

Article

Industry 4.0-Based Real-Time Scheduling and Dispatching in Lean Manufacturing Systems

Muawia Ramadan ^{1,*}, Bashir Salah ² , Mohammed Othman ¹  and Arsath Abbasali Ayubali ³

¹ Department of Industrial Engineering, Faculty of Engineering and Information Technology, An-Najah National University, P.O. Box 7, Nablus, West Bank, Palestine; m_othman@najah.edu

² Department of Industrial Engineering, College of Engineering, King Saud University, P.O. Box 800, Riyadh 11421, Saudi Arabia; bsalah@ksu.edu.sa

³ Department of Mechanical Engineering, College of Engineering Guindy, Anna University, Chennai 600025, India; arsathabbas@gmail.com

* Correspondence: mramadan@najah.edu

Received: 17 February 2020; Accepted: 12 March 2020; Published: 14 March 2020



Abstract: Lean manufacturing is one of the most popular improvement agents in the pursuit of perfection. However, in today's complex and dynamic manufacturing environments, lean tools are facing an inevitable death. Industry 4.0 can be integrated with lean tools to avoid their end. Therefore, the primary purpose of this paper is to introduce an Industry 4.0-based lean framework called dynamic value stream mapping (DVSM) to digitalize lean manufacturing through the integration of lean tools and Industry 4.0 technologies. DVSM with its powerful features is proposed to be the smart IT platform that can sustain lean tools and keep them alive and effective. This paper specifically tackles the scheduling and dispatching in today's lean manufacturing environments, where the aim of this research is developing a smart lean-based production scheduling and dispatching model to achieve the lean target through optimizing the flow along the VSM and minimizing the manufacturing lead time. The developed model, called the real-time scheduling and dispatching module (RT-SDM), runs on DVSM. The RT-SDM is represented through a mathematical model using mixed integer programming. Part of the testing and verification process, a simplified IT-based software, has been developed and applied on a smart factory lab.

Keywords: lean manufacturing; dynamic value stream mapping (DVSM); production scheduling; Industry 4.0; internet of things; cyber-physical system

1. Introduction

Lean manufacturing can be considered one of the most significant contributions in the history of operations management [1,2]. However, since its inception, most of the developments in traditional lean manufacturing tools are almost done independently from using information technology [3]. It was described in [4] that the major time of semi-finished products through the value stream is wasted through waiting for processing. One of the causes behind this problem is the way of scheduling and prioritizing them. This will significantly increase the manufacturing lead time (MLT) [4].

Therefore, scheduling and dispatching strategies are considered the main pillars of lean manufacturing to enhance the pull system and thus minimize the MLT, which plays a critical role in achieving lean targets [5]. In this regard, several research papers have tried to improve scheduling and dispatching levels to tackle the recent and rapid changes in the global markets. However, most of the developed scheduling models are buried in mathematical treatments with unrealistic assumptions [6]. This makes them inappropriate and impracticable in production shop floor (PSF).

Typically, the scheduling level in lean manufacturing uses simple tool called a heijunka-board (i.e., load-leveling box) at the pacemaker workstation to enhance the pull system. Here, “kanban” acts as the nervous system of the pull system through directing materials just-in-time to workstations. Accordingly, continuous flow, i.e., “one-piece flow”, as well as the “takt-time”, which is used to synchronize the pace of production with the pace of demand, will be achieved. This works well in cases of high-volume and low-mix working environment. However, these lean tools are inefficient in today’s highly dynamic and customized manufacturing environment with uncertainty in demand and materials supply, unpredictable material flow, high variety of products that move through different routes and sequences of workstations, different priorities, process times, due dates, etc. Moreover, in a dynamic system, kanban is unable to determine the right time for dispatching and assigning a job into the machines based on changes in production constraints, like customer importance, due date, quantity, sequence of the job, the resource availability, and current workload on the PSF [6]. Therefore, the Dispatching Rules (DRs) play an important role to meet the scheduling level. The importance of DRs lies in decreasing variability, reducing waiting times, increasing utilization of resources, and improving the production smoothness [7]. In the literature, many dispatching priority or sequencing rules, such as first come, first served (FCFS), Shortest Processing Time (SPT), and Earliest Due Date (EDD), have been proposed and investigated [8–11]. For example, FCFS selects the jobs arriving at a workstation first, which means the first job will be processed first without any preferences, SPT prioritizes jobs based on the shortest processing time, and EDD selects the jobs with the earliest due date, so that the total tardiness of the whole jobs is minimized [11]. In this context, due to high variability during production runs, no specific DR works well all the time [12]. Similarly, the production scheduling is generated within a specific timeframe (e.g., per week), while unpredictable events caused either internally or externally caused by customers or suppliers occur in real-time, such as machine failure, defects and re-work, misplaced tools, production congestions, changes in order due date/quantity/cancellation, change in product design, or raw materials shortages. This may lead to an infeasible and outdated production schedule [12]. Moreover, the production manager and supervisors may unable to address and tackle the urgent problems at the right time. As a result, there is a serious gap between what is planned and what is happening in real-time on PSF, this may cause serious consequences that deteriorate the overall leanness level.

Therefore, without up-to-date information from PSF, it is impossible to make accurate shop-floor scheduling and dispatching decisions to achieve lean targets. Hence, the scheduling research needs to shift its focus to real-time and smart scheduling and product-flow optimization [13]. In this regard, Industry 4.0, with its powerful real-time data capturing technology enablers like radio-frequency identification (RFID), machine data, multiple sensors, network gateways, and others, can bridge the serious gap between reality on the PSF and the offline scheduling algorithms or mathematical models [4,14,15].

Integrating Industry 4.0 in manufacturing systems provides the advantage of real-time tracking and monitoring of the entities of manufacturing systems [14,16]. Moreover, Industry 4.0 provides the ability of identifying, tracking, communicating, and controlling smart entities such as jobs, machines, tools, fixtures, workers, and other resources along the value stream in the lean manufacturing environment. This will enhance the competitiveness and efficiency of manufacturers [14,17,18].

Recently, there have been challenges in the deployment and implementation of lean tools in manufacturing systems, preventing manufacturing organizations that have deployed lean manufacturing initiatives from successfully achieving lean goals. However, due to rapid changes in products and technologies, lean initiatives simply diminish in the long run.

The superior features of Industry 4.0 provide lean practitioners with the opportunity to build an integrated smart lean manufacturing system. Such mechanisms provide the ability to digitalize smart lean tools by employing the main components of Industry 4.0 in the field of information and communication systems, such as cyber-physical systems (CPS), the Internet of Things (IoT), and future-oriented technologies with enhanced human-machine interaction paradigms [13,19–22].

Despite the significant advancements in Industry 4.0 and real-time data capturing systems like RFID, a few studies [23–25] have investigated the possibility of using RFID to improve production scheduling and dispatching levels. For instance, [26] investigated an integrated production scheduling system with intelligent machines, where real-time information for self-optimization, self-training behaviors, and self-assigning can be captured. As a result, a machine-learning algorithm was developed as a monitoring tool for an Industry 4.0 manufacturing system. In this field, a real-time scheduling model in IoT based manufacturing environment was developed by [27]. The proposed real-time scheduling model optimally assigns tasks to machines according to their real-time status. A CPS in production controlling was addressed by [28], who introduced a CPS and big-data machining optimization system, in which the mechanism includes different production controlling aspects, such as scheduling, rescheduling, monitoring, and learning during the production run. Technically, a real-time early warning and anomalous signs detection system for production systems was developed by [29,30]. Such systems help to detect the incidences in advance and try to avoid them. The results of [29] were promising, where the prediction accuracy was near 100%.

It is obvious that none of these studies tackled scheduling and dispatching problems from lean perspectives in a dynamic manufacturing environment.

The significant contribution of this research paper is developing a smart lean-based scheduling and dispatching model in lean manufacturing. It aims to develop an Industry 4.0-based lean framework to digitalize lean manufacturing through the integration between lean tools and Industry 4.0 technologies. The research presented in this paper is organized as follows. In Section 2, the lean-based Industry 4.0 system (DVSM) is discussed. Section 3 provides an overview of the real-time scheduling and dispatching model. Next, the real-time dispatching priority generator model developed in this research is presented in Section 4. Section 5 discusses the main findings of the study. Finally, in Section 6, conclusions and opportunities for future research are presented.

2. Lean-Based Industry 4.0 System—DVSM

In today's high dynamic manufacturing system, lean tools will die without continuous care, since lean was originally designed for stable manufacturing environments. Typically, value stream mapping (VSM) is considered the best lean practice to achieve lean in different manufacturing companies. However, based on much of the literature, [4] listed 14 weakness points in the conventional VSM that make it incapable of achieving and sustaining lean strategies in today's dynamic manufacturing environments. For example, some of these weakness points are highlighted: firstly, the static features and the offline mode of the VSM make it hard to see or map the flow in real-time and capture the dynamic aspects of the manufacturing system; secondly, VSM is unable to track the actual process parameters and the performance measures of the manufacturing system; thirdly, VSM is unable to track and present the current status of the used resources as well as the work in process (WIP) inventory level in front of each process; fourthly, VSM lacks the "what-if" analysis required to optimize the production processes; finally, VSM is unable to map multiple products' flow and analyze the movement of products using shared process resources. VSM is just a snapshot from PSF and cannot dynamically interact with the current situation on the PSF in real-time. This cannot help lean practitioners in sustaining lean tools and save them from an inevitable death or obsolescence in the long-term [4,14]. In order to overcome the drawbacks of VSM in today's manufacturing environment and keep lean systems effective in the long-term, [4] introduced the idea of a dynamic and real-time VSM which is built upon the main concepts of conventional VSM, where the time-based flow is greatly emphasized and considered the most critical success factor of lean. It is called Dynamic Value Stream Mapping (DVSM).

DVSM can be defined as "a digitalized event-driven lean-based IT system that runs in real-time according to lean principles that cover all manufacturing aspects through a diversity of powerful practices and tools that are mutually supportive and synergize well together to effectively reduce wastes and maximize value" [4]. Accordingly, DVSM represents a smart, comprehensive, and integrated real-time monitoring and controlling lean-based IT mechanism for the next generation of manufacturing

systems or smart factories with dynamic and intelligent aspects concerning lean targets [4]. It can be expressed as “built-in lean Industry 4.0” or Lean 4.0 [31].

The structure of the DVSM is shown in Figure 1, it contains different real-time engines. For instance, the Actual Value Stream Mapping (AVSM) engine represents the current status on the physical world and it is considered the DVSM interface with the PSF. AVSM translates the production activities applied on products into flow in terms of time and location (i.e., VSM-format). The Virtual Value Stream Mapping (VVSM) engine represents the virtual world and it is considered the DVSM interface with the highest enterprise levels. VVSM translates the standard/planned lean-based production data that are developed in the enterprise level into the context of event-instances (i.e., VSM-format). This happens before releasing the products for production; each product should have a VVSM that represents its standard/planned lean-based data.

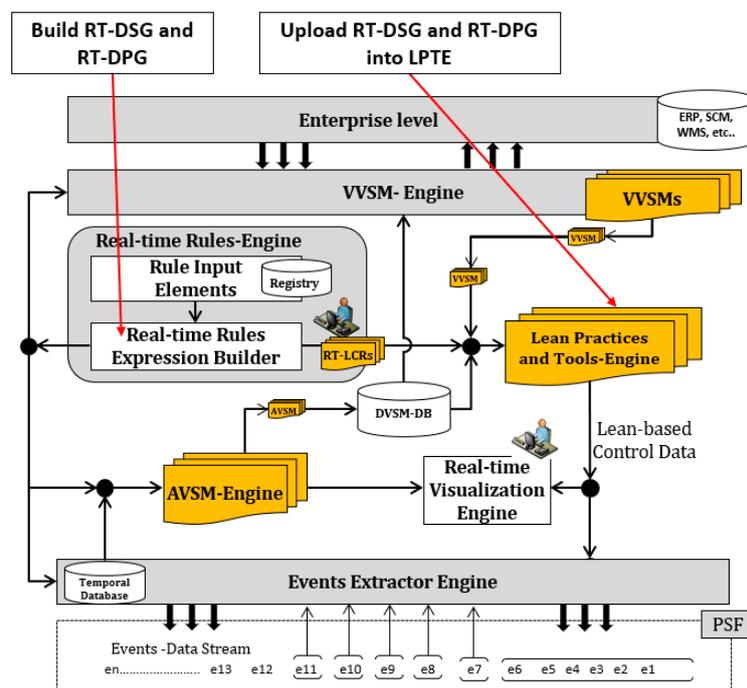


Figure 1. The IT system architecture of the Dynamic Value Stream Mapping (DVSM) for Lean 4.0.

To convert lean tools to become smart, firstly, lean specialists build real-time data-based algorithms or a mathematical model for each lean tool. As explained in next section, the Real-time Dynamic Scheduling Generator (RT-DSG) is constructed mathematically using mixed-integer programming, while in the Real-time Dispatching Priority Generator (RT-DPG) an algorithm is used. Thereafter, lean specialists use the Real-time Rules (RT-RE) engine to translate these algorithms or mathematical models into smart lean tool rules, which are called Real-Time Lean Control Rules (RT-LCRs). To run them online, the constructed RT-LCRs will be uploaded into the Lean Practices and Tools (LPTE) engine, which is considered the heart of DVSM. LPTE runs in real-time on DVSM and continuously invokes the required real-time data from both AVSM and VVSM in order to supply different RT-LCRs with real-time data to sustain the overall lean system. According to [32–34], cloud computing as one of Industry 4.0 technology enablers can significantly contribute to the realization of DVSM.

3. Real-Time Scheduling and Dispatching Model (RT-SDM)

This section introduces the RT-SDM to deal with the dynamic nature of job shops. The RT-SDM is divided into two sub-modules to run on DVSM-LPTE; the first is the Real-time Dynamic Scheduling Generator (RT-DSG) sub-module that generates the initial schedule at the global-VVSM level, the second is called the Real-time Dispatching Priority Generator (RT-DPG) sub-module, which re-prioritizes the

products that are waiting in supermarkets or queues. Both RT-DSG and RT-DPG work in parallel in order to generate the most optimal real-time re/actions.

RT-DSG consists of planning and scheduling levels, the objective functions and constraints of the constructed planning and scheduling level are built in RT-DSG, which runs online in DVSM-LPTE, which should minimize the idle time and thus total MLT, while maximizing the utilization of the available resources. The RT-DSG is equipped with RT-LCRs, which firstly are mathematically constructed and then expressed in complex event processing (CEP) format. The following section introduces the real-time mathematical model of the RT-DSG.

3.1. Mathematical Model of RT-DSG

The below proposed mathematical model for the complete planning and scheduling operation in a real-time job shop floor was constructed on the basis of mixed-integer programming. The following optimization model formulation of the flexible job shop problem adopted the modeling approach for efficient results of the problem, which are subjected to their considered objective function. Since all the available real-time constraints and processes cannot be implemented in such models, some assumptions are considered on a certain real-time basis (see Table 1).

Table 1. Mathematical model assumptions.

#	Assumption
1	Flexible job shop problem consists of a set of n independent jobs $I = i_n$, where $n \in \text{set on integers}$.
2	Each job has its own processing order through a set of M machines.
3	The operators, machines, tools, and jobs and other production-related objects are equipped with RFID-tags to become a smart object and trackable in real-time, where workstations, routs, machine areas, floor, gates, etc., are equipped with RFID readers to collect real-time production data.
4	Each operation is processed according to DVSM commands to obtain jobs from a real-time production order pool, which is responsible for prioritizing and sequencing the jobs at each workstation under each machine and continued to the next operations.
5	The number and order of processes must be performed to complete job i are defined in $VVSM_i$
6	An operation cannot be taken up for processing until its previous operations are completed.
7	The production horizon is divided into multiple time periods under time horizon T . Normally, the time unit of T is one working day (eight-unit hours of working). Therefore, the order time and due date DD are scaled in unit of working hours multiplied by the number of working days.
8	In order to capture the event-instances and data generated in real-time (measured in seconds or milliseconds) during a daily production operation run, the time horizon is further divided into multiple minor periods, which is t . Therefore, in AVSM the processing time, start time, and setup time are scaled by unit hours or minutes.
9	AM_i is a set of alternate machines which can process job i , $AM_i \subseteq M$ for a given processing time under the machines that suit the specified processing mechanism.
10	In each stage g , the WIPs are stored in a buffer B_g . The real-time data of the new arriving jobs and the old jobs which are available in the buffer can be obtained from AVSM.
11	The RT-DSG releases jobs to a specific machine and its labors according to sequencing and dispatching rules that have been selected by the RT-DPG with respect to AVSM data.
12	These sequencing and dispatching rules are assigned with respect to the weighted priority ω .
13	At every stage, the sequencing and the dispatching of the jobs and production orders must be accomplished on a check time of every ($Trs = \text{time of rescheduling}$) minutes to follow a real-time pull dispatching strategy.

3.1.1. Formulation

The formulation of mathematical optimization models in planning and scheduling levels is described, starting with defining the indices and sets.

3.1.2. Indices and Sets

The indices and sets are defined where the source of real-time information, either from VVSM or AVSM, is written in front of each (see Table 2).

Table 2. Description of indices and sets.

Indices and Sets	Description
i	Jobs ($i, i + 1 \in I$) (VVSM, for each i there is predefined (VVSM _{i}))
I	The total set of jobs (G-VVSM), where G = global VVSM, which represents the entire value stream for all products)
k	Machines ($k \in M$) (Global-VVSM)
M	The total set of machines (G-VVSM)
G	Total number of stages (AVSM)
g	Set of individual stages ($g \in G$) (AVSM)
PO	Set of total production orders (VVSM)
M_g	Total set of machines in stage g (G-VVSM)
AM_i	The set of alternative machines on which job i can be processed ($AM_i \subseteq M$) (VVSM _{i})
mO	Set of machine operators (VVSM)
O_{it_c}	Demand of job i that cannot be produced on time in day t_c (AVSM)
T_{rs}	Time of rescheduling in minutes

3.1.3. Input Parameters

DD_i	Due date of i_{th} job (VVSM)
DD_{PO}	Due date of production order (VVSM)
RD_i	Release date of i_{th} job (VVSM)
T	Time horizon (G-VVSM)
T_{POg}	Time horizon of production order (VVSM)
t_c	Current time (AVSM)
t_i^s	RFID stamp for processing start event of job i (AVSM)
t_i^f	RFID stamp for processing end event of job i (AVSM)
$N_r(PO)$	Number of machines that can process the product group (VVSM)
B_g	Buffer capacity at stage g (VVSM)
FO_i	Finish time of job i (VVSM)
ST_i	Setup time of job i of (VVSM)
P_i	Processing time of i (VVSM)
LST	Estimate latest start time (VVSM)
FO_{PO_g}	Finish time of production order PO_g (VVSM)
ST_{PO_g}	Setup time of production order PO_g (VVSM)
P_{PO_g}	Processing time of production order PO_g (VVSM)
RPT_i	Required processing time of i (VVSM)
$R(M_g)_i$	Required total set of machines at stage g for job i (VVSM)
CT	Tool change over time (VVSM)
ω_{PO_g}	Weighted priority of PO production order (1—highest and 5—least priority) (VVSM)
ω_i	Weighted priority of job i (1—highest and 5—least priority) (VVSM)

3.1.4. Input Variables

SO_i	Start time of job i (VVSM)
n_{CT_i}	Number of tool changeovers (AVSM)
$n_{CT_{PO}}$	Number of tool changeovers of production order (AVSM)
CL_{PO_g}	The cumulative workload of production order at stage g
AL_{PO}	Average workload of production order
C_i	The completion time of job i (AVSM, $C_i = t_i^s - t_i^f$)
C_{PO_g}	The completion time of PO_g (AVSM)
C_{max}	Maximum completion time of all jobs (make span) (VVSM)

3.1.5. Decision Variables

$$Y_{i,i+1,k} = \begin{cases} 1, & \text{if job } i + 1 \text{ precedes job } i \text{ on machine } k \\ 0, & \text{otherwise} \end{cases} \quad (\text{AVSM})$$

$$MS_{kg} = \begin{cases} 1, & \text{if machine } k \text{ at stage } g \text{ is selected,} \\ 0, & \text{otherwise} \end{cases} \quad (\text{AVSM})$$

$$MS(PO_{kg}) = \begin{cases} 1, & \text{if } PO_g \text{ can be processed by machine } k, \\ 0, & \text{Otherwise} \end{cases} \quad AM_i \subseteq M \quad (\text{AVSM})$$

$$R(mO_k, M_g) = \begin{cases} 1, & \text{if the operator is detected on machine } k \text{ at stage } g \\ 0, & \text{otherwise} \end{cases} \quad (\text{AVSM})$$

3.2. Production Planning Level

In the production planning level, the jobs are planned according to the following criteria:

1. Grouping the jobs in production orders with respect to the weighted priority of the jobs;
2. Choosing jobs with the minimum number of tool changeovers and total tool changeover time; and
3. Obtaining an optimal start time for each job with the bounded release date and due date.

These three criteria are considered in the planning stage, to optimally plan all the jobs into the respective production order, *PO*. Here, mainly the number of tool changeovers and the tool changeover time are considered, because different jobs must be processed in different machines according to their material and may need different tools and thus different tool change times. Also, an optimal start time for each job is determined from the objective function, considering the release date and due date as a bound. Hence, in this planning stage, grouping of jobs into the production orders according to the three considered criteria are analyzed and variables are constructed mathematically.

Objective function:

$$\text{Min} \sum_i^I (CT_i * n_{CT_i}) + [(SO_i + ST_i + P_i) - DD_i]. \quad (1)$$

Constraints:

$$RD_i \leq SO_i \leq DD_i \quad (2)$$

$$n_{CT_i} \leq \min(n_{CT_{PO}}), \forall i \in I \quad (3)$$

$$\min(n_{CT_i}) \leq n_{CT_i} \leq \max(n_{CT_i}), \forall i \in I \quad (4)$$

$$C_i \leq RPT_i \quad \forall i \in I \quad (5)$$

$$\sum_{i=1}^I (SO_i + ST_i + P_i) \leq DD_i. \quad (6)$$

The above formulated constraints work with the constructed objective function together to obtain an optimal production plan. The above equations are explained as follows: Equation (1) considers minimizing the total tool changeover time and an optimal start time; with these both attributes, possibly an effective planning level can be achieved. Equation (2) ensures that the start time of each job should lie between the release date and the due date. Equation (3) ensures that all jobs which have the minimum number of tool changeovers are grouped in *PO* (batch). Equation (4) ensures that the number of tool changeovers should lie between the minimum and maximum number of tool changeovers of all the jobs. Equation (5) ensures the completion time of each job, and Equation (6) maintains the finishing operation of each job within its due date.

For the effective functioning of the objective, certain relationships have been constructed mathematically to follow a smooth planning stage, as follows:

$$\omega_i \leq \omega_{i+1}, \quad (7)$$

$$1 \leq \omega_i \leq 5, \quad (8)$$

$$Y_{i,i+1,k} \leq 1 \in \{0, 1\} \quad \forall i, i+1 \in I, \forall k \in M_g. \quad (9)$$

Equation (7) ensures the sequence of weighted priority of each job according to the tool change over time. Equation (8) ensures that the weighted priority is assigned between 1 and 5, and Equation (9) maintains the precedence relationship of the jobs.

In the above constructed planning level, the considered weighted priority ω_i is estimated in accordance with the following relationship:

$$\forall i \in I, \text{ in real time job pool, if } n_{CT} \uparrow, \text{ then } \omega_i \downarrow. \quad (10)$$

Equation (10) allocates the value of ω_i . If there is a greater number of tool changeovers, then the value of ω_i is set as low, thereby ω_i has the highest weighted priority.

3.3. Production Scheduling Level

In the production scheduling level, each job contained in the set of each production orders must be sequenced in a real-time job pool. This scheduling stage problem can be considered as parallel machine scheduling with maximizing the utilization of the stages. Since each stage is dependent on one another, there will be an influence on time if there are any discrepancies on any of the dependent stages.

Here, scheduling level is carried out with constraints such as balancing the workload in each stage considering the time horizon T and available working load at each stage. Flowingly, to make sure the effective allocation of production order at each stage is allocated, including the buffer status.

The basic objective of workload balancing is to achieve effective utilization of production capacities. The concept of balancing is to evenly distribute the production orders among the stages. It also may be considered as evenly distributing jobs among the contained machines. The workflow in each stage is calculated before assigning the production orders. Workload balancing aims at providing equal distribution of load on parallel machines and helps to reduce the total elapsed time of production orders. The aim of the scheduling process is to increase the utilization of stages, thereby increasing the utilization of each machine at each stage.

The parallel machine scheduling problem consists of a set of k machines $k \in M$ at stage g , $\forall g \in G$ and a set of production orders PO containing i jobs $\forall i \in I$. Each PO requires processing time P_{PO} , for it to be processed on any one of the stages. When a stage becomes available, a production order from the list of unscheduled PO s with the highest priority based on ω_{PO} is selected and assigned to the stage for processing.

This scheduling level focuses on the measure of workload balancing among the stages. Instead of assigning a production order to any first available stage, the stages with less cumulative workload are selected for a production order from the unscheduled list of production orders. The cumulative workload of a stage is equal to the sum of all processing times of jobs, which is contained in the production order which is assigned to that stage. The cumulative workload of all stages is compared and the stage with the minimum cumulative workload is selected for assigning the next production order from the list of production orders arriving from the planning stage.

The difference between the squares of the cumulative workload of production order at the stages and the average workload of the production order with respect to total number of stages is called an imbalance in workload. Imbalances IM_g at each stage are expressed as follows:

$$IM_g = \frac{(CL_{PO_g} - AL_{PO})^2}{G}.$$

In our real-time scheduling model, the workload balancing strategies are adopted to minimize these imbalances on the stages. Uniformly distributed workloads on all the stages is the best schedule. Therefore, the objective function is constructed to minimize the imbalances in each stage, thereby allocating the production order to the stage which has minimum imbalance.

The objective function is a least-squares estimation function and therefore it can be converted into a linear objective function by using the least absolute deviation estimation technique, which minimizes the sum of absolute errors. Hence, the objective function takes the following form:

Objective function:

$$\text{Minimize } \frac{\max \sum |CL_{PO_g} - AL_{PO}|}{G}, \forall PO \in G, \quad (11)$$

subjected to:

$$AL_{PO} = \frac{\sum_{g=1}^G CL_{PO_g}}{G}, \forall PO \in G, \quad (12)$$

$$CL_{PO_g} = \frac{\sum P_i}{M_g}, \forall i \in I \text{ and } \forall g \in G. \quad (13)$$

The constraints are formulated in order to distribute the production orders to achieve a balanced workload, thereby minimizing the imbalances in workload of the stage for each production order. The above-constructed constraints and objective functions are explained in detail as follows. Equation (11) is the objective function of scheduling level which ensures the minimization of imbalances in each stage. Equation (12) determines the average workload of all the stages at the current time. Equation (13) determines the cumulative workload of single stage.

For the effective functioning of the objective and related constraints, certain relationships have been constructed mathematically to follow a smooth scheduling stage:

$$FO_{PO_g} \leq SO_{PO_g} + ST_{PO_g} + P_{PO_g}, \quad \forall g \in G, \quad (14)$$

$$\forall i \in PO_g; \text{ in stage } g \Rightarrow \max DD_i \leq T_{PO_g}, \quad (15)$$

$$\forall i \in PO_g; \text{ in stage } g \Rightarrow FO_i \leq \max DD_i, \quad (16)$$

$$\omega_{PO_g} \leq \omega_{PO_{g+1}} \quad \forall g \in G, \quad (17)$$

$$\sum t_i^s \leq SO_i, \quad \forall i \in PO_g, \quad (18)$$

$$\sum t_i^f \leq FO_i, \quad \forall i \in PO_g, \quad (19)$$

$$\sum_{k=1}^M \sum_{g=1}^G MS_{kg} = R(M_g)_i, \quad \forall i \in I. \quad (20)$$

In the scheduling level, according to the above-described mathematical relationships, which support the scheduling stage, the inputs include production orders, maximum due date of jobs, and production order at each stage. Current positions of the production orders are tracked by RFID in real-time. The relationships constructed above are explained in detail, as follows: Equation (14) ensures the finishing time of each job. Equation (15) ensures that all jobs which are allocated to PO_g arrive between the planned time horizon T and maximum due date of all jobs of the respective production order. Equation (16) ensures that all jobs are completed within their maximum due date. Equation (17)

ensures the sequence of the scheduling the production orders at each stage with respect to the weighted priority of the production order. Equation (18) ensures the capturing of the real-time processing start event by RFID. Equation (19) ensures the capturing of the real-time processing end event by RFID, and Equation (20) ensures that each job has completed all required processes in the respective stage.

In above-constructed scheduling level, the considered weighted priority of production order ω_{PO} is estimated in accordance with the following relationship:

$$\forall PO \in G, \text{ in real time, if } IM_g \downarrow, \text{ then } \omega_{PO} \downarrow, \tag{21}$$

where the constraint is: allocates the value of ω_{PO} . If the value of workload imbalance is low, then the value of ω_{PO} is set as low, thereby ω_{PO} has highest weighted priority.

Decision Output:

In the scheduling level, the real-time re/action is the assignment of a production order which has the minimum imbalance of workload to the initial stage. This real-time re/action decides which production order should be allocated to which stage and the following sequence of the production order in the scheduling level. The real-time functionality of the planning and scheduling model is illustrated in the flowchart in Figure 2.

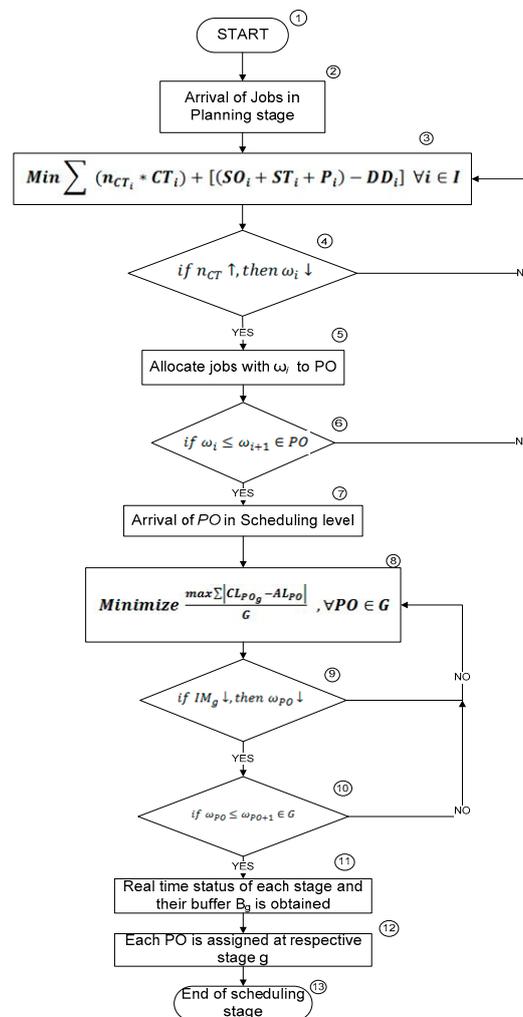


Figure 2. The real-time functionality framework of the planning and scheduling model.

4. Real-Time Dispatching Priority Generator (RT-DPG) Model

The horizon of the scheduler is measured in units of days or weeks, depending on uncertainty, while the dispatching horizon is measured in minutes or hours, depending on deterministic real-time data [35]. Therefore, in an attempt to follow and meet what is already planned and scheduled in the RT-DSG module in real-time, and mitigate the consequences of the unplanned or unpredictable events (e.g., machine breakdowns, material shortage, defective problems, disparity operators' performance), this part presents another real-time production control module, namely RT-DPG. RT-DPG utilizes RFID event-instances in AVSM and VVSM to smartly re-sequence or reprioritize products in real-time at the local workstations, supermarkets, or machines, queues, which cannot be addressed through RT-DSG.

The RT-DPG is proposed to run in real-time in DVSM-LPTE in order to generate the optimal sequence of WIPs that are waiting either in the machine's queue or workstation's supermarket throughout the production processes.

RT-DPG is proposed to be translated into RT-LCRs to live prioritize the material flow in unpredictable dynamic production environments to reduce the total waiting time along the value stream and maximize the utilization of all resources, thereby reducing the overall manufacturing lead time, avoiding discrepancies in executing the planned schedules, and meeting customer demand.

As mentioned, in reality, no specific traditional dispatching rule works well all the time [12]. Therefore, a dynamic priority method was developed to reprioritize the products in real-time. Here, we propose what is called the Dispatching Priority Value (DPV) method, which starts by integer "1" as the highest DPV and extends according to the number of product types that are waiting—a smaller value represents higher priority. According to the RT-DPG, the product with the highest DPV will be displayed on the tablet of logistic operators to be dispatched.

The Framework of the RT-Pull Dispatching Model

Once the scheduling phase is completed in each stage, now each product should be preceded for each operation in the machine which is assigned in that stage. This is hereby controlled by RT-DPG to dispatch each job from the constraint scheduled pool of jobs. For this dispatching procedure, the pull dispatching strategy is followed with the real-time function.

In general, the pull strategy control works with Kanban systems. In a pull scheduling system, a certain amount of inventory is held at each workstation, and the next process has been consumed with orders of its replenishment through the rate. Using the concept of pull dispatch, we propose a framework of an RT-pull dispatching system for our shop floor dispatching, shown in Figure 3. This RT-pull dispatching system includes two stages: (1) the product release times; (2) RT-dispatching rules.

Step 1: Back propagation scheduling:

This step aims to consider the latest starting time (LST) for all products on all machines to fulfill the due dates. Here, the back-propagation scheduling approach is used to take consideration of the LST, which is based largely on the due date and processing time for all products.

Step 2: Real-Time production orders prioritizing and planning:

This step aims to obtain the real-time product information, such as the arrival material data. Then, we can obtain the earliest starting time (EST) for all products on all machines after matching the production orders and the arrival customer order data. The allocation of the arriving customer order is mainly based on the first come, first served rule and based on the due date of the product. This process is initiated after the real time scheduling phase is complete.

Step 3: Determine the PO_g release order:

This step aims to obtain the appropriate product dispatch times. Using the real-time data from the resources AVSM, VVSM, G-VVSM information, we can obtain the time scale (LST) for all jobs on all machines, stages, buffers, and current time. That is, if the lot release times are within the time scale value, it does not affect the product delivery.

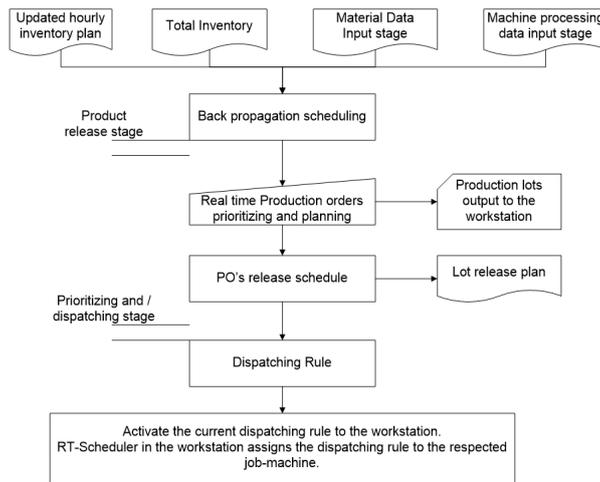


Figure 3. RT-pull dispatching system.

With the application of the above steps, we propose a heuristic algorithm for the product dispatching times for the multi-stage parallel flexible shop floor with the real-time control of each work stage. Also, there may be chance of mismatch between the dispatching jobs; these may be due to the disturbed updating of the online data transponding system or may be due to the customers' immediate order placing or the canceling of an order. One way in which these mismatches can be avoided between some jobs is that each job is activated in such a way that they already are assigned with dispatching priority value “ DPV_i ”, $i = 1, 2, 3 \dots n$, where “1” is the highest priority dispatchable job, and “n” is the number of jobs with respect to DPV, which decides in which sequence the rules KC_i must be checked and executed. For the execution procedure the pseudo codes are expressed as below, Figure 4 describes how the RT-DPG works.

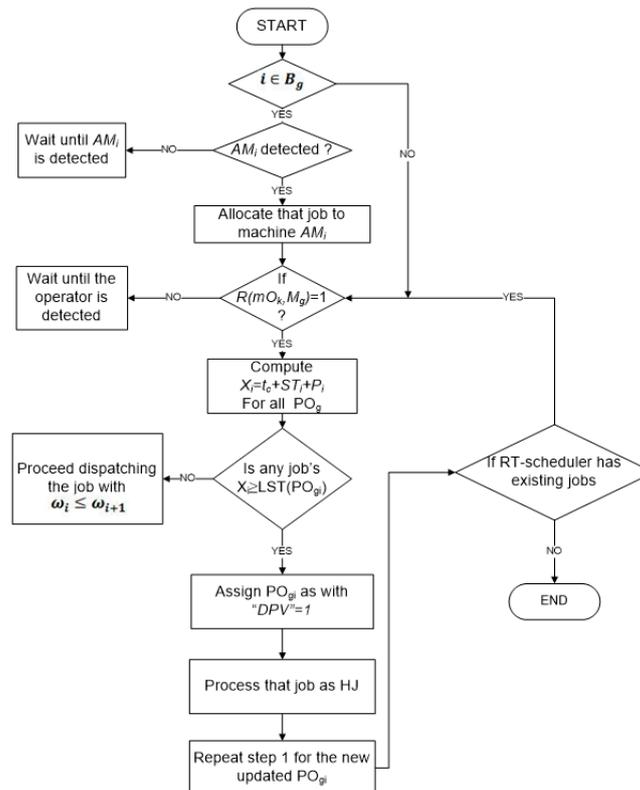


Figure 4. Description of the functionality of the Real-time Dispatching Priority Generator (RT-DPG) model.

The following relationships are used to estimate latest start time (LST):

$$LST_i = DD_i - P_i - ST_i - B_{ig}, \quad \forall i \in PO_g,$$

RT-LCR: Job dispatch schedule in RT-DPG

READ buffer information from stage g as B_g , available time of job i from $g - 1$ (1)
IF any job $i \in B_g$ is detected in buffer, (2)
 THEN retrieve the time t_i in B_g in real time (3)
 IF any available machine $AM_i \in (AM_i \subseteq M)$ is detected for the time t_i , (4)
THEN set i to AM_i (5)
ELSE
GOTO step 6
ENDIF
 IF $R(mO_k, M_g) = 1$, (6)
 THEN assign PO with highest priority of ω_{PO} to stage g ,
ELSE wait for $R(mO_k, M_g) = 1$
ENDIF
J = 1
N = Size of PO
WHILE $\{J \leq N\}$
 EVALUATE $[X_i = t_c + ST_i + P_i]$ for each job in PO_g , (7)
 IF $[X_i \geq LST(PO_{g_i})]$ is satisfied, select highest X_i (8)
 THEN set $[PO_{g_i}]$ with $DPV_i = 1$ where $[DPV_i(PO_{g_i})] = 1$, (9)
EXECUTE $KC_i(PO_{g_i})$ as HJ by start processing PO_{g_i}
ELSE (10)
 EXECUTE to dispatch jobs according to $\omega_i \leq \omega_{i+1}, \forall i \in PO_g$ (11)
ENDIF
J + 1
ENDWHILE (12)
CHECK for any PO in RT-DPG, (13)
IF any PO detected
THEN, start with step 1.
ENDIF
ELSE
 IF $R(mO_k, M_g) = 1$,
 THEN assign PO with highest priority of ω_{PO} to stage g ,
ELSE wait for $R(mO_k, M_g) = 1$
ENDIF
J = 1
N = Size of PO
WHILE $\{J \leq N\}$
 EVALUATE $[X_i = t_c + ST_i + P_i]$ for each jobs in PO_g ,
 IF $[X_i \geq LST(PO_{g_i})]$ is satisfied, select highest X_i
 THEN set $[PO_{g_i}]$ with $DPV_i = 1$ where $[DPV_i(PO_{g_i})] = 1$,
EXECUTE $KC_i(PO_{g_i})$ as HJ by start processing PO_{g_i}
ELSE
 EXECUTE to dispatch jobs according to $\omega_i \leq \omega_{i+1}, \forall i \in PO_g$
ENDIF
J + 1
ENDWHILE
CHECK for any PO in RT-DPG,
IF any PO detected
THEN start with step 1.
ENDIF

Step 1 retrieves the real-time buffer information from each stage and the status of the stage; step 2 checks for the availability of any job in the buffer. Step 3 obtains the time of the job present in the buffer, including the buffer time. Step 4 ensures for any available idle alternate machine on which the job in the buffer can be processed. Step 5: if any available alternate machine which can process the same job is found then that job in buffer is set on that machine. Step 6 detects the availability of the machine operator. Step 7 computes the value of $[X_i = t_c + ST_i + P_i]$, where t_c is the current time, which is updated by RFID information. Step 8 compares the value of X_i to the entire job's LST, which is available in PO_g . Step 9: if X_i is greater than any job's latest start time (LST) from overall jobs in PO, set the dispatching priority value as the highest priority 1, and then execute that job as a hot job and process that first. Step 10: if any of the jobs in PO_g does not satisfy step 8. Step 11: further process of dispatching the jobs with $\omega_i \leq \omega_{i+1}$ continues until no other products remain in the buffer and in production order. Step 12: the processing of first production order ends. Step 13: the RT-DPG checks for any further available production orders and if any production order is detected, it starts dispatching them.

5. Results and Discussion

The feasibility of DVSM was demonstrated through developing different scenarios to be applied on a smart factory/Industry 4.0 lab. Figure 5 illustrates the testing Industry 4.0 lab in King Saud University. It was a smart reconfigurable manufacturing system (RMS), developed step-by-step from scratch. The latest development was done by [36], where a virtual reality system was integrated with it; thus, students became able to anticipate the RMS design, interact with its operations, and understand them, and evaluate its performance.



Figure 5. Industry 4.0 lab in King Saud University.

This Industry 4.0 lab consisted of a fully automated yoghurt filling machine equipped with real-time data capturing systems, such as an RFID system, different types of sensors, and machine data from a robot arm, air compressor, and conveyor motor. Those technologies are considered Industry 4.0 real-time data capturing enablers. They supplied the DVSM with real-time data. The RFID readers and sensors were used to track smart products with required timestamps, such as processing start/end, setup start/end, or machineries like machine breakdown start/end or maintenance start/end.

An initial or alpha DVSM-IT system was developed to run different lean tools. For RT-DSG, RT-LCRs can be constructed and coded, where the required real-time data can be invoked and processed to generate reactions or commands in real-time.

The following production situation was proposed to test our concepts: the final products were four types of yogurt jars, three of them with flavors. There were three sizes of yogurt jars. Therefore, it was a high-mix production system with 12 product types. In order to test for mass customized

products, it was proposed that each customer could request their own label on the yogurt jar. The filled yogurt jars were packed and then stored in a refrigerated automated storage and retrieval system. The empty jars were available in front of the filling machine. They were labeled according to customer requests, where RFID-tag was attached in this stage. The customer orders were received online; here the students were the customers, where each of them placed their order through the IT-platform using their mobile phone or tablet. The maximum quantity of each order was 10 yogurt jars. To start the production, the orders were initially scheduled using SimAL scheduling software.

In order to prove the effectiveness of RT-DPG against the traditional DRs, three scenarios were proposed. The first two scenarios represented the traditional DRs, namely EDD and FCFS, whereas the third scenario represented our RT-DPG concept. To do so, several incidents and disruptions were proposed to be done intentionally during the production run, such as changing the priority of two orders, changing the due date of an order, failure in the conveyor motor for five minutes, unavailability of the small size of empty jars, unavailability of yogurt with mango flavor for three minutes. According to these incidents or unpredictable events, several RT-LCRs were constructed and coded to be activated on the IT-platform during the production run. Initially, these RT-LCRs were tested and validated using simulation software ProModel. The focus of the constructed RT-LCRs was to reduce the MLT through avoidance or reducing the impact of waste causes. In order to construct the RT-LCRs, the event-instances of the standard VSM were defined and saved on the VVSM-database, and several event-instances were defined and sorted to be captured either from AVSM or VVSM. For example, the customer orders and their status were considered as VVSM real-time data, where filling start/end, jar arrives/depart were captured from AVSM. In addition, reactions or command names were defined to be executed by the machines or workers.

As a result, once an incident was detected, the DVSM-IT activated the corresponding RT-LCR, and accordingly the DVSM generated the most suitable commands, which showed which type of yogurt jar must be uploaded on the conveyor. Therefore, the dynamic behavior of the system was captured, and the suitable reaction was generated. On the other hand, for two DRs no changes could be made in response to the dynamic behavior in the manufacturing system. The above scenarios were applied and repeated in the Industry 4.0 lab several times, and the results were precise, since it was a dynamic manufacturing environment. Therefore, the average values of the performance criteria were taken for RT-DPGs, FCFS, and EDD. Table 3 illustrates the results of each scenario in addition to simulation results that were used to test the validity of RT-LCRs. The simulation results were not equal to RT-LCR results, since the simulation could not address some real events which could not be modeled; thus, their impacts will be propagated and influence the performance criteria, while in reality, the events become deterministic in real-time and they will be directly solved through a suitable reaction.

Table 3. The performance of Industry 4.0 under different dispatching methods (in minutes).

DR	MLT	Max. Tardiness	Total Tardiness	Total Waiting Time
First Come First Served (FCFS)	76	12	77	111
Earliest Due Date (EDD)	91	9	68	102
Simulation	61	7	39	68
RT-LCR	53	3	21	47

Both RT-DSG and RT-DPG work in parallel in order to generate the most optimal real-time re/actions. If the collection of reactions in the RT-DPG level will significantly impact the time horizon in RT-DSG level, then the RT-LCRs in the RT-DSG will be activated to reschedule the overall production orders.

Finally, the results show the ability of DVSM to control the flow of smart products and re-sequence/re-prioritize them in real-time to avoid hindrance or tardiness along the value stream.

6. Conclusions

This paper introduced the DVSM, which can be used as an IT platform for the digitalization of lean tools to be smartly implemented, sustained, and keep them alive and more efficient in today's dynamic manufacturing systems. The idea of DVSM comes from the fact that lean extends to cover all manufacturing aspects through a diversity of powerful tools that are mutually supportive and synergize well together to effectively reduce waste and maximize value. Therefore, DVSM is considered a digitalized smart real-time lean-based mechanism to achieve an intelligent, comprehensive, integrated, and holistic real-time lean-based manufacturing system.

This research addressed the challenges of lean manufacturing in today's manufacturing systems, where lean has become inefficient. In addition, the paper explored the adoption benefits of RFID and recent advancements in information and communication technologies in manufacturing facilities to effectively reduce waste and maximize value in a real-time lean-based manufacturing system.

The focus was on developing a smart real-time scheduling and dispatching model in a lean manufacturing environment. The RT-SDM can track the flow of products and detect the mismatches and discrepancies between the physical and virtual world that are caused by lean waste. RT-SDM runs in real-time on DVSM to avoid a frozen production schedule and dispatching rule, and accordingly generates suitable reactions and commands to be executed either by machines or a human to mitigate the impact of incidents and try to re-match between VVSM and AVSM.

The proposed DVSM and the developed smart lean-based scheduling and dispatching models were tested and verified through developing an alpha IT platform for a lean smart factory lab in the University of King Saud. The results show the significant improvements on some performance measures after running RT-SDM.

Although several challenges were faced, the results are promising and pave the way for future research to increase the efficiency of DVSM. Technically, overlapping between RFID readings in contiguous points could be a future work. Also, huge amount of data needs more focus in the big data field. From the lean perspective, more attention must be given to the conflicts between commands that are generated based on different RT-LCRs; they must be perfectly coordinated.

Author Contributions: Conceptualization M.R.; Data curation B.S.; Formal analysis M.R. and M.O.; Funding acquisition, B.S.; Investigation M.R.; Methodology M.R., M.O. and A.A.A.; Project administration M.R. and B.S.; Resources B.S.; Software B.S. and M.R.; Supervision M.R. and B.S.; Validation M.R., M.O. and B.S.; Visualization M.R.; Writing—original draft M.R. and A.A.A.; Writing—review and editing M.R. and M.O. All authors have read and agreed to the published version of the manuscript.

Funding: This study received funding from King Saud University through Researchers supporting project number (RSP-2019/145).

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

References

1. Zhang, K.; Qu, T. IoT-enabled dynamic lean control mechanism for typical production systems. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1009–1023. [CrossRef]
2. Sony, M. Industry 4.0 and lean management: A proposed integration model and research propositions. *Prod. Manuf. Res.* **2018**, *6*, 416–432. [CrossRef]
3. Kang, H.S.; Lee, J.Y. Smart Manufacturing: Past research, present findings, and future directions. *IJPEM-GT* **2016**, *3*, 111–128. [CrossRef]
4. Ramadan, M. RFID-Enabled Dynamic Value Stream Mapping for Smart Real-Time Lean-Based Manufacturing System. Ph.D. Thesis, Duisburg-Essen University, Duisburg, Germany, 2016.
5. McKay, N.K.; Wiers, V.C. Planning, scheduling and dispatching tasks in production control. *Cogn. Technol. Work* **2003**, *5*, 82–93. [CrossRef]
6. Velaga, P. Various Approaches to Production Scheduling in Job Shops. Available online: https://optisol.biz/job_shop_scheduling.html (accessed on 25 June 2016).

7. Bohnen, F.; Buhl, M. Systematic procedure for leveling of low volume and high mix production. *J. Manuf. Sci. Technol.* **2013**, *6*, 53–58. [[CrossRef](#)]
8. Heizer, J.; Render, B. *Operations Management*, 10th ed.; Pearson: London, UK, 2011; p. 628.
9. Pinedo, M. *Scheduling Theory, Algorithms, and Systems*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2008; p. 377.
10. Blackstone, J.H.; Don, T.P. A State-of-the-art Survey of Dispatching Rules for Manufacturing Job Shop Operations. *Int. J. Prod. Res.* **1982**, *20*, 27–45. [[CrossRef](#)]
11. Jeong, K.-C.; Kim, Y.-D. A real-time scheduling mechanism for a flexible manufacturing system: Using simulation and dispatching rules. *Int. J. Prod. Res.* **1998**, *36*, 2609–2626. [[CrossRef](#)]
12. Hopp, W.; Spearman, M. *Factory Physics: Foundations of Manufacturing Management*, 2nd ed.; McGraw-Hill/Irwin: New York, NY, USA, 2008; pp. 488–501.
13. Jian, Z.; Ding, G. Review of job shop scheduling research and its new perspectives under Industry 4.0. *J. Intell. Manuf.* **2019**, *30*, 1809–1830.
14. Huang, Z.; Kim, J. Industry 4.0: Development of a multi-agent system for dynamic value stream mapping in SMEs. *J. Manuf. Syst.* **2019**, *52*, 1–12. [[CrossRef](#)]
15. Rossit, D.A.; Tohmé, F. Industry 4.0: Smart Scheduling. *Int. J. Prod. Res.* **2019**, *57*, 3802–3813. [[CrossRef](#)]
16. Kamble, S.; Gunasekaran, A. Industry 4.0 and lean manufacturing practices for sustainable organizational performance in Indian manufacturing companies. *Int. J. Prod. Res.* **2019**, *58*, 1–19.
17. Sanders, A.; Elangeswaran, C. Industry 4.0 implies lean manufacturing: Research activities in industry 4.0 function as enablers for lean manufacturing. *J. Ind. Eng. Manag.* **2016**, *9*, 811–833. [[CrossRef](#)]
18. Kagermann, H. *Change through Digitization-Value Creation in the Age of Industry 4.0. Management of Permanent Change*; Springer Gabler: Wiesbaden, Germany, 2015; pp. 23–45.
19. Lasi, H.; Fettke, P. Industry 4.0. *Bus. Inf. Syst. Eng.* **2014**, *6*, 239–242. [[CrossRef](#)]
20. Posada, J.; Carlos, T. Visual computing as a key enabling technology for industrie 4.0 and industrial internet. *IEEE Comput. Graph.* **2015**, *35*, 26–40. [[CrossRef](#)] [[PubMed](#)]
21. Valdeza, A.C.; Braunera, P. Reducing complexity with simplicity-usability methods for industry 4.0. In Proceedings of the 19th Triennial Congress of the IEA, Melbourne, Australia, 9–14 August 2015; Volume 9, p. 14.
22. Longo, F.; Nicoletti, L. Smart operators in industry 4.0: A human-centered approach to enhance operators' capabilities and competencies within the new smart factory context. *Comput. Ind. Eng.* **2007**, *113*, 144–159. [[CrossRef](#)]
23. Chongwatpol, J.; Sharda, R. RFID-enabled track and traceability in job-shop scheduling environment. *Eur. J. Oper. Res.* **2013**, *227*, 453–463. [[CrossRef](#)]
24. Zhong, R.Y.; Huang, G.Q. Mining SOTs and dispatching rules from RFID-enabled real-time shopfloor production data. *J. Intell. Manuf.* **2014**, *25*, 1–19. [[CrossRef](#)]
25. Zhong, R.Y.; Huang, G.Q.; Dai, Q.Y. RFID-enabled real-time manufacturing execution system for discrete manufacturing: Software design and implementation. In Proceedings of the 2011 International Conference on Networking, Sensing and Control, Delft, The Netherlands, 11–13 April 2011.
26. Fu, Y.; Ding, J. Two-objective stochastic flow-shop scheduling with deteriorating and learning effect in Industry 4.0-based manufacturing system. *Appl. Soft Comput.* **2018**, *68*, 847–855. [[CrossRef](#)]
27. Wang, J.; Zhang, Y. Multiagent and bargaining-game-based real-time scheduling for internet of things-enabled flexible job shop. *IEEE Internet Things* **2019**, *6*, 2518–2531. [[CrossRef](#)]
28. Liang, Y.C.; Lu, X. Cyber Physical System and Big Data enabled energy efficient machining optimisation. *J. Clean. Prod.* **2018**, *187*, 46–62. [[CrossRef](#)]
29. Kim, J.; Hwangbo, H. Sensor-Based Real-Time Detection in Vulcanization Control Using Machine Learning and Pattern Clustering. *Sensors* **2018**, *18*, 3123. [[CrossRef](#)] [[PubMed](#)]
30. Kim, J.; Hwangbo, H. Real-Time Early Warning System for Sustainable and Intelligent Plastic Film Manufacturing. *Sustainability* **2019**, *11*, 1490. [[CrossRef](#)]
31. Ramadan, M.; Salah, B. Smart Lean Manufacturing in the Context of Industry 4.0: A Case Study. *Int. J. IME* **2019**, *13*, 174–181.
32. Xu, L.D.; Xu, E.L. Industry 4.0: State of the art and future trends. *Int. J. Prod. Res.* **2018**, *56*, 2941–2962. [[CrossRef](#)]

33. Al-maamari, A.; Omara, F.A. Task Scheduling using Hybrid Algorithm in Cloud Computing Environments. *IOSR J. Comput. Eng.* **2015**, *17*, 96–106.
34. Mitra, A.; Kundu, A. A cost-efficient one time password-based authentication in cloud environment using equal length cellular automata. *J. Ind. Inf. Integr.* **2017**, *5*, 17–25. [[CrossRef](#)]
35. Mönch, L.; Fowler, J.W. *Production Planning and Control for Semiconductor Wafer Fabrication Facilities: Modeling, Analysis, and Systems*, 1st ed.; Springer: New York, NY, USA, 2013; Volume 52, p. 65.
36. Salah, B.; Abidi, M.H. Virtual reality-based engineering education to enhance manufacturing sustainability in industry 4.0. *Sustainability* **2019**, *11*, 1477. [[CrossRef](#)]



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).