position was within task region j at any time t' within the period $t, t-1, \dots, t-\tau$. Here, by fixing τ to 60 s, we determine the log of events that show the execution of a certain task by a certain worker with a start and end time. It must be noted that τ is not set to a lower value here to avoid considering short visits to the regions of other tasks (e.g., transfer of products and material) as workers being involved in the execution of these tasks.

3.2.2. Process Mining to Derive Process Models

In this step, the event log is used to discover a process model of the manufacturing process with a suitable process mining algorithm. Multiple process model discovery algorithms, including Alpha Miner, Heuristic Miner, and Inductive Miner, are available for discovering process models from event logs. Similar to [38], this paper applies the Inductive Miner, since it is an improvement on the Alpha and Heuristic algorithms. The Inductive Miner is a divide and conquer type polynomial-time algorithm that returns a process tree in

which each activity is seen once. We refer the reader to [39] for the theoretical foundations of this algorithm.

This paper uses the implementation of this algorithm available in ProM 6.11 (<https://www.promtools.org/>, accessed on 20 March 2023). The event log is provided as a comma-separated file to ProM, as illustrated in Figure 5. Before the Inductive Miner is applied, the converter tool in ProM is used to convert this to XES format.



workerid	activity	starttime	endtime	duration	caseid
Worker1	1	14:38:57	14:40:23	86	1
Worker1	1	14:40:50	14:43:11	141	1
Worker1	1	14:43:18	14:45:36	138	1
Worker1	1	14:55:14	14:56:28	74	1
Worker1	1	14:56:29	14:58:02	93	1
Worker1	1	15:04:31	15:05:38	67	1
Worker1	1	15:05:39	15:06:54	75	1
Worker1	1	15:09:17	15:10:47	90	1
Worker2	1	13:39:27	13:40:51	84	1
Worker2	1	13:44:21	13:46:22	121	1
Worker2	6	13:48:56	13:50:03	67	1
Worker2	5	14:05:28	14:06:55	87	1
Worker2	2	14:09:24	14:10:35	71	1
Worker2	2	14:11:14	14:13:43	149	1
Worker2	2	14:25:20	14:26:38	78	1
Worker2	2	14:26:39	14:31:45	306	1
Worker2	2	14:32:46	14:33:48	62	1
Worker2	2	14:38:27	14:39:54	87	1

Figure 5. Inspecting the Event Log provided in CSV format in ProM.

3.2.3. Building the Discrete Event Simulation

This paper builds a simulation model based on the process model obtained from the event log. The process model obtained from the Inductive Miner algorithm uses the business process model notation (BPMN) [40]. This notation can consist of objects that mark specific events, including the start and end of the process, which are denoted with small circles; activities, which are represented with rectangles; connections and flow objects showing the sequence relations, which are represented using arrows; and also gateways, which are shown with smaller diamond shapes. Gateways are objects that determine forking and merging of paths and they can model conditional events. For example, parallel gateways (e.g., splitting or joining) are used to model the completion of parallel processes that can only finish when each of its processes is performed.

To build a discrete event simulation framework that represents the state of the process and events that transform the state under the transition probabilities, this BPMN notation should be translated. Our approach models the start, end and intermediate events, and also activities as part of the state description and uses flow sequences and their frequencies to model the transition probabilities. We call the model of the entire process as *Process^{main}* which consists of several subprocesses because of the parallel gateways in the process model. The main process always starts from the same state, the start event marker in the process model, and will end the first time the state transitions to the end state. To complete the simulation framework the transition times should also be modelled, namely, how long it will take until the current state s will change to another state s' due to the completion of a task execution event. To model the transition times, we use the mean sojourn time information as supplied by the Inductive Miner and use stochastic distributions whose means are fitted to these. The details of how this simulation framework is built with the data of our case study are given in the Appendix A. We implement this simulation in C++ language, using Visual Studio.

The performance measure evaluated with this simulation is the average completion time of the manufacturing process of assembling 6 tricycles. In this evaluation, a large number of samples ($K = 50,000$) (to reduce the standard error) are taken, from multiple simulation runs under random seeding, and then the averages of their completion times are

calculated. The completion time of the process from each sample k is the first passage time until $Process^{\text{main}}$ reaches the end state. We denote the completion time with $C(Process^{\text{main}})$. So, based on the simulation samples, this is measured with

$$C(Process^{\text{main}}) = \frac{1}{K} \sum_{k=1}^K C(Process_k^{\text{main}}), \quad (2)$$

where $C(Process_k^{\text{main}})$ is the completion time measured in the k th sample.

3.2.4. Using the Simulation for Decision Support on Capacity Allocation

To facilitate the use of the simulation model for the allocation of workers to tasks, it is necessary to establish the relationship between the labour resource (e.g., person-minutes) and output for each stage of the process. This paper achieves this by fitting the total capacity allocated to the tasks, as observed from the event log, to the mean sojourn times of tasks, as provided in the process model derived with the Inductive Miner. Then, considering that with more capacity allocated to a task the faster its execution will be, we find the corresponding new mean task sojourn times to be used in the simulation model under a new capacity allocation. A linear model is used for this purpose.

The first step is the derivation of the task assignment matrix of the data ($A^{\text{data}} = [a_{ij}^{\text{data}}]$) based on the event log. This is obtained from the total time that workers spend for executing each of the manufacturing tasks. Then, from this the total capacity allocated to each task in the process observed with the data $R_j^{\text{data}}, \forall j = 1, 2, \dots, 6$ is found via

$$R_j = \sum_i a_{ij}^{\text{data}}. \quad (3)$$

Letting μ_j^{mined} denote the mean sojourn time of task j as obtained from the Inductive Miner, and μ_j^{new} the new (would be) mean sojourn time of task j under R_j^{new} , a new total capacity level for task j , we have

$$\mu_j^{\text{new}} = \frac{\mu_j^{\text{mined}} R_j^{\text{data}}}{R_j^{\text{new}}}. \quad (4)$$

So, our decision support involves changing the task allocation under a new task allocation matrix A^{new} and having new capacity levels for the tasks $R_j^{\text{new}}, \forall j = 1, 2, \dots, 6$, updating the mean sojourn times of the tasks (the transition times in the simulation) in the simulation and using the simulation to evaluate the average completion time under the new allocation. The aim of this procedure is to find suitable allocations that will reduce the completion time.

4. Results

4.1. Observing Deviations from the Fixed Task Assignment and Finding the Assignment from the Data

This section explores the event log to observe and identify deviations from the fixed task assignment, as given in Equation (1). For this, using the dotted chart visualisation tool in ProM first, Figure 6 is presented. The first observation to make is that the supposed task assignment is not being strictly followed, as there are deviations. Specifically, it is observed that worker 2 joins worker 1 for task 1 at the beginning of the shift, and worker 4 sometimes takes on the responsibility of task 5. Additionally, worker 6 assists with tasks 2 and 3. These deviations may be due to the fact that the tasks assigned to workers 2, 4, and 6 are dependent on outputs from earlier assembly stages and cannot be started independently. Additionally, it can be seen that some workers perform task continuously throughout the shift, such as worker 5. However, we typically see that the execution of tasks are clustered and there are breaks in between. These breaks may be due to the completion

of the current product and switching to another, since a total of six tricycles are completed during the shift.

Secondly, from the event log, the task assignment matrix of the data ($A^{\text{data}} = [a_{ij}^{\text{data}}]$) based on how much time each worker spends on performing each of the manufacturing tasks is calculated. This matrix is presented in Equation (5), which again demonstrates how workers deviate from the fixed task assignment and how much of their working capacity is put into several tasks. According to this matrix, it is found that $R_j^{\text{data}}, j = 1, 2, \dots, 6$, the total capacity put to each task according to these data as (1.28, 1.21, 1.41, 0.55, 1.32, 0.24), which indicates flexible worker behaviour. This shows that tasks 1, 2, 3, and 5 are given similar capacity levels, while the capacity allocated to tasks 4 and 6 is much lower. Considering what these tasks involve (see Figure 2), we see that tasks 4 and 6 are characteristically similar and somewhat different from other tasks such that they involve some final assembly steps of a number of sub-assemblies. According to A^{data} , it seems that workers collectively put less effort into these assembly steps.

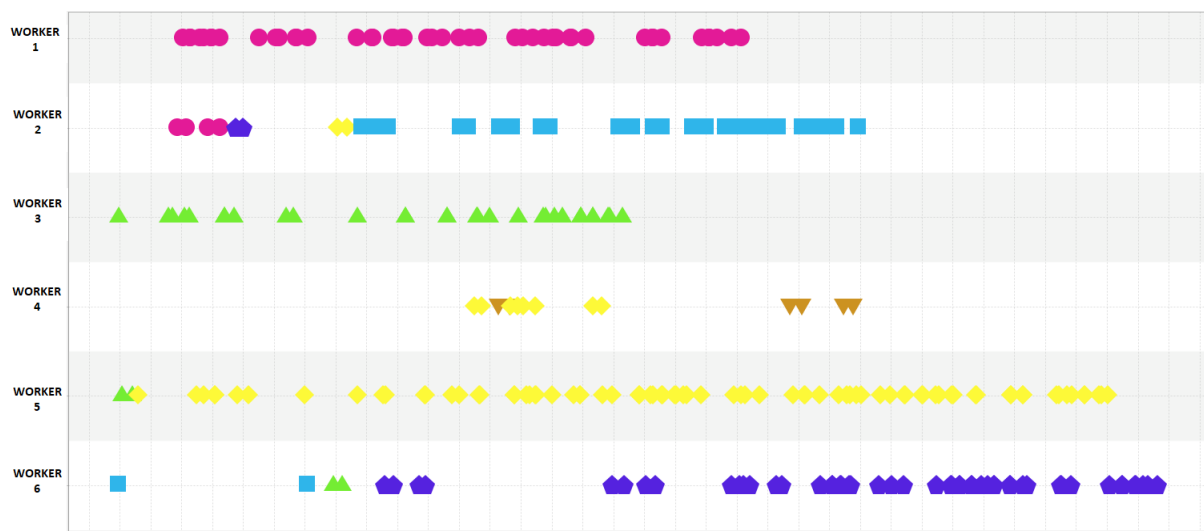


Figure 6. Dot chart of manufacturing tasks performed by 6 workers. (The points mark the start time of the activities (Task 1: Circle (Pink), Task 2: Square (Blue), Task 3: Up-pointing Triangle (Green), Task 4: Down-pointing Triangle (Orange), Task 5: Diamond (Yellow), Task 6: Pentagon (Purple)).

$$A^{\text{data}} = [a_{ij}^{\text{data}}] = \begin{matrix} & \begin{matrix} \text{Task 1} & \text{Task 2} & \text{Task 3} & \text{Task 4} & \text{Task 5} & \text{Task 6} \end{matrix} \\ \begin{matrix} \text{Worker 1} \\ \text{Worker 2} \\ \text{Worker 3} \\ \text{Worker 4} \\ \text{Worker 5} \\ \text{Worker 6} \end{matrix} & \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0.28 & 0.31 & 0 & 0 & 0.24 & 0.18 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.55 & 0.45 & 0 \\ 0 & 0 & 0.37 & 0 & 0.63 & 0 \\ 0 & 0.90 & 0.04 & 0 & 0 & 0.06 \end{pmatrix} \end{matrix} \quad (5)$$

4.2. Data-Driven Process Model

It is possible to obtain a variety of different process models using the Inductive Miner. This is made possible by varying the two filtering parameters of the algorithm. These are the filters on the activities, namely, the manufacturing tasks, and the paths among activities as observed in the event log. Varying the level of filtering applied affects the quality of the obtained process models with respect to their fitness and complexity. Model fitness, which measures the ability of the model to allow the behaviour observed in the data (i.e., the ability to replay the sequences observed in the logs), and simplicity are among of the most commonly considered process model quality metrics along with the quality mea-

asures of precision (i.e., not allowing the behaviour not seen in the logs) and generalisation. We refer the reader to [4] for more information on these four quality measures.

Specifically, with increased levels of filtering, simpler models can be obtained, however, their fitness may be lower. Our paper applies 10% filtering on the paths and obtain a model with 96% fitness (Figure 7). Here, we do not use the filtering on the activities so that we can see the involvement of each manufacturing task in the process. This is preferred because this model is to be used for building a discrete event simulation which will be used for the capacity allocation decisions to each of the manufacturing tasks. It can be observed that the model includes processes that are executed in parallel and forwarded to the next activity when all the involved processes are completed, as indicated with the splitting and joining parallel gateways. This dependency can be as a result of the precedence relations between the manufacturing tasks, as illustrated in Figure 2.

The model in Figure 7 is used when fitting our discrete event simulation. The mean sojourn times of the tasks μ_j^{mined} are (3.07, 4.18, 2.20, 22.15, 3.06, 3.59) in minutes. Although this suggests that the execution of task 4 takes the longest, it must be noted that this does not mean that the total time spent for task 4 until the process ends will be the longest. This is because the total time spent on tasks also depends on how many times they will have to be executed, information which is captured in the flow frequencies in the process model. It can be seen in Figure 7 that the number of times that task 4 will be executed is much lower than other tasks. For this reason, the flow frequencies must be accounted for, in our discrete event simulation. This paper investigates the total time spent in tasks using our simulation.

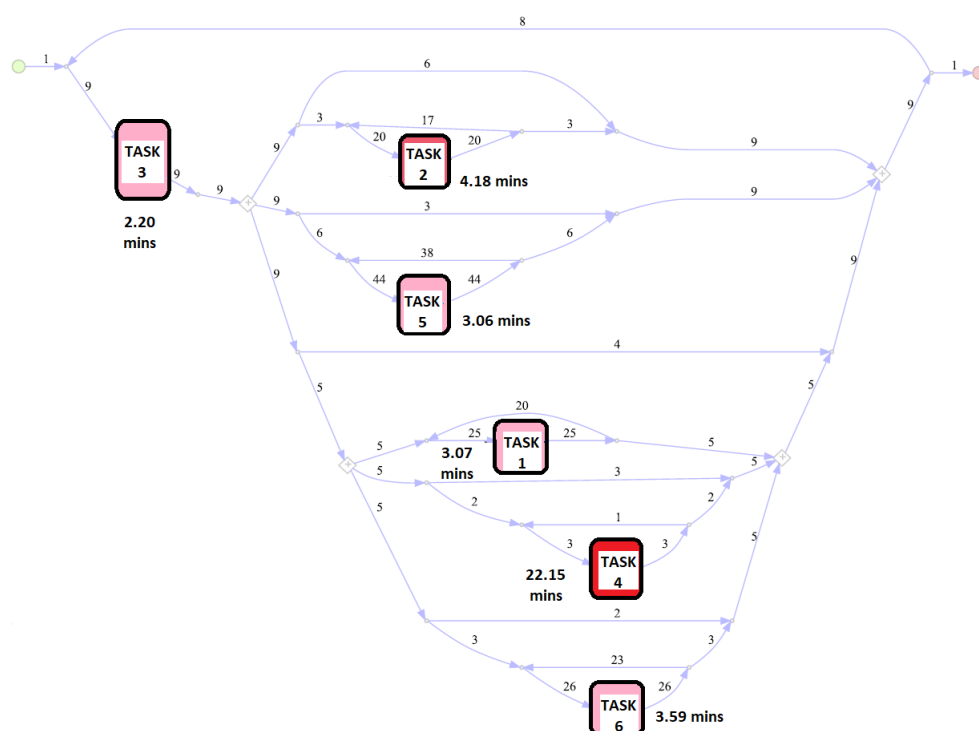


Figure 7. The process model with mean sojourn times (in minutes) of tasks and flow frequencies (mean sojourn times are (3.07, 4.18, 2.20, 22.15, 3.06, 3.59)).

4.3. Decision Support on Capacity Allocation with Simulation

This section conducts a series of investigations for the purposes of decision support regarding the capacity allocation. In doing so, we consider new task assignments and capacity allocation levels and use our discrete event simulation to evaluate the average completion time. In every evaluation, $K = 50,000$ samples are taken from the simulation, as described in Section 3.2.3. This gives a high-precision estimate on the average completion time with standard errors falling below 1.4 min the most of the time. Nevertheless, the

errors are provided in our results. The time unit used in the results presented in this section is minutes.

4.3.1. Comparing the Performance under the Worker-Driven Flexible Allocation to the Fixed Assignment

This section firstly evaluates the completion time according to the flexible allocation observed in the worker data (as manifested in A^{data} and R_j^{data}). Secondly, the completion time according to A^{fixed} , namely, under $R_j^{\text{fixed}} = 1, \forall j$ is evaluated. In other words, the second evaluation measures the completion time if the workers stuck to the fixed assignment prescribed to them and did not become involve in any other tasks besides the ones that were assigned to them. Comparing the completion time under these two allocation schemes is useful to understand if the flexible worker behaviour as observed in the data (and the practice of assisting with the other manufacturing tasks when necessary) resulted in an improved performance (i.e., completion of the day's workload earlier).

The average completion time under the fixed allocation gives 328.6 ± 1.4 , but this reduces to 309.8 ± 1.4 using the worker-driven flexible allocation. Considering the error margins, the difference in completion times is significant and indicates an improvement around 5.7% in the average completion time under the flexible allocation compared to the fixed allocation. This shows that indeed the workers' flexible behaviour benefits the overall efficiency of the entire manufacturing process and is a more appropriate approach than the strict allocation of fixed assignment.

Figure 8 presents the box and whisker plots of the completion times under the fixed and flexible allocations to illustrate the distributions of the completion times in more detail. We find that the medians of completion times are 211 and 222 min under the flexible and fixed allocations, respectively. Note that these values are close to the original total duration of the work shift observed in the data, which was around 180 min (3 h).

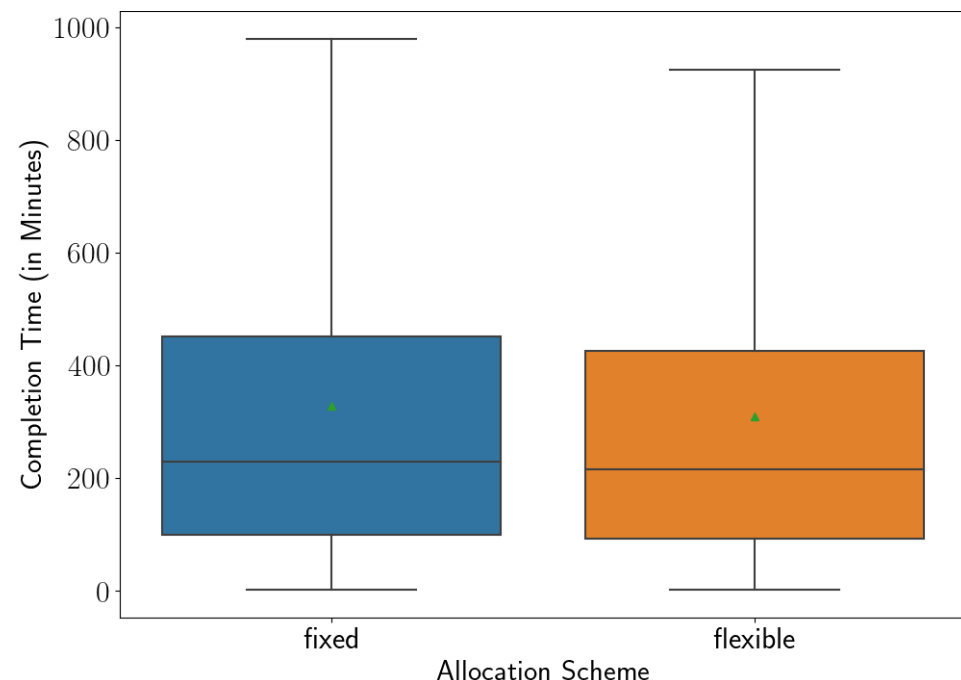


Figure 8. The completion time under the fixed and flexible allocations (green triangles mark the means. With 95% Confidence, the means of completion times under the fixed and flexible allocations are in the intervals (328.5, 328.7) and (309.7, 309.9), respectively.)

The total capacity allocated to each task in the worker-driven flexible allocation $R^{\text{data}} = (1.28, 1.21, 1.41, 0.55, 1.32, 0.24)$ is significantly different than the fixed allocation $R^{\text{fixed}} = (1, 1, 1, 1, 1, 1)$. For example, more capacity is allotted to the tasks 1–3 and 5 under

the flexible allocation, which, at the same time, results in a lower capacity for the tasks 4 and 6 compared to the fixed allocation. Because of this, the time spent for performing the specific manufacturing tasks in completing a day's work would be different as well. Figure 9 presents these task-specific times. It can be observed that how the increased capacity for the tasks 1–3 and 5 in the flexible allocation decreases the time spent in completing these tasks, while it increases the time required for the tasks 4 and 6. In addition, another important difference is noted in the times spent for tasks under these two capacity allocation schemes, which is about the variance observed in the times across all six tasks. We calculate the standard deviation of the average of the total times across tasks as 59.9 and 38.3 in the fixed and flexible allocations, respectively. This reveals that the times spent in tasks are more balanced under the flexible allocation. It can be argued that this balancing advantage over the fixed assignment could be a contributing factor to the lower completion time achieved with the flexible allocation, since the completion of the main manufacturing process depends on parallel subprocesses. More specifically, the completion time of a parallel process involving several tasks highly depends on its bottleneck (i.e., slowest) task since the parallel process can finish only when all of the tasks are finished and a balanced timing of the tasks can help to reduce the time required for this completion.

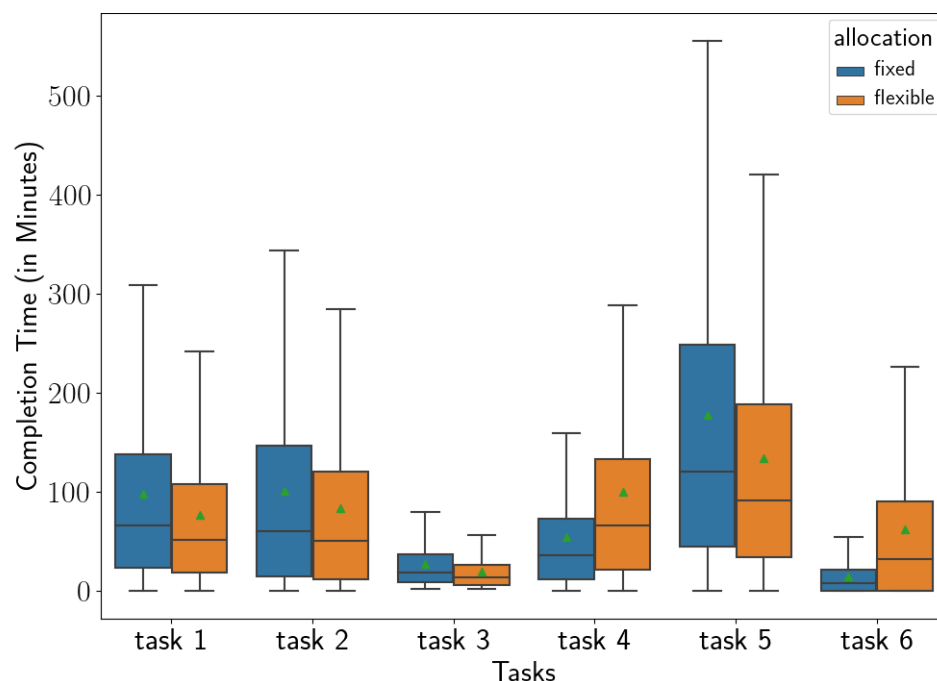


Figure 9. The total time spent in manufacturing tasks in the fixed versus flexible allocations (green triangles mark the means. $R^{\text{fixed}} = (1, 1, 1, 1, 1, 1)$ and $R^{\text{data}} = (1.28, 1.21, 1.41, 0.55, 1.32, 0.24)$).

4.3.2. How to Reduce Completion Time through Small Adjustments Based on the Worker-Driven Flexible Allocation?

The aim of this section is to provide insights to workers based on their natural work practices, as demonstrated in the flexible allocation R^{data} , to ease their workload and finish their shifts earlier. We achieve this by considering small adjustments on the flexible capacity allocation R^{data} , evaluating the completion time under the resulting new allocation scheme, and investigate whether it is possible to reduce the completion time of their work through this re-allocation of their capacity. In these evaluations, capacity adjustments that reduce the capacity from some task j by some percentage and transfer this to some other task j' are considered. The completion times under the evaluated schemes that shift 5%, 10%, 20%, and 50% capacity from one task to another are given in Table 2. Larger capacity shifts are not shown in Table 2 as no improvements in the completion time are found.

Given that the original un-adjusted scheme gives an average completion time of 309.8 ± 1.4 , we see that there are adjustments that can reduce the completion time significantly. The best adjustments that lead to the highest reductions are identified in bold in Table 2. These adjustments, which suggest shifting capacity from task 1 to task 4, from task 3 to task 4, from task 3 to task 5, and from task 3 to task 6, are able to achieve a reduction between 3 and 5%. Considering the total time spent in each task (see Figure 9), it is observed that each of these moves shifts capacity from a task taking shorter time to a task taking a relatively longer time. For example, it is known that task 3 is the task taking the shortest amount of time in the current assignment and this investigation finds several re-allocations to shift capacity from this task to other tasks that take longer, such as the tasks 4, 5, and 6. In other words, these capacity shifts are in favour of balancing the time required to complete each one of the manufacturing tasks. The reason that they are able to reduce the completion time could be due to this balancing effect since the process includes parallel tasks and they are dependent on each other. When the amount of capacity shifted is considered in these adjustments, it is found that the best reductions are possible through smaller but considerable shifts of 20%, whereas higher shifts (50%) start to lose their advantage in reducing the average completion time. Note that large capacity shifts mean large reductions in capacity from the tasks and this may cause delays in the process. For example, even when the task that the capacity is shifted to is a bottleneck and needs capacity, a large reduction from the capacity of another task to fix this may simply shift the place of the bottleneck without achieving significant reduction in the overall completion time. In the last row of Table 2, we investigate a combination of the two of the best small adjustments, shifting 20% capacity of task 1 to task 4 and 20% capacity of task 3 to task 6. With this, the reduction in completion time reaches above 7%, making it possible to finish their work around 20 min earlier. This shows that these small adjustments have potential to make considerable reductions in the completion time when they are based on the workers' behaviour as demonstrated in their movement data.

Table 2. The average completion times (in minutes) under shifting capacity between tasks based on the flexible allocation.

Adjustment	Shift 5%	Shift 10%	Shift 20%	Shift 50%
From Task 1 to 2	308.4 ± 1.4	307.6 ± 1.4	308.1 ± 1.4	332.7 ± 1.5
From Task 1 to 3	313.4 ± 1.4	314.4 ± 1.4	317.8 ± 1.4	347.2 ± 1.5
From Task 1 to 4	302.2 ± 1.3	299.5 ± 1.3	298.3 ± 1.3	326.7 ± 1.4
From Task 1 to 5	307.0 ± 1.4	305.0 ± 1.3	303.3 ± 1.3	323.9 ± 1.4
From Task 1 to 6	303.8 ± 1.3	301.1 ± 1.3	301.6 ± 1.3	333.2 ± 1.4
From Task 2 to 1	311.2 ± 1.4	313.7 ± 1.4	320.8 ± 1.4	370.2 ± 1.7
From Task 2 to 3	311.3 ± 1.4	314.1 ± 1.4	321.6 ± 1.4	370.4 ± 1.6
From Task 2 to 4	307.3 ± 1.3	306.3 ± 1.3	309.0 ± 1.4	354.0 ± 1.6
From Task 2 to 5	307.8 ± 1.4	307.8 ± 1.4	310.7 ± 1.4	352.0 ± 1.6
From Task 2 to 6	303.5 ± 1.3	302.7 ± 1.3	307.0 ± 1.3	356.7 ± 1.6
From Task 3 to 1	308.4 ± 1.4	308.3 ± 1.4	308.9 ± 1.4	320.0 ± 1.4
From Task 3 to 2	308.8 ± 1.3	307.1 ± 1.3	305.1 ± 1.3	310.8 ± 1.4
From Task 3 to 4	302.5 ± 1.3	298.6 ± 1.3	294.5 ± 1.3	300.6 ± 1.3
From Task 3 to 5	306.2 ± 1.3	303.1 ± 1.3	298.7 ± 1.3	299.1 ± 1.3
From Task 3 to 6	302.4 ± 1.3	298.7 ± 1.3	296.8 ± 1.3	308.1 ± 1.3
From Task 4 to 1	313.3 ± 1.4	316.6 ± 1.4	325.4 ± 1.4	384.4 ± 1.7
From Task 4 to 2	311.0 ± 1.4	313.6 ± 1.4	321.1 ± 1.4	376.9 ± 1.7
From Task 4 to 3	312.2 ± 1.4	315.7 ± 1.4	324.7 ± 1.4	383.2 ± 1.7
From Task 4 to 5	309.8 ± 1.4	311.9 ± 1.4	318.3 ± 1.4	371.1 ± 1.6
From Task 4 to 6	307.7 ± 1.4	308.5 ± 1.4	314.2 ± 1.4	371.6 ± 1.6
From Task 5 to 1	314.8 ± 1.4	319.3 ± 1.4	331.5 ± 1.5	411.7 ± 1.8
From Task 5 to 2	309.5 ± 1.3	312.5 ± 1.4	322.1 ± 1.4	397.4 ± 1.7
From Task 5 to 3	313.2 ± 1.4	318.1 ± 1.4	330.6 ± 1.4	409.6 ± 1.8
From Task 5 to 4	307.9 ± 1.3	308.8 ± 1.4	316.7 ± 1.4	394.0 ± 1.7
From Task 5 to 6	305.1 ± 1.3	306.2 ± 1.3	315.9 ± 1.4	396.7 ± 1.7
From Task 1 to 4 and From Task 3 to 6			287 ± 1.4	

4.3.3. How to Use Additional Worker Capacity Based on the Worker-Driven Flexible Allocation?

The previous investigation provides insights for which small adjustments to the worker-driven flexible allocation improves the completion time best such that the workers can complete their work earlier, without requiring to hire and engage additional worker capacity. The assembly system we analyse in this paper has, in total, six workers. This section investigates which capacity allocation strategy would be the best if there was one more worker available to assist in performing the tasks. For this, the worker-driven flexible allocation $R^{\text{data}} = (1.28, 1.21, 1.41, 0.55, 1.32, 0.24)$ is considered as the current practice under the capacity of six workers and then allocate the available one unit capacity of the additional worker to improve this.

Firstly, the strategies that concentrate the entire additional capacity on one single manufacturing task are evaluated (Figure 10). This shows that the best strategy is to assign the new worker to task 5 and the second best option would be task 4. With the strategy of assigning the entire additional capacity to task 4, the completion time reduces from 309.8 ± 1.4 to 273.3 ± 1.2 , having a reduction potential around 11.6% in the completion time. As discovered in the previous investigations, this preference for the tasks 4 and 5 seem to be in line with the logic of trying to balance the time spent on the manufacturing tasks. From Figure 9, it is known that these tasks take the largest amount of time according to the current practice R^{data} with six workers and assigning the new worker to these would have a balancing effect on the time spent on each one of the tasks by reducing the time for these tasks.

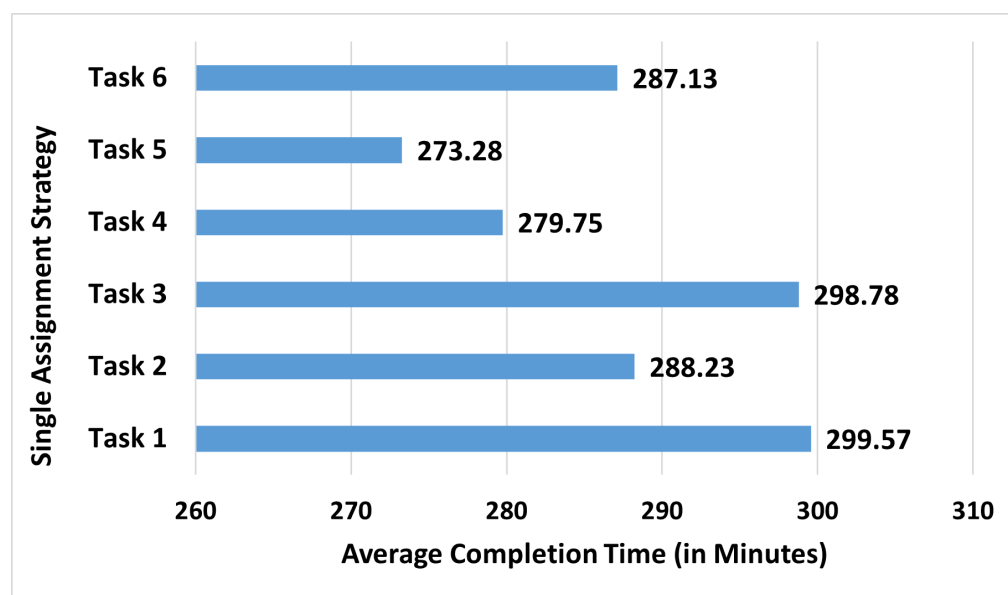


Figure 10. The average completion time (in minutes) after adding the capacity of the additional worker to the worker-driven flexible allocation through single-task assignment (The standard errors vary between 1.2 and 1.3 min)

As demonstrated in the data, it is known that workers can flexibly allocate their capacity and become involved in multiple tasks during their shift. Keeping this in mind, flexible strategies that assigns the new worker to more than one task are investigated next. Figure 11 considers strategies that split the capacity in two in order to support two different manufacturing tasks. It can be observed that the strategy that assigns the half of the new worker's capacity to task 4 and the other half to task 5 is the best. Moreover, it seems that this split strategy reduces the completion time much better than the single-task assignment strategies of assigning the entire capacity to task 4 or task 5, achieving a reduction of around 16.8%. Even more flexible allocations that would assign the new worker to more than two tasks have the potential to reduce the completion time more. However, this can create a

complicated situation for the new worker in terms of scheduling and time management. Moreover, because of these challenges it may not be possible to realise the high-reduction potential of such multi-task assignment strategies are beyond the scope of the current work.

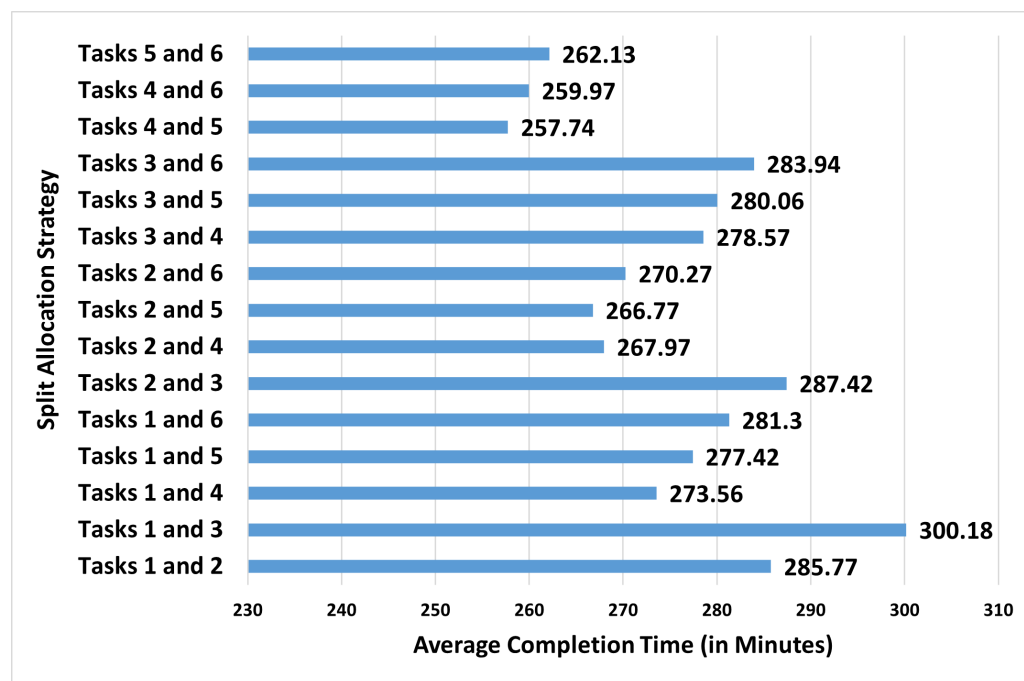


Figure 11. The average completion time (in minutes) after adding the capacity of the additional worker to the worker-driven flexible allocation through two-tasks assignment (The standard errors vary between 1.1 and 1.3 min).

5. Conclusions and Discussion

This paper presents a novel methodology for supporting human-driven decision-making in labour-intensive manufacturing processes by analysing worker position data. This involves integrating process mining and discrete event simulation to identify opportunities to improve capacity allocation decisions. Our main goal is to demonstrate the value of using localisation sensor data for data-driven decision-making in manufacturing, with a focus on improving process efficiency. By showcasing the benefits of this technology, our aim is to help manufacturers make informed decisions about whether to invest in indoor localisation sensors or not.

This methodology is applied to a real-world dataset involving six workers performing six manual manufacturing tasks to assemble tricycles with a fixed assignment in which each worker is responsible for performing one of the tasks. Our first identification is that workers demonstrate a much more flexible behaviour than the fixed task assignment and involve in various tasks. By putting this flexible assignment observed in the data in the centre, this paper provides decision support on how to reduce the completion time of the manufacturing process through small adjustments on the capacity levels that workers allocated to the tasks. In our capacity adjustment investigations, we first show that with small adjustments of shifting capacity between two manufacturing tasks one can reduce the completion time by 7%, without requiring any additional workers. Secondly, we investigate the situation where there is one more worker available to support the manufacturing process and show that this additional capacity can reduce the completion time by more than 11% if the new worker is to be assigned to a single task, while this can reach above 16% if two tasks are assigned.

Our work presented in this paper has certain limitations. Firstly, our approach for detecting specific manufacturing tasks performed by workers may face challenges with accuracy. This is because our approach relies on position data provided by sensors and the fac-

tory layout, which dictates the regions where workers can perform each task. For instance, inaccurate sensor data or workers performing tasks outside their designated zones may result in inaccuracies or failures in detecting activity events. Another limitation of our research is the allocation of worker capacity. While suggesting adjustments to the capacity levels allocated to tasks, we only consider the total available capacity of workers. This may overlook scheduling decisions and inefficiencies that may arise due to complications in assigning workers to multiple manufacturing tasks.

Future research can delve into decision support on dynamic capacity allocation and/or worker assignment strategies. This paper used the entire data coming from a work shift to find a process model representing the process from beginning to end. However, to understand the changing work patterns in time, the data can be split into several segments in time and, instead of deriving a single process model, one can obtain multiple models with process mining, by considering the event logs of each segment separately. In this way, one can build models of the process segmented in time and investigate dynamic strategies that can vary in time. For example, it could be that allocating more capacity to the first task is beneficial during the first hour of the shift but not so in the last hour. Having separate models as these would be useful for finding such insights that might improve the process efficiency better. Furthermore, there might also be some research possibilities for using the worker position data for other types of decision-making in manufacturing rather than worker capacity allocation to tasks. For example, one can investigate the decisions regarding the products, such as how many products to manufacture in a day or when to start the activities for each new product. Given that indoor localisation sensors are becoming commonplace in manufacturing environments due to their affordability and recent advances in this technology, it is of interest to investigate the potential of worker position data to inform and improve the manufacturing processes. Another possibility for future research can involve the implementation of the capacity adjustment strategies provided in this paper as suggestions to the workers in the considered assembly line. After the suggestions have been placed, new position data can be collected from workers to observe how they change their work practices and to see how much improvement has been achieved. The new data can then be also used for deriving process models and building a discrete event simulation for decision support using our methodology to provide further suggestions for process enhancement.

Author Contributions: Conceptualization, A.A., G.V. and J.C.; methodology, A.A.; software, A.A.; validation, A.A.; formal analysis, A.A.; data curation, A.A.; writing—original draft preparation, A.A., H.E.-R. and J.H.; writing—review and editing, G.V., J.Q. and J.C.; funding acquisition, G.V., J.Q., J.C. and A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Engineering and Physical Sciences Research Council, UK [grant number EP/V051113/1 -Productivity and Sustainability Management in the Responsive Factory].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data (UWB position data and the event log in CSV format) and C++ source codes of the simulation can be found in the data repository of Edinburgh Napier University. See <https://doi.org/10.17869/enu.2023.3100035>.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RTLS	Real-time localisation system
IPS	Indoor positioning system
RFID	Radio frequency identification system
UWB	Ultra-wide Band
DES	Discrete event simulation
BPMN	Business process model and notation

Appendix A

Appendix A.1. Details of the Discrete Event Simulation

We build our discrete event simulation according to the process model presented in Figure 7. We use the flow frequencies to model the state transitions. To be used in modelling the transition times, we consider the mean sojourn times of tasks $\mu_j, j = 1, 2, \dots, 6$. When the simulation is to be used under the capacity allocation as observed in the data (A^{data}), we let $\mu_j = \mu_j^{\text{data}}$, and when we investigate the performance under a different capacity allocation, we let $\mu_j = \mu_j^{\text{new}}$ (see Section 3.2.4).

The entire process, which we denote by $Process^{\text{main}}$ contains task 3 and three parallel subprocesses. The first of these involves task 2 ($Process^1$), the second one involves task 5 ($Process^2$) and the last subprocess ($Process^3$) has its own subprocesses. This one either finishes without involving any manufacturing tasks (see the first flow with frequency 4) or involves three parallel subprocesses. The first of these ($Process^{3,a}$) involves task 1, the second one ($Process^{3,b}$) involves task 4, and, lastly, the third one ($Process^{3,c}$) involves task 6.

To model $Process^{\text{main}}$ we use four states ($s \in S = \{0, 1, 2, 3\}$). Here, 0 represents the start state, state 1 models the execution of task 3, state 2 represents the splitting parallel gateway and 3 marks the exit state. Let us first describe the transition probability matrix ($P = [p_{s,s'}]_{s,s' \in S}$) among these four states. Following the flow frequencies, we find that $p_{0,1} = 1, p_{1,2} = 1, p_{2,1} = 8/9, p_{2,3} = 1/9$, and $p_{3,3} = 1$, as it will be always the case that the exit state can only transition to itself. The remaining of the probabilities are all equal to zero. Note that this defines a proper state transition matrix as $\sum_{s' \in S} p_{s,s'} = 1$ for any $s \in S$ and $p_{s,s'} \in [0, 1], \forall s, s'$. Let us now describe the transition times (D_s) from each state s . In our model D_s represents the time it takes to complete the activities involved at that state, state s , which can be above zero only for the states involving manufacturing tasks. Here, these are the states 1 and 2. So, for other states $D_s = 0$. For D_1 , we need to consider the time required to perform task 3. Using the mean sojourn time μ_3 in a normal distribution with σ standard deviation, we have $D_1 \sim N(\mu_3, \sigma)$. For D_2 , we need to consider the time it takes to complete all of the subprocesses, since this one is a parallel gateway. Representing the time it takes to complete $Process^n, n = 1, 2, 3$ with $C(Process^n)$, this time is modelled as $D_2 \sim \text{Max}(C(Process^1), C(Process^2), C(Process^3))$. Each sample k of this simulation is used to find $C(Process_k^{\text{main}})$, which is the first time that the process will enter the exist state, state 3. For this, we start with a counter variable C which is set to zero initially. As the process moves between states, the counter is increased with the transition times each time the process moves into a state which involves manufacturing tasks. The first time that the process steps into the exit state, we stop the process and let $C(Process_k^{\text{main}}) = C$. We must note that since the transitions are probabilistic and also the transition times are modelled using random variables, the completion time is non-deterministic; however, we use the averages of these samples to measure the mean completion time (see Section 3.2.3).

We now describe the subprocesses and how to find their completion times, which we need to find $\text{Max}(C(Process^1), C(Process^2), C(Process^3))$ each time the main process transitions to state 2, the parallel gateway state. The steps we take to model these subprocesses are very similar to those described above for the main process. Let us start with $Process^1$ involving task 2. We use four states ($s \in S = \{0, 1, 2, 3\}$) to model $Process^1$. Here, 0 represents the start state, state 1 represents the intermediate step after the start state, state 2 models the execution of task 2, and, finally, state 3 marks the exit. We find the transition probability matrix as follows. We have $p_{0,1} = 3/9, p_{0,3} = 6/9, p_{1,2} = 1, p_{2,1} = 17/20, p_{2,3} = 3/20, p_{3,3} = 1$ and all the remaining probabilities are zero. Let us now describe the transition time for

the task related state 2. As this state relates to task 2, we have $D_2 \sim N(\mu_2, \sigma)$. Let us now describe how we model $Process^2$ involving task 5. We use four states ($s \in S = \{0, 1, 2, 3\}$) to model $Process^2$. Here, 0 represents the start state, state 1 represents the intermediate step after the start state, state 2 models the execution of task 5, and finally state 3 marks the exit. The transition probability matrix is as follows. We have $p_{0,1} = 6/9$, $p_{0,3} = 3/9$, $p_{1,2} = 1$, $p_{2,1} = 38/44$, $p_{2,3} = 6/44$, $p_{3,3} = 1$ and all the remaining probabilities are zero. Let us now describe the transition time for the task related state 2. As this state relates to task 5, we have $D_2 \sim N(\mu_5, \sigma)$.

To model $Process^3$, we use three states ($s \in S = \{0, 1, 2\}$). Here, 0 represents the start state, state 1 represents the splitting parallel gateway, and state 2 marks the exit. We find the transition probability matrix as follows. We have $p_{0,1} = 5/9$, $p_{0,2} = 4/9$, $p_{1,2} = 1$, $p_{2,2} = 1$ and all the remaining probabilities are zero. Here, state 1 involves manufacturing tasks. However, this is a parallel gateway involving the subprocesses $Process^{3,a}$, $Process^{3,b}$, and $Process^{3,c}$. Thus, we have $D_1 \sim \text{Max}(C(Process^{3,a}), C(Process^{3,b}), C(Process^{3,c}))$ and each time the process steps into state 1, we must find this maximum from the simulation models of the subprocesses $Process^{3,a}$, $Process^{3,b}$, and $Process^{3,c}$. Let us now describe these subprocesses of $Process^3$. $Process^{3,a}$ involving task 1 can be modelled by using three states ($s \in S = \{0, 1, 2\}$). In its transition probability matrix we have $p_{0,1} = 1$, $p_{1,0} = 20/25$, $p_{1,2} = 5/25$, $p_{2,2} = 1$ and all the remaining probabilities are zero. Here, state 1 relates to task 1 and, thus, we have $D_1 \sim N(\mu_1, \sigma)$. We use four states ($s \in S = \{0, 1, 2, 3\}$) to model $Process^{3,b}$ involving task 4. In its transition probability matrix we have $p_{0,1} = 3/5$, $p_{0,3} = 2/5$, $p_{1,2} = 1$, $p_{2,1} = 1/3$, $p_{2,3} = 2/3$, $p_{3,3} = 1$, and all the remaining probabilities are zero. Here, state 2 involves a manufacturing task, task 4. So, we let $D_2 \sim N(\mu_4, \sigma)$. Likewise, we use four states ($s \in S = \{0, 1, 2, 3\}$) to model $Process^{3,c}$ involving task 6. This time in its transition probability matrix we have $p_{0,1} = 2/5$, $p_{0,3} = 3/5$, $p_{1,2} = 1$, $p_{2,1} = 23/26$, $p_{2,3} = 3/26$, $p_{3,3} = 1$ and all the remaining probabilities are zero. In this model, state 2 involves manufacturing task 6, thus we let $D_2 \sim N(\mu_6, \sigma)$. Similarly to the main process, we use a counter variable to find the completion time of $Process^{3,a}$, $Process^{3,b}$ and $Process^{3,c}$ each time we need to sample $C(Process^3)$ and then consider the maximum of this.

So, to find D_2 of the main process each time the process moves into state 2 we find the completion time of $Process^1$, $Process^2$, and $Process^3$ from the sampling of their own simulation models described above and then we consider their maximum.

References

1. Süer, G. A. Optimal Operator Assignment and Cell Loading in Labor-intensive Manufacturing Cells. *Comput. Ind. Eng.* **1996**, *31*, 155–158. [CrossRef]
2. Crețu-Sircu, A.L.; Schiøler, H.; Cederholm, J.P.; Sircu, I.; Schjørring, A.; Larrad, I.R.; Berardinelli, G.; Madsen, O. Evaluation and Comparison of Ultrasonic and UWB Technology for Indoor Localization in an Industrial Environment. *Sensors* **2022**, *22*, 2927. [CrossRef] [PubMed]
3. Delamare, M.; Duval, F.; Boutteau, R. A New Dataset of People Flow in an Industrial Site with UWB and Motion Capture Systems. *Sensors* **2020**, *20*, 4511. [CrossRef] [PubMed]
4. dos Santos Garcia, C.; Meincheim, A.; Faria, Junior E.R.; Dallagassa, M.R.; Sato, D.M.V.; Carvalho, D.R.; Santos, E.A.P.; Scalabrin, E.E. Process Mining Techniques and Applications—A Systematic Mapping Study. *Expert Syst. Appl.* **2019**, *133*, 260–295. [CrossRef]
5. Süer, G.A.; Dagli, C. Intra-cell Manpower Transfers and Cell Loading in Labor-intensive Manufacturing Cells *Comput. Ind. Eng.* **2005**, *48*, 643–655. [CrossRef]
6. Song, B.L.; Wong, W.K.; Fan, J.T.; Chan, S.F. A Recursive Operator Allocation Approach for Assembly Line-balancing Optimization Problem with the Consideration of Operator Efficiency. *Comput. Ind. Eng.* **2006**, *51*, 585–608. [CrossRef]
7. Chang, P.-C.; Lin, Y.-K.; Chan, S. F. Fuzzy-based System Reliability of a Labour-intensive Manufacturing Network with Repair. *Int. J. Prod. Res.* **2015**, *53*, 1980–1995. [CrossRef]
8. Aslan, A.; El-Raoui, H.; Hanson, J.; Vasantha, G.; Quigley, J.; Corney, J. Data-driven Discovery of Manufacturing Processes and Performance from Worker Localisation Data. Flexible Automation and Intelligent Manufacturing FAIM 2023. Lecture Notes in Mechanical Engineering. 2023. Available online: <https://napier-repository.worktribe.com/output/3055612> (accessed on 1 May 2023).

9. Thiede, S.; Sullivan, B.; Damgrave, R.; Lutters, E. Real-time Locating Systems (RTLS) in Future Factories: Technology Review, Morphology and Application Potentials. *Procedia CIRP* **2021**, *104*, 671–676. [\[CrossRef\]](#)
10. Barbieri, L.; Brambilla, M.; Trabattoni, A.; Mervic, S.; Nicoli, M. UWB Localization in a Smart Factory: Augmentation Methods and Experimental Assessment. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–18. [\[CrossRef\]](#)
11. Kunhoth, J.; Karkar, A.; Al-Maadeed, S.; Al-Ali, A. Indoor Positioning and Wayfinding Systems: A Survey. *Hum. Cent. Comput. Inf. Sci.* **2020**, *10*, 18. [\[CrossRef\]](#)
12. Witrisal, K.; Hinteregger, S.; Kulmer, J.; Leitinger, E.; Meissner, P. High-accuracy Positioning for Indoor Applications: RFID, UWB, 5G, and Beyond. *IEEE Int. Conf. RFID* **2016**, 1–7. [\[CrossRef\]](#)
13. Rácz-Szabó, A.; Ruppert, T.; Bántay, L.; Löcklin, A.; Jakab, L.; Abonyi, J. Real-time Locating System in Production Management. *Sensors* **2020**, *20*, 6766. [\[CrossRef\]](#)
14. Thiesse, F.; Fleisch, E. On the Value of Location Information to Lot Scheduling in Complex Manufacturing Processes. *Int. J. Prod. Econ.* **2008**, *112*, 532–547. [\[CrossRef\]](#)
15. Arkan, I.; Van Landeghem, H. Evaluating the Performance of a Discrete Manufacturing Process using RFID: A Case Study. *Robot. Comput.-Integr. Manuf.* **2013**, *29*, 502–512. [\[CrossRef\]](#)
16. Chongwatpol, J.; Sharda, R. RFID-enabled Track and Traceability in Job-shop Scheduling Environment. *Eur. J. Oper. Res.* **2013**, *227*, 453–463. [\[CrossRef\]](#)
17. Guo, Z.X.; Ngai, E.W.T.; Yang, C.; Liang, X. An RFID-based Intelligent Decision Support System Architecture for Production Monitoring and Scheduling in a Distributed Manufacturing Environment. *Int. J. Prod. Econ.* **2015**, *159*, 16–28. [\[CrossRef\]](#)
18. Ruppert, T.; Abonyi, J. Software Sensor for Activity-time Monitoring and Fault Detection in Production Lines. *Sensors* **2018**, *18*, 2346. [\[CrossRef\]](#)
19. Mieth, C. Semantic Enrichment of Spatio-temporal Production Data to Determine Lead Times for Manufacturing Simulation. In Proceedings of the 2019 Winter Simulation Conference, National Harbor, MD, USA, 8–12 December 2019. Available online: <https://www.informs-sim.org/wsc19papers/200.pdf> (accessed on 10 April 2023).
20. Nwakanma, C.I.; Islam, F.B.; Maharani, M.P.; Lee, J.-M.; Kim, D.-S. Detection and Classification of Human Activity for Emergency Response in Smart Factory Shop Floor. *Appl. Sci.* **2021**, *11*, 3662. [\[CrossRef\]](#)
21. Tran, T.-A.; Ruppert, T.; Abonyi, J. Indoor Positioning Systems can Revolutionise Digital Lean. *Appl. Sci.* **2021**, *11*, 5291. [\[CrossRef\]](#)
22. Islam, F.B.; Lee, J.-M.; Kim, D.-S. Smart Factory Floor Safety Monitoring Using UWB Sensor. *IET Sci. Meas.* **2022**, *16*, 412–425. [\[CrossRef\]](#)
23. Sullivan, B.P.; Ghafoorpoor Yazdi, P.; Suresh, A.; Thiede, S. Digital Value Stream Mapping: Application of UWB Real Time Location Systems. *Procedia CIRP* **2022**, *107*, 1186–1191. [\[CrossRef\]](#)
24. Lugaresi, G.; Ciappina, A.D.; Rossi, M.; Matta, A. Exploiting a Combined Process Mining Approach to Enhance the Discovery and Analysis of Support Processes in Manufacturing. *Int. J. Comput. Integr.* **2022**, *133*, 1–21. [\[CrossRef\]](#)
25. Lorenz, R.; Senoner, J.; Sihni, W.; Netland, T. Using Process Mining to Improve Productivity in Make-to-stock Manufacturing. *Int. J. Prod. Res.* **2021**, *59*, 4869–4880. [\[CrossRef\]](#)
26. Choueiri, A.C.; Sato, D.M.V.; Scalabrin, E.E.; Santos, E.A.P. An Extended Model for Remaining Time Prediction in Manufacturing Systems Using Process Mining. *J. Manuf. Syst.* **2020**, *56*, 188–201. [\[CrossRef\]](#)
27. Mannhardt, F.; Bovo, R.; Oliveira, M.F.; Julier, S. A Taxonomy for Combining Activity Recognition and Process Discovery in Industrial Environments. In *Intelligent Data Engineering and Automated Learning—IDEAL 2018. IDEAL 2018. Lecture Notes in Computer Science*; Yin, H., Camacho, D., Novais, P., Tallón-Ballesteros, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; Volume 11315. [\[CrossRef\]](#)
28. Knoch, S.; Ponpathirkoottam, S.; Schwartz, T. Video-to-model: Unsupervised Trace Extraction from Videos for Process Discovery and Conformance Checking in Manual Assembly. In *Business Process Management. BPM 2020. Lecture Notes in Computer Science*; Fahland, D., Ghidini, C., Becker, J., Dumas, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2020; Volume 12168. [\[CrossRef\]](#)
29. Law, A.M.; Kelton, W.D. Simulation Modeling and Analysis. In *Industrial Engineering Series*; McGraw-Hill: New York, NY, USA, 1982.
30. Friederich, F.; Francis, D.P.; Lazarova-Molnar, S.; Mohamed, N. A Framework for Data-driven Digital Twins of Smart Manufacturing Systems. *Comput. Ind.* **2022**, *136*, 103586. [\[CrossRef\]](#)
31. Lazarova-Molnar, S.; Li, X. Deriving Simulation Models from Data: Steps of Simulation Studies Revisited. In Proceedings of the 2019 Winter Simulation Conference (WSC), National Harbor, MD, USA, 8–11 December 2019; pp. 2771–2782. [\[CrossRef\]](#)
32. Rozinat, A.; Mans, R.S.; van der Aalst, W.M.P. Discovering Simulation Models. *Inf. Syst.* **2009**, *34*, 305–327. [\[CrossRef\]](#)
33. Tamburis, O. Bridging the Gap Between Process Mining and DES Modeling in the Healthcare Domain. In Proceedings of the 2019 E-Health and Bioengineering Conference (EHB), Iasi, Romania, 21–23 November 2019; pp. 1–4. [\[CrossRef\]](#)
34. Tamburis, O.; Esposito, C. Process Mining as Support to Simulation Modeling: A Hospital-based Case Study. *Simul. Model. Pract.* **2020**, *104*, 102149. [\[CrossRef\]](#)
35. Goel, P.; Mehta, S.; Kumar, R.; Castaño, F. Sustainable Green Human Resource Management Practices in Educational Institutions: An Interpretive Structural Modelling and Analytic Hierarchy Process Approach. *Sustainability* **2022**, *14*, 12853. [\[CrossRef\]](#)
36. Wan, P.K.; Leirimo, T.L. Human-centric Zero-defect Manufacturing: State-of-the-art Review, Perspectives, and Challenges. *Comput. Ind.* **2023**, *144*, 103792. [\[CrossRef\]](#)
37. Wang, L. A Futuristic Perspective on Human-centric Assembly. *J. Manuf. Syst.* **2022**, *62*, 199–201. [\[CrossRef\]](#)

38. Pradana, M.I.A.; Kurniati, A.P.; Wisudiawan, G.A.A. Inductive Miner Implementation to Improve Healthcare Efficiency on Indonesia National Health Insurance Data. In Proceedings of the 2022 International Conference on Data Science and Its Applications (ICoDSA), Bandung, Indonesia, 6–7 July 2022; pp. 239–244. [\[CrossRef\]](#)
39. Leemans, S.J.J.; Fahland, D.; van der Aalst, W.M.P. Discovering Block-Structured Process Models from Event Logs - A Constructive Approach. In *Application and Theory of Petri Nets and Concurrency, PETRI NETS 2013. Lecture Notes in Computer Science*; Colom, J.M., Desel, J., Eds.; Springer, Berlin/Heidelberg, Germany, 2013; Volume 7927. [\[CrossRef\]](#)
40. Sherry, K. J. *Business Process Modelling with BPMN: Modelling and Designing Business Processes Course Book Using the Business Process Model and Notation Specification Version 2.0.*; CreateSpace Independent Publishing Platform: Scotts Valley, CA, USA, 2012.

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