Continuous Workload Control Order Release Revisited:
An Assessment by Simulation

Matthias Thürer*, Ting Qu, Mark Stevenson, Thomas Maschek and Moacir Godinho Filho

Name: Matthias Thürer
Institution: Federal University of São Carlos
Address: Department of Industrial Engineering
               Rodovia Washington Luís, km 235
               Caixa Postal 676 – Monjolinho
               13565-905 – São Carlos – SP - Brasil
E-mail: matthiasthurer@workloadcontrol.com

Name: Dr Ting Qu
Institution: Guangdong University of Technology
Address: Guangdong CIMS Provincial Key Lab
               School of Electromechanical Engineering
               Guangdong University of Technology
               Guangzhou, China.
Email: quting@gdut.edu.cn

Name: Dr Mark Stevenson
Institution: Lancaster University
Address: Department of Management Science
               Lancaster University Management School
               Lancaster University
               LA1 4YX, U.K.
E-mail: m.stevenson@lancaster.ac.uk
Tel: 00 44 1524 593847

Name: Thomas Maschek
Institution: Technical University Dortmund
Address: Mechanical Engineering Department
               Institute of Production Systems and Industrial Engineering
               Leonhard-Euler-Str. 5
               44227 Dortmund, Germany
E-mail: thomas.maschek@tu-dortmund.de

Name: Dr Moacir Godinho Filho
Institution: Federal University of São Carlos
Address: Department of Industrial Engineering
               Rodovia Washington Luís, km 235
               Caixa Postal 676 – Monjolinho
               13565-905 – São Carlos – SP - Brasil
E-mail: moacir@dep.ufscar.br

Keywords: *Continuous Order Release; Workload Control; Job Shop; Simulation.*

Continuous Workload Control Order Release Revisited:
An Assessment by Simulation

Abstract

Order release is a key component of the Workload Control concept. Jobs do not enter the shop floor directly - they are retained in a pre-shop pool and released in time to meet due dates while keeping work-in-process within limits or norms. There are two important groups of release methods: continuous methods, for which the workload falling to a specified level can trigger a release at any moment in time; and, periodic release methods, for which releases take place at fixed intervals. Continuous release methods in general have been shown to outperform periodic release methods. Yet there is incongruence in the results presented in the literature on the relative performance of the various continuous release methods. We use a job shop simulation model to examine the performance of continuous release methods from the literature and find that the contradictory results are explained by the different rules applied to sequence jobs in the pool - a factor neglected in previous work. Finally, a new breed of continuous release methods has recently emerged, but these have not been compared with prior approaches. Therefore, we also examine these methods and show that they significantly improve overall performance, although this is to the detriment of jobs with large processing times.

1. Introduction

Workload Control is a production planning and control concept developed for high-variety contexts, such as small and medium-sized make-to-order companies which often have a job shop configuration (Zäpfel & Missbauer, 1993; Stevenson *et al*., 2005; Thürer *et al*., 2013). One of the key control points within the Workload Control concept is order release, which decouples the shop floor from higher level planning. Jobs are not released onto the shop floor immediately but flow into a pre-shop pool, which buffers the shop floor against variance in the incoming order stream (Melnyk & Ragatz, 1989; Land & Gaalman, 1996; Thürer *et al*., 2012). Release methods can be categorized according to *when* the release decision takes place: either at fixed *periodic* time intervals or *continuously* at any moment in time. It has long-since been shown that continuous release methods have the potential to outperform periodic release methods (e.g. Sabuncuoglu & Karapinar, 1999; Thürer *et al*., 2012). Yet studies which have compared the various continuous order release methods presented in the literature disagree on their relative performance (e.g. Melnyk & Ragatz, 1989; Hendry & Wong, 1994; Sabuncuoglu & Karapinar, 1999).

 'Classical' continuous release methods presented in the literature are based on the re-order point methodology (Bergamaschi *et al*. 1997), whereby a new order is released whenever a workload measure falls below a predetermined level (see, e.g. Melnyk & Ragatz, 1989). The main difference between classical continuous release methods lies in the workload measure that is considered for triggering a release – either a bottleneck, work center or shop load workload measure – but simulation results from previous studies provide no clear conclusion on which measure performs the best and should therefore be applied. For example, results in Melnyk & Ragatz (1989) and Hendry & Wong (1994) indicated that an aggregate shop load measure – which triggers the release procedure when the load of the whole shop falls below a pre-determined level – outperforms a work center measure, which triggers release when the workload of any work center falls below a pre-determined level. This, however, contradicts the findings of Sabuncuoglu & Karapinar (1999), where better results were obtained when more detailed workload information was considered. Sabuncuglu & Karapinar's (1999) results are in line with the literature on periodic release methods, which has shown that performance improves as the information fed-back from the shop floor becomes more detailed (e.g. Henrich *et al*. 2004).

While, on the surface, the three studies by Melnyk & Ragatz (1989), Hendry & Wong (1994) and Sabuncuoglu & Karapinar (1999) appear to have used similar simulation models and dispatching rules – and hence may have been expected to produce similar results – closer inspection reveals differences in terms of the sequence in which jobs were considered for release from the pool. It would therefore appear that the pre-shop pool sequencing decision may explain the differing results, but none of these previous studies discussed the potential impact on performance that this decision may have. It is argued here that the choice of pre-shop pool sequencing rule may provide the key to explaining the contradictory results observed in the literature and, therefore, to improving the design and performance of order release methods in the future.

Adding to this debate, a new breed of continuous release methods have recently emerged that replace the workload trigger used in the classical methods with an upper workload bound (see Land *et al*., 2010; Fernandes & Carmo-Silva, 2011). Rather than releasing the next job in the sequence regardless of its load contribution, jobs are only released if they fit within a workload norm. This is akin to the method typically applied in periodic release methods (see, e.g. Land & Gaalman, 1996; Henrich *et al*., 2004); but, here, the release decision can be taken at any moment in time. These new methods have been introduced without comparison with the classical continuous release methods, thus making it difficult to evaluate the performance impact that this change has had. Interestingly, the new rules presented in Land *et al*. (2010) and Fernandes & Carmo-Silva (2011) use planned release dates to sequence jobs in the pool, which, again, is a difference with the sequencing rules used in previous research (e.g. Melnyk & Ragatz, 1989; Hendry & Wong, 1994). This makes even an informal, rough comparison of the results presented across papers unhelpful. From the above, it follows that there is a need to consolidate research on continuous release methods by:

1. Conducting simulations that explicitly consider classical and new continuous release methods, comparing their performance; and

2. Assessing the performance impact of the sequence in which jobs are considered for release from the pool.

The remainder of this paper is structured as follows. First, the continuous order release methods to be considered in the study are identified from the literature in Section 2. This section also includes a review of the rules used to sequence jobs in the pool that need to be included in

this study. The simulation model applied to evaluate the performance of the continuous release methods is then described in Section 3 before the results are presented, discussed and analyzed in Section 4. Finally, conclusions are drawn in Section 5.

2. Literature Review: Order Release and Rules for Sequencing Jobs in the Pool

2.1 Continuous Workload Control Order Release

There are many order release methods in the literature – for a review, see for example: Wisner (1995), Land & Gaalman (1996), Bergamaschi *et al.* (1997) or Fredendall *et al.* (2010). In this study, the focus is on continuous order release methods, i.e. release methods that may release a job onto the shop floor at any moment in time, usually initiated by an event on the shop floor. In contrast to periodic methods – i.e. release methods that take the release decision only at fixed time intervals – most continuous order release methods do not apply a workload norm (or limit); instead, a workload trigger is used. For classical continuous release methods, a critical workload is determined, which, if violated, triggers the release procedure, thereby pulling orders from the pool onto the shop floor until the critical workload is no longer violated. This may allow the next job to be selected even if its load contribution will exceed the critical load, i.e. there is no maximum workload constraint. Order release methods of this type can best be classified according to the workload used to trigger the release (bottleneck, work center or shop load), as explained below:

- *Bottleneck:* Bottleneck workload trigger methods activate the release procedure if the direct (or imminent) load of the bottleneck work center falls below a pre-determined load limit. Only the bottleneck work center is considered and only jobs which have to pass through the bottleneck are controlled by the order release method. As soon as the bottleneck load falls below the specified limit, a job is released according to the job sequencing rule applied in the pool (e.g. earliest due date first). Examples of rules based on the bottleneck load are the Starvation Avoidance (SA) method by Glassey & Resende (1988) and the Bottleneck Load Oriented Release (BLOR) method applied by Enns & Prongue-Costa (2002).
- *Work center:* Work center workload trigger methods activate the release procedure if the direct (or imminent) load of any work center falls below a predetermined load limit. Jobs in the pool for which the triggering work center is the first work center in their routing are

considered for release according to the job sequencing rule applied in the pool (e.g. earliest due date first). An example is the Work Center workload trigger Earliest Due Date (WCEDD) selection method presented by Melnyk & Ragatz (1989). A work center workload trigger was also used, for example, in Hendry & Wong (1994) and Sabuncuoglu & Karapinar (1999).

- *Shop load:* Shop load workload trigger methods activate the release procedure if the load of the whole shop floor (typically measured as the total remaining work) falls below a predetermined load limit. Jobs are released onto the shop floor in accordance with the sequencing rule applied in the pool, such as the earliest due date rule or the Work-In-Next-Queue (WINQ) rule. For WINQ, only those jobs that have the work center with the smallest queue as the first work center in their routing are considered. Examples of this type of order release method are the Aggregate workload trigger Work-in-Next-Queue (AGGWNQ) method presented by Melnyk & Ragatz (1989) and the WIPLoad control method applied by Qi *et al*. (2009). An aggregated workload trigger based on Melnyk & Ragatz's (1989) AGGWNQ rule was also used in Hendry & Wong (1994) and Sabuncuoglu & Karapinar (1999).

Many classical continuous release methods exist in the literature – as described above – but they all use a workload trigger based on the re-order point methodology and they largely differ in terms of the detail of the workload measure used to trigger release. More recently, however, Land *et al*. (2010) and Fernandes & Carmo-Silva (2011) extended the set of available continuous release methods by introducing three new rules. The first uses a classical work center workload trigger but measures the workload used to trigger release differently, based on theory from the wider Workload Control literature. The second replaces the classical workload trigger altogether with an explicit upper workload bound. Finally, the third also replaces the classical workload trigger with an explicit upper workload bound, but relaxes this to better balance workloads across resources. These three methods are summarized in sections 2.1.1 to 2.1.3, respectively.

2.1.1 Work Center Workload Trigger Method based on a New Measure of the Workload
Fernandes & Carmo-Silva (2011) refined the classical work center workload trigger by combining it with the corrected aggregate load method defined in Oosterman *et al*. (2000). For the corrected aggregate load approach, the processing time of each operation in the routing of a

job contributes to the workload measure of the corresponding work center(s) from release until the operation is complete. To find the corrected aggregate load, the load contribution is converted (or corrected) by dividing the processing time by the position of a work center in the routing of the job. A job becomes part of the imminent load of the first work center in its routing as soon as it is released, and the full contribution lasts until completion at this work center. For the second work center, for example, the job again contributes to the workload but the conversion compensates for the fact that it will only be part of the imminent load of the second work center for, roughly speaking, 50% of the time that it contributes by dividing the workload by the position of the work center in the routing of the job (two). For the third work center in the routing of a job, only one third of its load contributes to the work center at release, and so on.

2.1.2 Alternative Use of an Upper Workload Bound

As discussed above, classical continuous release methods do not apply an explicit upper workload bound. For this second new rule, both Land *et al*. (2010) and Fernandes & Carmo-Silva (2011) introduced an upper workload bound based on the release mechanism that is typically applied under periodic release (see, e.g. Oosterman *et al*., 2000; Henrich *et al*., 2004). The release procedure is triggered whenever an operation is completed or a new order arrives at the pre-shop pool. At this point, all jobs in the pool are considered for release once according to the job sequencing rule applied in the pool (e.g. earliest due date first). The workload of each operation in the routing of a job currently being considered for release contributes to the load of the corresponding work centers. The load contribution is in accordance with the corrected aggregate load approach – i.e. as above, the processing time of the operation is divided by the position of a work center in the routing of the job (see Section 2.1.1). The new workload of each work center is compared against the predetermined workload limits or norms used to control queue lengths on the shop floor and, if one or more of these norms are violated, the job is retained in the pre-shop pool. If, however, the norms are not violated, the job is selected for release onto the shop floor and its workload is formally assigned to the workloads of the work centers in its routing. These steps are repeated until all jobs in the pool have been considered for release once. Note that this release method is similar to the standard periodic release procedure applied in the Workload Control literature but it uses a very small period between periodic releases, which means it technically approaches a continuous release procedure. The impact of the time interval between periodic releases on performance has been assessed, e.g. by Perona &

Portioli (1998) and Land (2006). Both showed that a shorter period between releases yields a lower percentage of tardy jobs, but that this is offset by an increase in the standard deviation of lateness and tardiness.

2.1.3 Relaxed Upper Bound to Balance Workloads across Resources
Authors such as Land & Gaalman (1998) have previously noted that a rigid upper workload bound can be criticized for hindering the release of an order which significantly contributes to an under-loaded work center, thereby avoiding starvation, if it leads to even a very small overload at another work center in its routing. Indeed, authors such as Cigolini & Portioli-Staudacher (2002) and Fernandes & Carmo-Silva (2011) have since argued against the application of rigid norms and introduced a release method called "workload balancing". This method is similar to the use of an upper workload bound, but the release of an order is allowed even if it violates a norm at one or more work centers as long as the overload is smaller than the under-load that is reduced at other work centers. Fernandes & Carmo-Silva (2011) found that this release method can achieve significant improvements in percentage tardy and lead time performance compared to the application of a rigid upper workload bound, as described in Section 2.1.2 above.

While these three new release methods have significantly changed the continuous release procedure, their performance impact compared to classical continuous release methods has not been evaluated. In response, this study uses a job shop simulation model to compare classical and new continuous release methods. All types of continuous release methods identified from the literature are considered except bottleneck workload trigger methods. As in Hendry & Wong (1994) and Sabuncuoglu & Karapinar (1999), we do not consider bottleneck workload trigger methods because the constraints in the pure job shop environment are constantly shifting; hence, no specific, fixed bottlenecks can be identified. Next, we will review the rules used to sequence jobs in the pool which determine the order in which they are considered for release – a potentially important explanatory factor overlooked in the literature.

2.2 Rules for Sequencing Jobs in the Pool from the Literature
The following five sequencing rules can be identified from the literature:

- *First-Comes-First-Served (FCFS)*, which sequences jobs according to their time of arrival in the pool. This rule was applied, e.g. by Park & Salegna (1995), Sabuncuoglu & Karapinar (1999) and Enns & Prongue-Costa (2002).

- *Earliest Due Date (EDD)*, which sequences jobs according to their due date. This rule was applied, e.g. by Ragatz & Mabert (1988), Melnyk & Ragatz (1989) and Hendry & Wong (1994).

- *Planned Release Date (PRD)*, which sequences jobs according to planned release dates given by Equation (1) below. This rule was applied, e.g. by Land & Gaalman (1998), Fernandes & Carmo-Silva (2011) and Thürer *et al.* (2012). Two variants have been used in the literature, where it is either assumed that waiting times or operation throughput times are controlled.

$$\tau_j = \delta_j - \sum_{i \in R_j}\left(a_i + p_{ij}\right) \text{ or } \tau_j = \delta_j - \sum_{i \in R_j}b_i \tag{1}$$

τ_j= planned release date of job j

δ_j = due date of job j

R_j = the ordered set of operations in the routing of job j

p_{ij} = the processing time at the ith operation in the routing of a job

a_i = constant for estimated waiting time at the ith operation in the routing of a job

b_i = constant for estimated throughput time at the ith operation in the routing of a job

- *Shortest Processing Time (SPT)*, which sequences jobs according to their processing time at the first work center in their routing. This rule has been applied e.g. by Hendry & Wong (1994).

- *Lowest Workload Imbalance (LWIB)*, which sequences jobs according to the lowest workload imbalance index (B_j), as given by Equation (2) below. This rule was applied by van Ooijen (1998).

$$B_j = \sum_{i \in R_j}\left|N_k - W_k\right| \tag{2}$$

N_k = target workload level at work center k performing the ith operation in the routing of a job

W_k = workload measure at work center k performing the ith operation in the routing of a job

While a broad set of pre-shop pool sequencing rules has been applied in the literature, the impact of the choice of sequencing rule has not been examined. This complicates the interpretation of results across studies and may provide an explanation for the contradictory results observed in previous studies that have compared continuous release methods, such as the contributions by Melnyk & Ragatz (1989), Hendry & Wong (1994) and Sabuncuoglu & Karapinar (1999). In response, all five of the sequencing rules identified from the literature are considered in this study and their impact on the performance of five continuous release methods is examined.

3. Simulation Model

This research started by asking: *What is the performance impact of continuous Workload Control order release methods?* In response, simulation was used to examine the performance impact of five continuous release methods identified from the literature. For the first time, classical continuous release methods and three new methods are compared. In addition, the effect of five different rules for sequencing jobs in the pool – a factor neglected in previous research, but which may explain differences in performance results across studies – is considered.

The shop and job characteristics modeled in the simulation are outlined in Section 3.1 before the operationalization of the order release methods and sequencing rules applied are summarized in Section 3.2. The shop floor dispatching rules used for controlling the progress of orders on the shop floor is then described in Section 3.3. Finally, the experimental design is outlined and the measures used to evaluate performance are presented in Section 3.4.

3.1 Overview of Modeled Shop and Job Characteristics

To allow for comparison, the simulation model used in this study is based on the model typically used in previous research on Workload Control and order release, e.g. Melnyk & Ragatz (1989), Hendry & Wong (1994), and Land *et al*. (2010). Hence, a simulation model of a randomly routed job shop (Conway *et al*., 1967) – later referred to as a pure job shop (Melnyk & Ragatz, 1989) – has been implemented in Python$^{©}$ using the SimPy$^{©}$ module. The shop contains six work centers, where each work center is a single constant capacity resource. The routing length of jobs varies uniformly from one to six operations. All work centers have an equal probability of being visited and a particular work center is required at most once in the routing of a job.

Operation processing times follow a truncated 2-Erlang distribution with a truncated mean of 1 time unit and a maximum of 4 time units. The inter-arrival time of orders follows an exponential distribution with a mean of 0.648, which – based on the average number of work centers in the routing of a job – deliberately results in a utilization level of 90%. Due dates are set exogenously by adding a random allowance factor, uniformly distributed between 30 and 50 time units, to the job entry time. The minimum value will be sufficient to cover a minimum shop floor throughput time corresponding to the maximum processing time (4 time units) for the maximum number of possible operations (6) plus an arbitrarily set allowance for the waiting or queuing times of 6 time units. Finally, Table 1 summarizes the simulated shop and job characteristics.

[Take in Table 1]

3.2 Order Release and its Alternative Pre-Shop Pool Sequencing Rules

As in previous simulation studies on Workload Control (e.g. Land & Gaalman, 1998; Fredendall *et al*., 2010; Thürer *et al*., 2012), it is assumed that all jobs are accepted, materials are available and all necessary information regarding shop floor routing, processing times, etc is known. Jobs flow into a pre-shop pool to await release according to five release methods: (i) the Classical Aggregate (shop) Workload Trigger work-in-next-queue selection (AGGWLT); (ii) the Classical Work Center Workload Trigger (WCWLT); (iii) the Corrected Workload Trigger (CorrWLT); (iv) the Upper Bound Release (UBR); and (v) the Upper Bound Release with Load Balancing (UBRLB). Remember that the bottleneck workload trigger method has not been considered as we focus on a pure job shop environment in which bottlenecks are constantly shifting. As a baseline for comparison, experiments without controlled release have also been executed, i.e. where jobs are released onto the shop floor IMMediately (IMM) upon arrival.

More specifically for the three workload trigger methods – AGGWLT, WCWLT and CorrWLT – the release procedure will be summarized in Section 3.2.1, before the release procedure for the upper workload bound methods – UBR and UBRLB – is detailed in Section 3.2.2. Performance results for the release methods are obtained by tightening the workload trigger (or norm) stepwise; six levels are considered, as specified in Table 2. Table 2 summarizes the five release methods and the five pool sequencing rules – First-Comes-First-Served (FCFS), Earliest Due Date (EDD), Planned Release Date (PRD), Lowest Workload Imbalance (LWIB)

and Shortest Processing Time (SPT) – considered in this study. The parameters for the sequencing rules are also specified in Table 2 and were identified via preliminarily simulation experiments as resulting in the best overall performance.

[Take in Table 2]

3.2.1. Continuous Release Methods using a Workload Trigger

The following release procedure takes place whenever the workload measure falls below the trigger level:

(1) For WCWLT, the direct (or imminent) workload at the triggering work center s (W_s^I) is calculated.

(2) According to the particular sequencing rule being applied, a priority value is determined for each job in the set of jobs J in the pool that, in the case of AGGWLT, is based on having the work center with the lowest direct load in their routing; and, for WCWLT and CorrWLT, on having the triggering work center as the first in their routing.

(3) The job $j \in J$ with the highest priority is considered for release first.

(4) Job j is released and,

for AGGWLT, its total processing time contributes to the current workload of the whole shop (W),

that is $W := W + \sum_{i \in R_j} p_{ij}$

for WCWLT, its processing time p_{1j} at the 1st operation in its routing is added to the imminent workload at the triggering work center,

that is $W_s^I := W_s^I + p_{1j}$.

for CorrWLT, its processing time p_{ij} at the ith operation in its routing R_j – corrected for the operation number i – contributes to the current workload at the work center k corresponding to operation i,

that is $W_k := W_k + \dfrac{p_{ij}}{i} \quad \forall i \in R_j$

(5) If the set of jobs in the pool J contains any jobs that have not yet been considered for release and the workload measure is smaller than or equal to the trigger level (N),

that is $W \le N$ for AGGWLT, and $W_s^I \le N_s$ for WCWLT and $W_s \le N_s$ for CorrWLT

then return to Step 3 and consider the job with the next highest priority. Otherwise, the release procedure is complete and the selected jobs are released.

3.2.2 Continuous Release Methods using an Upper Workload Bound

The following release procedure is triggered whenever a new job arrives at the pre-shop pool or an operation is completed:

(1) According to the sequencing rule applied, a priority value is determined for each job in the set of jobs J in the pool.

(2) The job $j \in J$ with the highest priority is considered for release first.

(3) If, for all the operations in job j's routing R_j, processing time p_{ij} at the ith operation – corrected for the operation number i – together with the current workload at the work center k corresponding to operation i (W_k) fits within the workload norm at this work center (N_k),

that is $\dfrac{p_{ij}}{i} + W_k \le N_k \quad \forall i \in R_j$

then the job is selected for release, i.e. removed from J, and its load contribution is included,

that is $W_k := W_k + \dfrac{p_{ij}}{i} \quad \forall i \in R_j$

Otherwise: (i) for UBR, the job remains in the pool and its load contribution is reset to zero; and, (ii) for UBRLB, the job is released even if a norm is violated, as long as the sum of the contribution to under-loaded work centers L_k^U, is greater than or equal to the sum of the overload L_k^O caused,

that is $\displaystyle\sum_{i \in R_j} L_k^U - L_k^O \ge 0$ with

$L_k^U = \dfrac{p_{ij}}{i}; \; L_k^O = 0 \text{ for } (\dfrac{p_{ij}}{i} + W_k) - N_k \le 0$

$L_k^U = 0; \; L_k^O = (\dfrac{p_{ij}}{i} + W_k) - N_k \text{ for } (\dfrac{p_{ij}}{i} + W_k) - N_k > 0$

(4) If the set of jobs in the pool *J* contains any jobs that have not yet been considered for release, then return to Step 2 and consider the job with the next highest priority. Otherwise, the release procedure is complete and the selected jobs are released.

3.3 Shop Floor Dispatching

Hendry & Wong (1994) and Sabuncuoglu & Karapinar (1999) compared the performance of the AGGWLT and WCWLT release methods and found differences in their performance to be independent of which dispatching rule was applied on the shop floor. However, in Melnyk & Ragatz (1989), performance was heavily dependent on which rule was applied on the shop floor. When dispatching was based on some measure of the urgency of jobs, WCWLT performed the best, but when dispatching was based on the job with the shortest processing time, AGGWLT performed the best. This result means that two shop floor dispatching rules must be considered in this study. Dispatching, therefore, either follows operation due dates – i.e. the job with the earliest operation due date from the set of jobs queuing in front of a work center is processed first – or follows shortest processing times. For dispatching based on operation due dates, operation due dates are determined when a job is released by distributing the available slack – i.e. the due date of job *j* (δ_j) minus its release date (t^r_j) – over the operations in its routing in accordance with Equation (3) below. This procedure is based on Land *et al.* (2013) and is especially suitable when order release control is applied as it takes deviations from the schedule caused by order release into account.

$$\text{if } \left(\delta_j - t^r_j\right) \geq 0;\ \delta_{ji} = t^r_j + i \cdot \frac{\left(\delta_j - t^r_j\right)}{n_j} \qquad \forall i \in R_j \qquad (3)$$

$$\text{else if } \left(\delta_j - t^r_j\right) < 0;\ \delta_{ji} = t^r_j \qquad \forall i \in R_j$$

δ_{ji} = operation due date of job *j* at the ith operation in its routing

3.4 Experimental Design and Performance Measures

The experimental factors are: (i) 6 levels of the workload trigger (or norm) for the Classical Aggregate Workload Trigger (AGGWLT), the Classical Work Center Workload Trigger (WCWLT), the Corrected Workload Trigger (CorrWLT), the Upper Bound Release (UBR) and Upper Bound Release with Load Balancing (UBRLB); (ii) the five different pool sequencing rules, i.e. First-Comes-First-Served (FCFS), Earliest Due Date (EDD), Planned Release Date

(PRD), Lowest Workload Imbalance (LWIB) and Shortest Processing Time (SPT); and (iii) the two different dispatching rules, i.e. dispatching based on operation due dates and on shortest processing times. A full factorial design was used with 300 cells, where each cell was replicated 100 times. Results were collected over 10,000 time units following a warm-up period of 3,000 time units. These parameters allowed us to obtain stable results while keeping the simulation run time to a reasonable level.

The principal performance measures considered in this study are as follows:

- *Mean throughput time:* The mean of the completion date minus the release date across jobs; this indicates the mean time that a job spends on the shop floor.
- *Mean lead time:* The mean of the completion date minus the pool entry date across jobs; this indicates the mean time that a job spends in the system (i.e. the time a job spends in the pre-shop pool plus its throughput time).
- *Percentage tardy:* The percentage of jobs completed after the due date; this indicates the proportion of jobs that would be delivered tardy to the customer.
- *Mean tardiness:* The conditional mean of the tardiness (T_j); this is the extent to which jobs are tardy (see Equation (4) below).

$$T_j = \max(0, L_j) \tag{4}$$

L_j = lateness (i.e. the actual delivery date minus the due date) of job j

4. Results

To give a first indication of the relative impact of the experimental factors, statistical analysis has been conducted by applying ANOVA. ANOVA is here based on a block design, where the release method is the blocking factor. Thus, statistical analysis is restricted to the main effects of the release method, as each release method may be considered a different system. The block design allows the main effect of the release method and the interaction effects of the rules used to sequence jobs in the pool and dispatch jobs on the shop floor to be captured. A linear model has been implemented with the independent variables modeled as dummies. All three factors – the release method, sequencing rule and dispatching rule – are shown to be significant, as are the

15

two-way interactions between the pre-shop pool sequencing and dispatching rules for all performance measures, except the mean tardiness (see Table 3).

[Take in Table 3]

Although prior literature has neglected the sequence in which jobs are considered for release, our statistical analysis suggests that this factor has a significant impact on performance. The Scheffé multiple-comparison test has been used to further prove the significance of the differences between the outcomes of the individual rules used to sequence jobs in the pool identified from the literature. These tests found significant differences for all the rules for at least one performance measure, except for FCFS and EDD which were statistically equivalent (see Table 4). Detailed performance results are presented next in Section 4.1 for dispatching based on operation due dates and on shortest processing times before the underlying causes of the performance differences observed are examined in more detail in Section 4.2.

[Take in Table 4]

4.1 Performance Assessment

4.1.1 Performance Assessment for Dispatching based on Operation Due Dates

Table 5 summarizes the performance results for the five release methods and the five sequencing rules for dispatching based on operation due dates. In addition, the results obtained when orders are released immediately (IMM) are also included as a benchmark and represent the outcome with no order release control, i.e. when control is only exercised through the shop floor dispatching rule.

[Take in Table 5]

First, it can be observed that the AGGWLT release method performs worse than WCWLT under all tested conditions; this contradicts Hendry & Wong (1994) who found the opposite. Examining the results across the different rules used to sequence jobs in the pool reveals only minor performance differences for rules that use some measure of urgency (i.e. FCFS, EDD and PRD). Yet, all three are outperformed in terms of percentage tardy and mean tardiness by SPT. This may explain why our results differ from Hendry & Wong's (1994), as they applied SPT sequencing for AGGWLT and applied EDD sequencing for the WCWLT release method. Hence,

16

they either chose to ignore the impact that this factor would have on performance or assumed that it would not have an effect at all. In contrast, Sabuncuoglu & Karapinar (1999) applied the FCFS rule to both AGGWLT and WCWLT and yielded similar results to those in Table 5.

Second, while the WCWLT release method achieves the highest reduction in throughput times, it is outperformed by both of the new upper workload bound release methods, i.e. UBR and UBRLB, in terms of the percentage tardy and mean tardiness results. In fact, UBR and UBRLB allow performance in terms of all four performance measures shown in Table 5 to be significantly enhanced compared to IMM providing that the workload norms are set appropriately. Unlike classical release methods based on a workload trigger only, UBR and UBRLB contribute to improved workload balancing, which results in a shorter pool waiting time (the lead time minus the throughput time) and shorter lead times. But while the results for UBR and UBRLB are compelling, refining classical release to consider the corrected aggregate load (WCWLT compared to CorrWLT) rather than the direct (or imminent) workload yields only marginal improvements in tardiness performance. Moreover, it introduces significant sensitivity to tight workload norms – this draws into questions its utility for implementation in practice. Under CorrWLT, a job on the shop floor contributes to all of the work centers in its routing, thus a work center may run idle with no direct load because of the distorting effect of the load contribution of jobs that are still upstream and yet to arrive at the work center. Finally, allowing the release of jobs that violate the upper workload bound providing they contribute to workload balancing (UBRLB) does not yield a significant reduction in the percentage tardy and actually increases the mean tardiness compared to UBR. Therefore, and in contrast to Fernandes & Carmo-Silva (2011), we cannot confirm from our results for UBR *vs.* UBRLB that significant performance improvements result from relaxing the rigid upper bound to allow for workload balancing across resources. Rather, we confirm the results in Cigolini & Portioli-Staudacher (2002) on the negative impact of workload balancing on tardiness measures, and this will be explored further in Section 4.2 below.

In terms of the rules used to sequence jobs in the pool, it can be observed that FCFS, EDD and PRD are outperformed by SPT in terms of the percentage tardy and mean tardiness measures if a workload trigger is applied (i.e. if either the AGGWLT, WCWLT or CorrWLT release method is used). Under the UBR and UBRLB release methods, SPT yields a lower percentage tardy, but this is offset by an increase in mean tardiness compared to rules that use some measure

of urgency (i.e. FCFS, EDD and PRD). The performance of the LWIB pre-shop pool sequencing rule is particularly interesting: this rule performs the worst across all of the rules used to sequence jobs in the pool if a workload trigger is applied (i.e. the AGGWLT, WCWLT and CorrWLT release methods). This is because LWIB treats overload and under-load equally; for example, a job that causes two time units of overload is prioritized over a job that reduces the under-load by one time unit from four to three. This problem does not occur if an upper workload bound is used; hence, for UBR and UBRLB, the LWIB sequencing rule performs similar to the other rules or, in some cases, even improves performance. In general, the performance differences across sequencing rules are negligible for the UBR and UBRLB release methods. While under a workload trigger method, the next job in the sequence is always selected for release, under UBR and UBRLB, another constraint is introduced through the upper workload bound, which may hinder the release of a job with high priority if it does not fit within the workload norm. Thus, the restriction that jobs must fit within the workload norm ultimately determines which jobs are chosen for release more so than the sequence in which the jobs are considered for release.

4.1.2 Performance Assessment for Dispatching based on Shortest Processing Times
The performance results for shortest processing time dispatching, as summarized in Table 6, confirm our conclusions on the performance of release methods and rules used to sequence jobs in the pool. Focusing on the results without order release control (i.e. IMM), we can observe the expected significant improvement in both throughput time and percentage tardy performance; but we can also observe that this is achieved at the expense of mean tardiness performance. IMM performs the best in our experiments in terms of percentage tardy performance, and this fundamentally questions the use of order release control *if* dispatching is based on shortest processing times.

As might have been expected, dispatching based on shortest processing times significantly reduces throughput times compared to dispatching based on operation due dates. This effect on throughput times is the strongest for LWIB. For LWIB, also a significant reduction in percentage tardy performance can be observed for the three release methods that apply a workload trigger (AGGWLT, WCWLT and CorrWLT) - meanwhile, tardiness performance remains similar for FCFS, EDD, PRD and the SPT pool sequencing rule. This largely explains the significant two-way interactions observed in Table 3. For the two workload bound release methods, both

18

percentage tardy and mean tardiness performance deteriorate compared to dispatching based on operation due dates. This is likely to be because it can be difficult to release large jobs under both UBR and UBRLB, which increases the likelihood of large jobs becoming tardy; this will be examined in detail in Section 4.2. The risk of becoming tardy is further increased if the shop floor dispatching rule also prioritizes jobs with short processing times. In general, the best performance in terms of percentage tardy and mean tardiness was obtained for dispatching based on operation due dates combined with the UBR release method (see Table 5). Therefore, our results argue for the use of dispatching based on operation due dates rather than based on shortest processing times.

[Take in Table 6]

Summarizing from the above, five important conclusions can be drawn:

- Continuous Workload Control order release methods should be combined with shop floor dispatching based on operation due dates rather than based on shortest processing times; this is because dispatching based on operation due dates leads to the best tardiness performance.
- The WCWLT release method outperforms AGGWLT in all of our experiments.
- Contradictory performance results observed in previous research may partly be explained by the different rules used to sequence jobs in the pre-shop pool and hence to determine the order in which they are considered for release.
- The LWIB pre-shop pool sequencing rule, as introduced by van Ooijen (1998), should not be applied if a workload trigger is used (as, e.g. in van Ooijen, 1998) as this may significantly deteriorate performance. However, it is a viable alternative if workload bound methods are used.
- The new release methods introduced by Land *et al*. (2010) and Fernandes & Carmo-Silva (2011) significantly enhance performance compared to classical release methods. However, looking at the average results aggregated for all jobs may hide significant performance differences across job sizes, as observed by Perona & Portioli (1998) and Land (2006) for periodic release methods if the time between releases is shortened. This will be examined more closely in the next section.

4.2 Performance Analysis

Following Land (2006), improvements in tardiness performance may be the result of a reduced lead time (i.e. the time from job entry to job completion) and/or a reduced standard deviation of lateness. Figure 1 illustrates the lateness distribution for the five different release methods and, as a benchmark, for IMM. In Figure 1a and 1b, shop floor dispatching is based on operation due dates; in Figure 1a, jobs in the pool are sequenced according to PRD and, in Figure 1b, according to SPT. In Figure 1c and 1d, shop floor dispatching is based on shortest processing times; in Figure 1c, jobs in the pool are sequenced according to PRD and, in Figure 1d, according to SPT. PRD was chosen as the best-performing time-based rule for sequencing jobs in the pool, while SPT as the rule that bases the sequence entirely on the workload of jobs. Results are given for the trigger or norm level that resulted in the best overall performance; the trigger or norm level applied is given in the legend of Figure 1.

[Take in Figure 1]

From Figure 1a and 1c – where jobs in the pool are sequenced according to PRD – it can be seen that UBR and UBRLB gain their advantage over alternative (workload trigger) release methods through lead time *and* standard deviation reduction. While the latter is more apparent under operation due date oriented dispatching (Figure 1a), the former is more apparent under dispatching according to shortest processing times (Figure 1c). Under PRD sequencing, the pool waiting time is much lower for UBR and UBRLB than it is for alternative release methods, as will be seen below. This shortens the overall lead time while, at the same time, giving more control to the dispatching rule on the shop floor. A further effect has to be considered if dispatching follows operation due dates. Here, UBR and UBRLB gain their advantage through SPT effects in periods of high load (Land *et al*., 2010), i.e. when the capacity gap available to be filled by new orders is typically small. Thus, only jobs with small processing times tend to be selected for release in high load periods. This effect is the strongest at the first work center in the routing of a job as, here, the complete workload is considered. At downstream operations, the workload is depreciated through the corrected aggregate load approach. Thus, UBR and UBRLB can be compared to a workload trigger based release method that sequences jobs in the pool according to the shortest processing time. Consequently, by comparing Figure 1a and 1c with

20

Figure 1b and 1d, it can be seen that the advantage of UBR and UBRLB partially disappears if jobs in the pool are considered for release in SPT sequence.

The above also implies that a job with large processing times is retained in the pre-shop pool even if it is urgent. To further examine the individual performance effects of each release method, we subdivided jobs according to two dimensions of job size: routing length and average processing time across operations (see Table 7). For the Routing Length (RL), we considered six classes (RL1, RL2 to RL6). For the average processing time, we spit jobs arbitrarily into: small jobs, with an average processing time of less than 0.6 time units; large jobs, with an average processing time of more than 1.4 time units; and, medium jobs, referring to the remaining jobs with a processing time of at least 0.6 time units but no more than 1.4 time units.

[Take in Table 7]

Table 8 summarizes the results for the time to release – i.e. the time that a job has to spend in the pool prior to release – for each class of routing length and average processing time (see Table 7). Results are only shown for dispatching based on operation due dates and for the PRD sequencing rule. The PRD sequencing rule was chosen as it is the best-performing rule for sequencing jobs in the pool according to a time-based (or urgency-based) measure and because performance differences between the release methods are stronger compared to SPT, which aids analysis.

[Take in Table 8]

First, from the results in Table 8 it can be observed that, under PRD sequencing, jobs with longer routings receive higher priority and consequently have to wait less time in the pool compared to under FCFS or EDD sequencing where there is no difference in the time to release across classes (results are not shown). This explains the performance difference observed for PRD *vs.* FCFS or EDD; the performance results for FCFS and EDD are statistically equivalent. Further, performance differences between small, medium and large jobs for the AGGWLT, WCWLT and CorrWLT release methods can be explained by the inclusion of processing times in the PRD calculation.

Second, results further confirm that the performance improvements for UBR over classical methods like WCWLT can be attributed to the performance of small and medium sized jobs.

These jobs are released to the shop floor faster, while the time that large jobs spend in the pool waiting to be released increases. This is especially evident if norms are tight. As large jobs are also the ones that require the longest time on the shop floor (especially when the routing length is long), there is a greater risk of them becoming tardy. Yet this negative performance effect is mitigated compared to the results in Perona & Portioli (1998) and Land (2006) for short intervals between periodic releases as the corrected aggregate load approach is applied for workload accounting over time. This depreciates the load contribution at downstream work centers and favors the release of large jobs.

Finally, under UBRLB, jobs with long routings may gain an advantage over jobs with short routings compared to UBR as overloads caused at downstream operations do not hinder their release providing that the overload caused does not exceed the under-load that is relieved. Nonetheless, this improvement is at the expense of jobs with short routings and does not result in any significant overall performance improvements.

5. Conclusion

Order release control is one of the key decision levels in Workload Control. One important group of release methods are continuous release methods – i.e. release methods that take the release decision at any moment in time. This class of release methods has been shown to consistently outperform periodic release methods in the literature (e.g. Sabuncuoglu & Karapinar, 1999; Thürer *et al.*, 2012). The main difference between the 'classical' continuous release methods lies in the workload measure applied for triggering releases, but the results of previous simulations provided no clear conclusions on which measure to apply (see, e.g. Hendry & Wong, 1994; Sabuncuoglu & Karapinar, 1999). This study has shown that triggering the release decision based on the workload of each work center outperforms an aggregate measure where the load of the shop as a whole is used to trigger release. The results of some previous studies had suggested an aggregated measure was a better solution (e.g. Hendry & Wong, 1994), but we can now conclude that this was because of the way in which jobs were sequenced or prioritized for release from the pool. In general, it has been shown that the sequence in which jobs are considered for release has a significant impact on performance and should not be overlooked when comparing results – it is an important explanatory factor.

This study has also compared the performance of classical methods with new continuous release methods only recently presented in the literature (from Land *et al.*, 2010; Fernandes & Carmo-Silva, 2011). Two of these new continuous release methods introduce an upper workload bound: rather than releasing the next job in sequence, regardless of its load contribution, jobs are only released if they fit within a workload norm. In one of the rules, this was a strict limit, while the other rule relaxed the bounding procedure if release improved the workload balance across resources. Our results have shown that these new methods lead to significant performance improvements over classical continuous release methods and over immediate release if workload norms are set appropriately. However, the ultimate decision on which rule to apply in practice depends on a trade-off between performance and sophistication. Using a classical work center workload trigger with an appropriate rule for sequencing jobs in the pool may be a viable alternative for practice because its release procedure is arguably less complex: jobs can simply be pulled from the pool in accordance with the sequence when a signal is received that capacity is available at one of the work centers on the shop floor without the need for any further considerations.

While introducing an upper workload bound significantly enhanced the performance of continuous release methods, future research should find better ways of balancing performance improvements across jobs. One way of achieving this may be to develop new rules for sequencing jobs in the pool. Indeed, we have shown that considering the sequence in which jobs are released has the potential to further enhance the performance of order release. Finally, the main limitation of this study is that we neglected important environmental factors – like routing characteristics – which were shown to have a significant impact on the performance of continuous release methods in previous research (e.g. Thürer *et al.*, 2012). We restricted the experimental setting to the pure job shop (undirected routing) so as to keep the focus on the different release methods and rules for sequencing jobs in the pool. Therefore, future research should examine the performance of continuous workload control release methods in a broader environmental setting that includes, for example, different routing characteristics which may introduce fixed bottlenecks and allow bottleneck workload trigger release methods to be compared with the other workload trigger methods and with the new upper workload bounding approaches.

Acknowledgments

References

Bergamaschi, D., Cigolini, R., Perona, M., and Portioli, A., 1997, Order review and release strategies in a job shop environment: A review and a classification, *International Journal of Production Research*, 35, 2, 399 – 420.

Cigolini, R., and Portioli-Staudacher, A., 2002, An experimental investigation on workload limiting methods with ORR policies in a job shop environment, *Production Planning and Control*, 13, 7, 602– 613.

Conway, R., Maxwell, W.L., and Miller, L.W., 1967, *Theory of Scheduling*, Reading, MA: Addisson-Wesley.

Enns, S.T., and Prongue Costa, M., 2002, The effectiveness of input control based on aggregate versus bottleneck workloads, *Production Planning and Control*, 13, 7, 614 - 624.

Fernandes, N.O., and Carmo-Silva, S., 2011, Workload Control under continuous order release, *International Journal of Production Economics,* 131, 257 – 262.

Fredendall, L.D., Ojha, D., and Patterson, J.W., 2010, Concerning the theory of workload control, *European Journal of Operational Research*, 201, 1, 99 – 111.

Glassey, C.R., and Resende, M.G., 1988, Closed-loop job release control for VLSI circuit manufacturing, *IEEE Transactions on Semiconductor Manufacturing*, 1, 36 – 46.

Hendry, L.C., and Wong, S.K., 1994, Alternative order release mechanisms: A comparison by simulation, *International Journal of Production Research*, 32, 12, 2827 – 2842.

Henrich, P., Land, M.J., Gaalman, G.J.C., and Van der Zee, D-J., 2004, Reducing feedback requirements of workload control, *International Journal of Production Research*, 42, 24, 5235 – 5252.

Land, M.J., 2006, Parameters and sensitivity in workload control, *International Journal of Production Economics*, 104, 2, 625 – 638.

Land, M.J., and Gaalman, G.J.C., 1998, The performance of workload control concepts in job shops: Improving the release method, *International Journal of Production Economics*, 56 – 57, 347 – 364.

Land, M.J., and Gaalman, G.J.C, 1996, Workload control concepts in job shops: A critical assessment, *International Journal of Production Economics*, 46 – 47, 535 – 538.

Land, M.J., Su, N.P.B. and Gaalman, G..J.C, 2010, In search of the key to delivery improvement, *16th International Working Seminar on Production Economics, 1st – 5th March, Innsbruck, Austria, Conference Proceedings, 2, 297-308.*

Land, M.J., Stevenson, M., and Thürer, M., 2013, Integrating load-based order release and priority dispatching, *International Journal of Production Research*, (in print).

Melnyk, S.A., and Ragatz, G.L., 1989, Order review/release: research issues and perspectives, *International Journal of Production Research*, 27, 7, 1081 – 1096.

Oosterman, B., Land, M.J., and Gaalman, G., 2000, The influence of shop characteristics on workload control, *International Journal of Production Economics*, 68, 1, 107 – 119.

Park, P.S., and Salegna, G.J., 1995, Load smoothing with feedback in a bottleneck job shop, *International Journal of Production Research*, 33, 6, 1549 – 1568.

Perona, M., and Portioli, A., 1998, The impact of parameter setting in load oriented manufacturing control, *International Journal of Production Economics*, 55, 133 – 142.

Qi, C., Sivakumar, A.I., and Gershwin S.B., 2009, An efficient new job release control methodology, *International Journal of Production Research*, 47, 3, 703 – 731.

Ragatz, G. L., and Mabert, V. A., 1988, An evaluation of order release mechanisms in a job shop environment, *Decision Sciences*, 19, 167 – 189.

Sabuncuoglu, I., and Karapinar, H.Y., 1999, Analysis of order review/release problems in production systems, *International Journal of Production Economics*, 62, 259 - 279.

Stevenson, M., Hendry, L.C., and Kingsman, B.G., 2005, A review of production planning and control: The applicability of key concepts to the make to order industry, *International Journal of Production Research*, 43, 5, 869 - 898.

Thürer, M., Stevenson, M., Silva, C, Land, M.J., Fredendall, L.D., and Melnyk, S.A., 2013, Lean Control for Make-to-Order Companies: Integrating Customer Enquiry Management and Order Release, *Production & Operations Management*, (Article in Press).

Thürer, M., Stevenson, M., Silva, C, Land, M.J., and Fredendall, L.D., 2012, Workload control (WLC) and order release: A lean solution for make-to-order companies, *Production & Operations Management*, 21, 5, 939 – 953.

Wisner, J. D., 1995, A review of the order release policy research, *International Journal of Operations and Production Management*, 15, 6, 25 – 40.

Van Ooijen, H.P.G., 1998, Delivery performance improvement by controlled work-order release and work-center load balancing, *International Journal of Production Economics*, 56 – 57, 661 – 675.

Zäpfel, G. and Missbauer, H., 1993, New concepts for production planning and control, *European Journal of Operational Research*, 67, 297 – 320.

Tables & Figures

Table 1: Summary of Simulated Shop and Job Characteristics

Table 2: Summary of the Release Methods and Sequencing Rules Applied in this Study

Table 3: ANOVA Results

Table 4: Results for Scheffé Multiple Comparison Procedure

Table 5: Comparison of Performance Results under Operation Due Date Dispatching

Table 6: Comparison of Performance Results under Shortest Processing Time Dispatching

Table 7: Job Classes: Frequency Distribution for Routing Lengths 1 to 6

Table 8: Performance Analysis of Job Size Classes for PRD Sequencing (under Operation Due Date Dispatching): Time to Release (in Time Units)

Figure 1: Distribution of Lateness: (a) PRD Sequencing and (b) SPT Sequencing under Operation Due Date Dispatching; and (c) PRD Sequencing and (d) SPT Sequencing under Shortest Processing Time Dispatching

Table 1: Summary of Simulated Shop and Job Characteristics

Shop Characteristics	Routing Variability	Random routing; no re-entrant flows
	No. of Work Centers	6
	Interchange-ability of Work Centers	No interchange-ability
	Work Center Capacities	All equal
	Work Center Utilization Rate	90%
Job Characteristics	No. of Operations per Job	Discrete Uniform[1, 6]
	Operation Processing Times	Truncated 2–Erlang; (mean = 1; max = 4)
	Due Date Determination Procedure	Due Date = Entry Time + d; d U ~ [30, 50]
	Inter-Arrival Times	Exp. Distribution; mean = 0.648

Table 2: Summary of the Release Methods and Sequencing Rules Applied in this Study

	Abbr.	Full Name	Brief Description	Parameter
Classical Release Methods	AGGWLT	Aggregate Workload Trigger	If the total shop load (of all uncompleted operations) falls below the norm level, all jobs in the pool with the work center which has the lowest direct load as the first in their routing are considered for release.	Trigger 60-210 (6 levels)
	WCWLT	Work Center Workload Trigger	If the direct load of any work center falls below the workload norm level, all jobs in the pool with this work center as the first in their routing are considered for release.	Trigger 0-5 (6 levels)
New Release Methods	CorrWLT	Corrected Workload Trigger	As for the classical work center workload trigger (WCWLT), but release is triggered when the corrected aggregate load of any work center falls below the norm level.	Trigger 2-7 (6 levels)
	UBR	Upper Bound Release	Release takes place whenever a new job arrives at the pool or an operation is completed. All jobs in the pool are considered for release once. The workload of each operation in the routing of a job contributes to the load of the corresponding work centers in accordance with the corrected aggregate load approach. The new workload of each work center is compared against the predetermined workload limits or norms. If one or more norms are violated, the job is retained in the pre-shop pool. If norms are not violated, the job is selected for release onto the shop floor and its workload is formally assigned to the workloads of the work centers in its routing.	Norm 5-10 (6 levels)
	UBRLB	Upper Bound Release with Load Balancing	As for UBR, but the release is allowed even if the norm is violated providing that the overload caused is smaller than the under-load that is reduced or relieved	Norm 4-9 (6 levels)
Sequencing Rules for Jobs in the Pool	FCFS	First-Come-First-Served	The job which arrived in the pool first is considered for release first.	None
	EDD	Earliest Due Date	The job with the earliest due date is considered for release first.	None
	PRD	Planned Release Date	The job with the earliest planned release date is considered for release first.	a=4
	LWIB	Lowest Workload ImBalance	The job with the lowest imbalance index (see Equation (2)) is considered for release first.	None
	SPT	Shortest Processing Time	The job with the shortest processing time at the first work center in its routing is considered for release first	None

Table 3: ANOVA Results

	Source of Variance	Sum of Squares	df[1]	Mean Squares	F-Ratio	p-Value
$t_t{}^{2}$	Release Methods (RM)	51821.923	4	12955.481	2198.389	0.000
	Sequencing Rule (SR)	60692.601	4	15173.150	2574.701	0.000
	Dispatching (Disp)	267188.155	1	267188.155	45338.617	0.000
	SR x Disp	9688.851	4	2422.213	411.020	0.000
	Error	176712.580	29986	5.893		
$t_l{}^{3}$	Release Methods (RM)	255891.905	4	63972.976	5881.853	0.000
	Sequencing Rule (SR)	80504.708	4	20126.177	1850.457	0.000
	Dispatching (Disp)	258973.961	1	258973.961	23810.786	0.000
	SR x Disp	12742.675	4	3185.669	292.899	0.000
	Error	326137.620	29986	10.876		
Pt^{4}	Release Methods (RM)	73.432	4	18.358	7934.892	0.000
	Sequencing Rule (SR)	19.302	4	4.826	2085.757	0.000
	Dispatching (Disp)	0.216	1	0.216	93.494	0.000
	SR x Disp	6.709	4	1.677	724.972	0.000
	Error	69.375	29986	0.002		
Mt^{5}	Release Methods (RM)	34227.435	4	8556.859	2922.791	0.000
	Sequencing Rule (SR)	9000.328	4	2250.082	768.567	0.000
	Dispatching (Disp)	8358.306	1	8358.306	2854.971	0.000
	SR x Disp	19.078	4	4.770	1.629	0.164
	Error	87787.993	29986	2.928		

[1]) degrees of freedom; [2]) Throughput time; [3]) Lead time; [4]) Percentage tardy; [5]) Mean tardiness

Table 4: Results for Scheffé Multiple Comparison Procedure

Sequencing Rule (x)	Sequencing Rule (y)	Throughput Time		Lead Time		Percentage Tardy		Mean Tardiness	
		lower[1]	upper	lower	upper	lower	upper	lower	upper
EDD	FCFS	-0.135	0.138*	-0.205	0.166*	-0.005	0.001*	-0.148	0.044*
EDD	LWIB	-3.699	-3.426	-2.702	-2.332	-0.040	-0.034	-1.213	-1.021
EDD	PRD	-0.134	0.139*	-0.041	0.330*	0.000	0.006	-0.014	0.178*
EDD	SPT	-0.166	0.107*	2.475	2.845	0.040	0.046	0.453	0.645
FCFS	LWIB	-3.700	-3.427	-2.683	-2.312	-0.038	-0.032	-1.161	-0.969
FCFS	PRD	-0.136	0.137*	-0.022	0.349*	0.002	0.007	0.038	0.230
FCFS	SPT	-0.168	0.105*	2.494	2.865	0.042	0.048	0.505	0.697
LWIB	PRD	3.428	3.701	2.476	2.847	0.037	0.042	1.102	1.295
LWIB	SPT	3.396	3.669	4.991	5.362	0.077	0.083	1.570	1.762
PRD	SPT	-0.168	0.105*	2.330	2.701	0.038	0.043	0.371	0.563

[1]) 95% confidence interval; * not significant at α=0.05

Table 5: Comparison of Performance Results under Operation Due Date Dispatching

		FCFS				EDD				PRD				LWIB				SPT			
		t_t [1]	t_l [2]	P_{tard} [3]	M_{tard} [4]	t_t	t_l	P_{tard}	M_{tard}	t_t	t_l	P_{tard}	M_{tard}	t_t	t_l	P_{tard}	M_{tard}	t_t	t_l	P_{tard}	M_{tard}
IMM		23.5	23.5	10.5%	0.8	23.5	23.5	10.5%	0.8	23.5	23.5	10.5%	0.8	23.5	23.5	10.5%	0.8	23.5	23.5	10.5%	0.8
AGGWLT	WLT60	11.2	26.7	16.7%	4.3	11.3	26.7	16.4%	4.2	11.2	26.6	15.9%	4.2	24.9	43.7	46.2%	12.5	11.6	21.6	7.7%	2.9
	WLT90	12.2	26.0	15.4%	3.6	12.3	25.9	15.1%	3.6	12.3	25.7	14.4%	3.4	24.2	37.9	37.7%	7.8	12.6	21.5	7.6%	2.4
	WLT120	12.8	25.9	15.5%	3.5	12.8	25.9	15.2%	3.4	12.8	25.6	14.6%	3.3	24.6	35.1	34.6%	5.4	13.0	21.6	8.0%	2.3
	WLT150	13.1	25.9	15.7%	3.4	13.1	25.9	15.5%	3.3	13.0	25.6	14.8%	3.2	25.1	34.1	33.9%	4.6	13.2	21.6	8.3%	2.2
	WLT180	13.3	25.9	16.0%	3.4	13.3	25.9	15.8%	3.3	13.2	25.5	14.9%	3.1	25.6	34.0	34.0%	4.4	13.4	21.7	8.5%	2.2
	WLT210	13.4	25.9	16.2%	3.3	13.4	25.8	15.9%	3.2	13.3	25.5	15.1%	3.1	25.7	33.8	33.7%	4.3	13.4	21.7	8.6%	2.2
WCWLT	WLT 0	8.1	25.9	18.1%	4.4	8.1	25.8	17.6%	4.3	8.1	25.4	16.9%	4.1	20.8	41.6	38.9%	12.2	8.0	19.0	8.0%	3.0
	WLT 1	9.5	25.9	17.1%	4.1	9.5	25.9	16.9%	4.0	9.4	25.4	16.0%	3.8	19.5	33.4	26.5%	6.2	9.5	19.6	7.5%	2.8
	WLT 2	11.7	25.8	15.1%	3.5	11.7	25.8	14.7%	3.4	11.6	25.4	14.0%	3.3	19.6	29.8	19.6%	3.6	11.8	20.6	6.5%	2.4
	WLT 3	13.4	25.4	13.0%	2.9	13.4	25.4	12.7%	2.9	13.3	25.1	12.2%	2.8	20.3	28.6	16.9%	2.6	13.5	21.1	5.5%	2.0
	WLT 4	14.8	25.0	11.4%	2.5	14.8	25.0	11.2%	2.4	14.7	24.7	10.7%	2.3	21.2	28.2	15.8%	2.1	14.9	21.5	4.7%	1.7
	WLT 5	16.0	24.7	10.3%	2.1	16.0	24.7	10.1%	2.0	15.9	24.4	9.7%	1.9	21.8	28.0	15.6%	1.8	16.1	21.8	4.2%	1.5
CorrWLT	WLT 2	11.8	29.7	19.7%	5.7	11.8	29.5	19.3%	5.5	11.9	29.0	18.1%	5.3	19.6	33.9	27.1%	6.5	11.9	21.7	7.4%	3.1
	WLT 3	14.4	25.2	11.4%	2.6	14.4	25.1	10.9%	2.4	14.5	24.9	10.3%	2.4	19.8	30.4	20.6%	3.9	14.6	21.1	4.6%	1.9
	WLT 4	16.4	24.2	8.9%	1.7	16.4	24.2	8.6%	1.6	16.5	24.0	8.1%	1.6	20.5	28.8	17.0%	2.6	16.6	21.5	3.4%	1.3
	WLT 5	17.8	23.8	8.1%	1.2	17.8	23.8	7.9%	1.2	17.9	23.7	7.7%	1.2	21.3	28.2	15.6%	2.0	18.0	21.9	2.9%	1.0
	WLT 6	18.9	23.7	8.1%	1.0	18.9	23.6	8.0%	1.0	18.9	23.6	8.0%	1.0	22.1	27.9	15.3%	1.7	19.1	22.3	3.2%	0.8
	WLT 7	19.7	23.6	8.5%	0.9	19.7	23.6	8.4%	0.9	19.7	23.5	8.6%	0.9	22.6	27.8	15.3%	1.5	19.9	22.5	4.0%	0.7
UBR	N5	15.3	20.0	3.2%	1.5	15.3	20.0	3.1%	1.5	15.3	20.0	3.1%	1.5	15.1	19.8	3.4%	1.7	15.1	19.6	3.2%	1.8
	N6	17.3	20.8	2.5%	1.0	17.3	20.8	2.4%	1.0	17.3	20.8	2.4%	1.0	17.1	20.6	2.5%	1.2	17.2	20.5	2.4%	1.2
	N7	18.8	21.5	2.4%	0.8	18.8	21.5	2.4%	0.8	18.8	21.5	2.4%	0.8	18.7	21.4	2.2%	0.9	18.7	21.2	2.0%	0.9
	N8	19.9	22.0	3.0%	0.6	19.9	22.0	2.9%	0.6	19.9	22.0	3.0%	0.6	19.7	21.9	2.5%	0.7	19.8	21.8	2.3%	0.7
	N9	20.6	22.3	4.1%	0.5	20.6	22.3	4.0%	0.5	20.7	22.3	4.1%	0.5	20.6	22.2	3.4%	0.6	20.6	22.2	3.1%	0.6
	N10	21.3	22.6	5.4%	0.5	21.3	22.6	5.4%	0.5	21.3	22.6	5.5%	0.5	21.2	22.5	4.6%	0.5	21.2	22.5	4.5%	0.5
UBRLB	N4	14.4	19.9	3.5%	2.2	14.4	19.9	3.4%	2.2	14.4	20.0	3.5%	2.2	14.3	19.7	3.5%	2.3	14.2	19.7	3.6%	2.5
	N5	16.7	20.5	2.5%	1.3	16.7	20.5	2.5%	1.3	16.8	20.6	2.5%	1.3	16.6	20.3	2.5%	1.4	16.6	20.3	2.5%	1.5
	N6	18.4	21.3	2.2%	1.0	18.4	21.3	2.2%	0.9	18.4	21.3	2.3%	1.0	18.3	21.1	2.1%	1.0	18.3	21.1	2.1%	1.1
	N7	19.6	21.8	2.6%	0.7	19.6	21.8	2.6%	0.7	19.6	21.9	2.6%	0.7	19.6	21.7	2.2%	0.8	19.6	21.7	2.2%	0.8
	N8	20.5	22.3	3.6%	0.6	20.5	22.3	3.6%	0.6	20.5	22.3	3.7%	0.6	20.4	22.2	3.1%	0.7	20.4	22.2	2.9%	0.7
	N9	21.1	22.5	5.0%	0.6	21.1	22.5	5.0%	0.5	21.1	22.6	5.2%	0.6	21.1	22.5	4.4%	0.6	21.1	22.5	4.3%	0.6

[1] t_t - Throughput time; [2] t_l - Lead time; [3] P_{tard} – Percentage tardy; [4] M_{tard} – Mean tardiness

Table 6: Comparison of Performance Results under Shortest Processing Time Dispatching

		FCFS				EDD				PRD				LWIB				SPT			
		t_t[1]	t_l[2]	P_{tard}[3]	M_{tard}[4]	t_t	t_l	P_{tard}	M_{tard}	t_t	t_l	P_{tard}	M_{tard}	t_t	t_l	P_{tard}	M_{tard}	t_t	t_l	P_{tard}	M_{tard}
IMM		13.5	13.5	5.4%	2.4	13.5	13.5	5.4%	2.4	13.5	13.5	5.4%	2.4	13.5	13.5	5.4%	2.4	13.5	13.5	5.4%	2.4
AGGWLT	WLT60	8.7	23.9	17.6%	4.7	8.8	23.8	17.2%	4.7	8.8	23.6	16.7%	4.5	13.9	31.6	24.4%	11.1	8.8	18.4	9.2%	3.3
	WLT90	9.2	23.0	16.5%	4.4	9.2	23.0	16.2%	4.4	9.2	22.7	15.6%	4.2	13.9	25.9	18.9%	7.2	9.3	18.0	8.9%	3.0
	WLT120	9.4	22.7	16.1%	4.3	9.4	22.6	15.8%	4.3	9.5	22.4	15.4%	4.1	14.3	23.8	16.6%	6.0	9.5	17.8	8.8%	2.9
	WLT150	9.6	22.5	16.0%	4.3	9.6	22.5	15.7%	4.2	9.6	22.2	15.2%	4.1	14.7	23.1	15.8%	5.7	9.6	17.8	8.7%	2.9
	WLT180	9.6	22.3	15.9%	4.2	9.6	22.4	15.7%	4.2	9.6	22.0	15.1%	4.0	14.9	23.0	15.6%	5.6	9.6	17.7	8.6%	2.9
	WLT210	9.7	22.3	15.9%	4.3	9.7	22.3	15.6%	4.2	9.7	22.0	15.2%	4.0	15.1	23.0	15.5%	5.7	9.6	17.7	8.6%	3.0
WCWLT	WLT 0	6.6	24.8	19.2%	4.8	6.6	24.8	18.7%	4.7	6.6	24.4	18.0%	4.5	12.5	33.6	24.1%	13.2	6.6	17.7	9.0%	3.2
	WLT 1	7.5	24.5	18.7%	4.7	7.5	24.5	18.3%	4.6	7.5	24.2	17.7%	4.4	11.9	26.0	17.8%	7.3	7.5	17.9	8.7%	3.1
	WLT 2	8.6	23.5	17.5%	4.4	8.6	23.6	17.2%	4.3	8.6	23.2	16.4%	4.0	12.0	22.5	14.1%	4.9	8.6	17.8	8.4%	2.9
	WLT 3	9.6	22.4	16.1%	4.0	9.6	22.4	15.8%	3.9	9.5	22.1	15.3%	3.7	12.4	20.9	12.2%	4.0	9.6	17.5	8.1%	2.7
	WLT 4	10.2	21.2	14.9%	3.7	10.2	21.2	14.6%	3.6	10.1	20.9	14.1%	3.4	12.7	20.0	11.1%	3.5	10.2	17.0	8.0%	2.5
	WLT 5	10.5	20.0	13.4%	3.5	10.5	19.9	13.2%	3.4	10.5	19.8	12.9%	3.2	13.0	19.3	10.3%	3.3	10.5	16.4	7.6%	2.5
CorrWLT	WLT 2	8.6	26.1	20.8%	5.7	8.6	26.1	20.3%	5.6	8.6	25.6	19.5%	5.3	12.0	25.7	17.8%	7.2	8.6	18.6	9.1%	3.7
	WLT 3	9.8	21.3	14.9%	3.7	9.8	21.3	14.5%	3.6	9.8	21.1	14.1%	3.4	12.1	22.0	13.9%	4.7	9.9	16.6	7.4%	2.6
	WLT 4	10.7	19.1	12.1%	3.1	10.7	19.1	11.9%	3.0	10.7	19.0	11.6%	2.9	12.5	20.2	11.7%	3.7	10.8	15.9	6.9%	2.3
	WLT 5	11.3	17.8	10.4%	2.8	11.3	17.8	10.2%	2.7	11.3	17.7	10.1%	2.7	12.9	19.3	10.3%	3.2	11.4	15.4	6.6%	2.3
	WLT 6	11.7	16.8	9.2%	2.6	11.7	16.8	9.1%	2.6	11.7	16.8	9.1%	2.5	13.2	18.7	9.5%	3.0	11.8	15.0	6.3%	2.2
	WLT 7	12.0	16.2	8.4%	2.6	12.0	16.2	8.3%	2.5	12.0	16.2	8.3%	2.5	13.5	18.3	8.8%	2.9	12.1	14.8	6.1%	2.2
UBR	N5	10.1	14.8	6.2%	2.2	10.1	14.8	6.2%	2.2	10.1	14.7	6.1%	2.1	10.1	14.7	6.2%	2.4	10.0	14.6	5.9%	2.4
	N6	10.9	14.3	6.0%	2.1	10.9	14.3	6.0%	2.0	10.9	14.3	6.0%	2.0	10.9	14.2	6.0%	2.2	10.9	14.1	5.8%	2.2
	N7	11.5	14.1	5.9%	2.1	11.5	14.1	5.9%	2.1	11.5	14.1	5.9%	2.1	11.5	14.0	5.9%	2.2	11.5	13.9	5.7%	2.2
	N8	11.9	13.9	5.8%	2.1	11.9	13.9	5.7%	2.1	11.9	13.9	5.7%	2.1	11.9	13.9	5.8%	2.2	11.9	13.8	5.7%	2.2
	N9	12.3	13.8	5.7%	2.2	12.3	13.8	5.6%	2.2	12.3	13.8	5.7%	2.2	12.2	13.8	5.7%	2.3	12.2	13.7	5.6%	2.3
	N10	12.5	13.7	5.6%	2.2	12.5	13.7	5.6%	2.2	12.5	13.7	5.6%	2.2	12.5	13.7	5.6%	2.3	12.5	13.7	5.5%	2.3
UBRLB	N4	9.6	15.5	6.0%	2.8	9.6	15.5	6.0%	2.7	9.6	15.6	6.0%	2.8	9.6	15.4	5.9%	2.9	9.5	15.5	5.8%	3.2
	N5	10.6	14.5	5.8%	2.1	10.6	14.5	5.8%	2.1	10.6	14.5	5.8%	2.1	10.6	14.3	5.7%	2.2	10.6	14.4	5.7%	2.3
	N6	11.3	14.2	5.8%	2.1	11.3	14.2	5.8%	2.1	11.3	14.3	5.8%	2.1	11.3	14.1	5.7%	2.1	11.3	14.1	5.7%	2.2
	N7	11.8	14.1	5.8%	2.1	11.8	14.1	5.7%	2.1	11.8	14.1	5.8%	2.1	11.8	14.0	5.7%	2.2	11.8	14.0	5.6%	2.2
	N8	12.2	13.9	5.7%	2.2	12.2	13.9	5.7%	2.2	12.2	14.0	5.7%	2.2	12.2	13.9	5.6%	2.3	12.2	13.9	5.6%	2.3
	N9	12.5	13.9	5.6%	2.3	12.5	13.9	5.6%	2.3	12.5	13.9	5.6%	2.3	12.5	13.8	5.5%	2.3	12.5	13.8	5.5%	2.3

[1] t_t - Throughput time; [2] t_l - Lead time; [3] P_{tard} – Percentage tardy; [4] M_{tard} – Mean tardiness

Table 7: Job Classes: Frequency Distribution for Routing Lengths 1 to 6

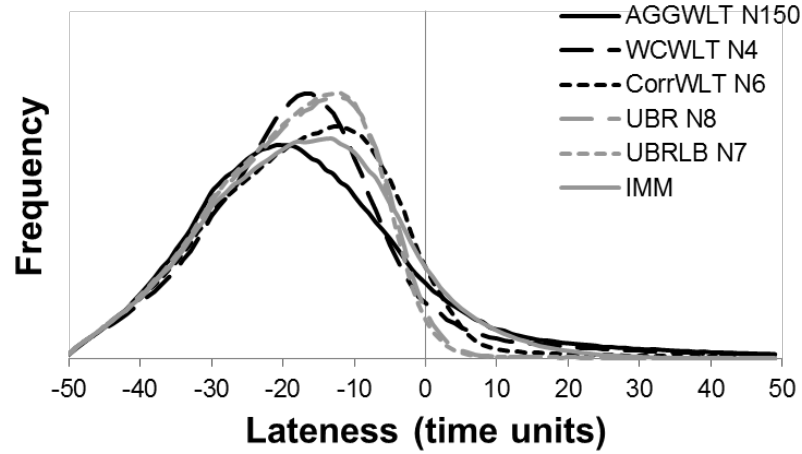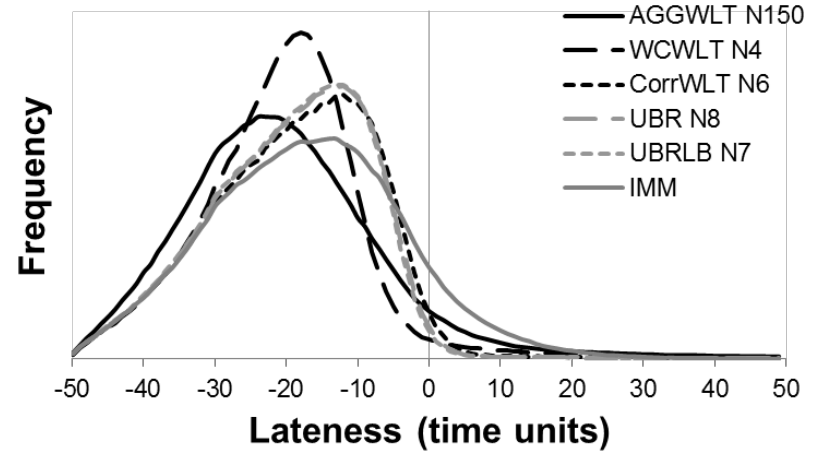|	RL[1]1	RL2	RL3	RL4	RL5	RL6
Small	5.6%	3.6%	2.5%	1.8%	1.3%	1.0%
Medium	6.9%	9.7%	11.3%	12.8%	14.0%	14.4%
Large	3.9%	3.2%	2.6%	2.2%	1.8%	1.4%

[1] RL – Routing Length

Table 8: Performance Analysis of Job Size Classes for PRD Sequencing (under Operation Due Date Dispatching): Time to Release (in Time Units)

		Small Jobs						Medium Jobs						Large Jobs					
		RL[1]1	RL2	RL3	RL4	RL5	RL6	RL1	RL2	RL3	RL4	RL5	RL6	RL1	RL2	RL3	RL4	RL5	RL6
AGGWLT	WLT60	19.4	17.9	16.5	15.2	13.8	12.5	19.2	17.5	16.1	14.5	13.1	11.9	18.6	17.0	15.3	13.8	12.4	11.2
	WLT90	17.0	15.5	14.3	13.3	11.9	11.1	16.7	15.2	13.9	12.6	11.5	10.6	16.1	14.8	13.2	12.1	11.0	10.0
	WLT120	16.3	14.9	13.8	12.7	11.5	10.6	16.0	14.7	13.3	12.1	11.0	10.2	15.5	14.2	12.7	11.6	10.5	9.7
	WLT150	16.0	14.6	13.5	12.5	11.2	10.3	15.7	14.3	13.0	11.8	10.6	9.9	15.2	13.9	12.4	11.3	10.2	9.6
	WLT180	15.8	14.4	13.2	12.1	11.1	10.0	15.5	14.1	12.8	11.6	10.4	9.7	15.0	13.8	12.3	11.1	9.9	9.4
	WLT210	15.7	14.4	13.1	12.0	11.0	9.9	15.4	14.1	12.7	11.5	10.3	9.6	15.0	13.6	12.2	11.0	9.8	9.3
WCWLT	WLT 0	22.6	20.8	18.9	17.1	15.4	13.7	22.4	20.2	18.3	16.4	14.4	12.8	21.8	19.6	17.4	15.4	13.4	11.7
	WLT 1	20.9	19.2	17.4	15.8	14.1	12.7	20.7	18.7	16.9	15.1	13.3	11.8	20.1	18.1	16.1	14.2	12.3	10.8
	WLT 2	18.0	16.6	15.1	13.7	12.3	10.9	17.9	16.1	14.5	13.0	11.4	10.1	17.3	15.6	13.8	12.2	10.6	9.2
	WLT 3	15.4	14.1	12.8	11.6	10.3	9.2	15.3	13.8	12.4	11.1	9.8	8.6	14.8	13.3	11.8	10.4	9.1	7.9
	WLT 4	13.2	12.0	11.0	10.0	8.8	7.8	13.0	11.7	10.6	9.4	8.3	7.3	12.6	11.3	10.1	8.9	7.7	6.6
	WLT 5	11.2	10.2	9.4	8.4	7.6	6.7	11.1	10.0	9.0	8.0	7.1	6.2	10.7	9.6	8.6	7.6	6.6	5.7
CorrWLT	WLT 2	22.3	20.4	18.7	17.0	15.2	13.6	22.0	19.9	18.0	16.2	14.3	12.8	21.4	19.3	17.2	15.2	13.3	11.7
	WLT 3	13.9	12.5	11.3	10.1	9.0	7.9	13.7	12.2	10.9	9.6	8.3	7.3	13.3	11.7	10.3	8.9	7.7	6.6
	WLT 4	10.4	9.3	8.4	7.5	6.5	5.7	10.2	9.0	8.0	7.0	6.0	5.3	9.8	8.6	7.5	6.5	5.6	4.7
	WLT 5	8.0	7.2	6.5	5.8	5.0	4.4	7.9	7.0	6.2	5.4	4.7	4.1	7.6	6.7	5.8	5.0	4.3	3.7
	WLT 6	6.4	5.7	5.2	4.6	4.0	3.6	6.3	5.6	4.9	4.3	3.7	3.2	6.1	5.3	4.6	4.0	3.4	2.9
	WLT 7	5.2	4.6	4.2	3.8	3.3	2.9	5.1	4.5	4.0	3.5	3.1	2.7	5.0	4.3	3.8	3.3	2.8	2.4
UBR	N5	0.2	0.4	0.6	0.7	0.7	0.7	1.4	2.6	3.5	4.0	4.3	4.6	11.2	13.2	14.0	14.9	14.7	15.0
	N6	0.2	0.4	0.5	0.6	0.6	0.6	1.2	2.1	2.7	3.0	3.2	3.3	8.4	9.7	10.1	10.6	10.4	10.5
	N7	0.1	0.3	0.4	0.4	0.5	0.4	1.0	1.7	2.1	2.4	2.5	2.5	6.6	7.4	7.7	8.0	7.7	7.7
	N8	0.1	0.2	0.3	0.4	0.4	0.3	0.8	1.4	1.7	1.9	1.9	1.9	5.2	6.0	6.0	6.2	6.0	5.8
	N9	0.1	0.2	0.3	0.3	0.3	0.3	0.6	0.9	1.1	1.2	1.2	1.2	4.3	4.7	4.7	4.9	4.6	4.4
	N10	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.3	3.7	3.7	3.8	3.6	3.5
UBRLB	N4	1.1	1.5	1.5	1.5	1.5	1.4	4.5	4.5	4.2	3.8	3.5	3.3	33.1	18.3	12.6	9.7	7.6	7.0
	N5	1.0	1.2	1.2	1.3	1.1	1.1	3.6	3.4	3.1	2.7	2.5	2.2	20.1	11.7	8.1	6.4	5.0	4.5
	N6	0.9	1.1	1.0	1.0	0.9	0.8	3.0	2.7	2.4	2.1	1.8	1.6	14.9	8.7	5.9	4.6	3.6	3.1
	N7	0.8	0.9	0.9	0.8	0.8	0.7	2.5	2.2	1.9	1.6	1.4	1.2	11.5	6.6	4.5	3.5	2.7	2.3
	N8	0.6	0.8	0.7	0.7	0.6	0.5	2.1	1.8	1.5	1.3	1.1	1.0	9.1	5.2	3.4	2.7	2.1	1.6
	N9	0.6	0.7	0.7	0.6	0.5	0.4	1.7	1.5	1.2	1.0	0.9	0.7	7.1	4.1	2.7	2.1	1.6	1.3

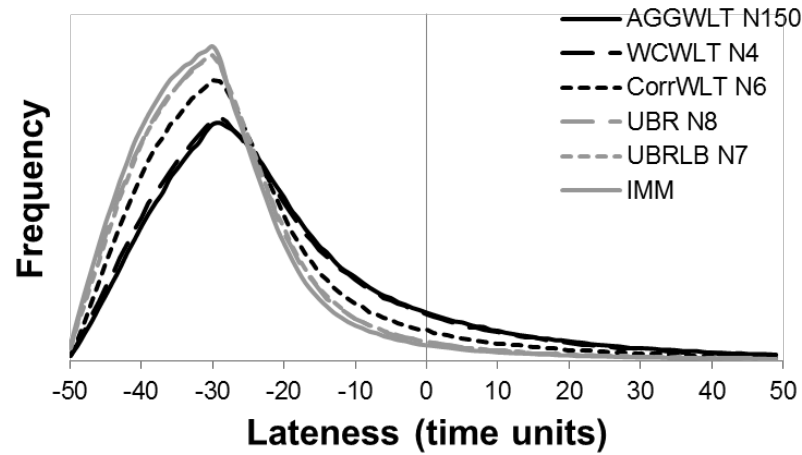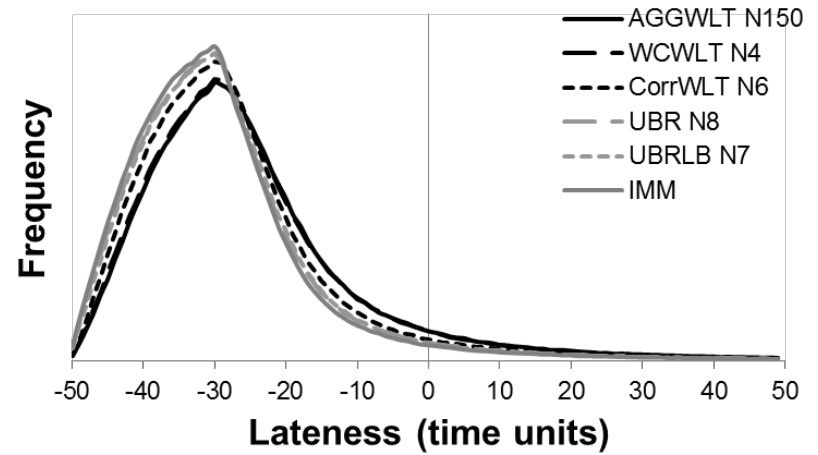[1] RL – Routing Length

Figure 1: Distribution of Lateness: (a) PRD Sequencing and (b) SPT Sequencing under Operation Due Date Dispatching; and (c) PRD Sequencing and (d) SPT Sequencing under Shortest Processing Time Dispatching