On the Backlog-Sequencing Decision for Extending the Applicability of ConWIP to High-Variety Contexts: An Assessment by Simulation

Matthias Thürer (corresponding author), Nuno O. Fernandes, Mark Stevenson and Ting Qu

Name: Prof. Matthias Thürer
Institution: Jinan University
Address: Institute of Physical Internet
School of Electrical and Information Engineering
Jinan University (Zhuhai Campus)
519070, Zhuhai, PR China
E-mail: matthiasthurer@workloadcontrol.com

Name: Prof. Nuno O. Fernandes
Institution: Instituto Politécnico de Castelo Branco
Address: Av. do Empresário, 6000-767
Castelo Branco - Portugal
E-mail: nogf@ipcb.pt

Name: Prof. 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

Name: Prof. Ting Qu
Institution: Jinan University
Address: Institute of Physical Internet
School of Electrical and Information Engineering
Jinan University (Zhuhai Campus)
519070, Zhuhai, PR China
E-mail: quting@jnu.edu.cn

Keywords: *Constant Work-in-Process (ConWIP); make-to-order (MTO) production; dispatching; Workload Control; backlog-sequencing rule.*

On the Backlog-Sequencing Decision for Extending the Applicability of ConWIP to High-Variety Contexts: An Assessment by Simulation

Abstract

Constant Work-in-Process (ConWIP) is a card-based control system that was developed for simple flow shops – a lack of load balancing capabilities hinders its application to more complex shops. In contrast, load balancing is an integral part of Workload Control, a production planning and control concept developed for high-variety environments. One means of load balancing evident in the Workload Control literature is through the use of a capacity slack-based backlog-sequencing rule. This study therefore investigates the potential of the backlog-sequencing decision to improve load balancing in the context of ConWIP, thereby making it suitable for more complex, high-variety environments. Using simulation, we demonstrate that: (i) the choice of backlog-sequencing rule significantly impacts throughput times and tardiness related performance measures; and, (ii) capacity slack-based sequencing rules achieve significant performance improvements over 'classical' ConWIP backlog-sequencing rules. These results significantly extend the applicability of ConWIP. Results from the Workload Control literature however do not directly translate across to ConWIP. The simplified release procedure of ConWIP makes backlog-sequencing based on planned release dates dysfunctional. This negatively impacts the performance of modified capacity slack-based sequencing rules that were recently shown to be the best choice for Workload Control.

Keywords: *Constant Work-in-Process (ConWIP); make-to-order (MTO) production; dispatching; Workload Control; backlog-sequencing rule.*

1. Introduction

Constant Work-in-Process (ConWIP; e.g. Spearman *et al*., 1990; Hopp & Spearman, 2001) is a simple card-based production control system. It is essentially a pull system (Hopp & Spearman, 2004) that uses a so-called Work-In-Process (WIP) limit or cap (WIP-Cap) that is pre-established by management to realize input/output control (Wight, 1970; Plossl & Wight, 1971). In accordance with input/output control, the output of work from the shop floor determines the input of work to the shop floor from a so-called pre-shop pool or 'backlog' (in Spearman *et al*., 1990). Jobs are only permitted to enter the shop floor if the WIP-Cap is not violated; otherwise, they form a 'backlog' and have to wait in the pre-shop pool until some of the jobs on the shop floor have been completed. Cards circulate between the shop floor and the pool; and the return of a card signals that a job has been completed.

ConWIP is a simple means of exercising pull control, providing that product variety is restricted – its applicability to high-variety make-to-order environments is rather limited (Thürer *et al*., 2016a). A key reason for this is its lack of load balancing capabilities (Germs & Riezebos, 2010). Load balancing is here defined as a leveling of the workload across resources. ConWIP's WIP-Cap restricts the work-in-process released to the shop floor but it does not balance the workload on the shop floor across resources if, for example, processing times, routings, and/or the occurrence of demand follow a stochastic process. In this context, tools for load balancing, such as line balancing and task analysis, which presuppose a certain degree of repetitiveness, do not apply. An alternative approach for improving load balancing has been presented in the Workload Control literature in the form of the "backlog-sequencing decision" (Philipoom *et al*., 1993; Fredendall *et al.*, 2010; Thürer *et al*., 2015, 2016b). Workload Control – and its card-based variant, Control of Balance by Card Based Navigation (COBACABANA: Land, 2009; Thürer *et al*., 2014) – is an alternative production planning and control system to ConWIP that was developed for high-variety contexts (Stevenson *et al*., 2005). In contrast to ConWIP, Workload Control incorporates load balancing as part of its workload limiting strategy. Yet Thürer *et al*. (2015) recently demonstrated that load balancing can and should be enhanced using an appropriate backlog-sequencing rule to influence the sequence in which jobs are considered for release. Specifically, the capacity slack-based backlog-sequencing rule proposed by Philipoom *et al*. (1993) was shown by Thürer *et al*. (2015) to have much promise. It is therefore argued here that load balancing should be embedded within ConWIP in the form of an appropriate backlog-sequencing rule; and that doing so will extend the scope and applicability of this important card-based system.

While the importance of the so-called ‘backlog-sequencing problem’ has been recognized in some of the ConWIP literature, previous studies have often focused on complex optimization algorithms (e.g. Woodruff & Spearman, 1992; Herer & Masin, 1997; Golany *et al*., 1999; Framinan *et al*., 2001; Zhang & Chen, 2001; Cao & Chen, 2005). In this body of work, a fixed set of orders has been assumed and the sequence in which these orders should be released by a ConWIP system to optimize a certain performance parameter has been determined. However, in a make-to-order system, where job arrivals follow a stochastic process, jobs may arrive at any moment in time. As a consequence, not only does the optimization algorithm need to be executed at each release instance, but a so-called optimal solution may turn out to be far from optimal when a new job arrives that needs to be incorporated into the existing schedule. Therefore, we agree with Lingayat *et al*. (1995) that a greedy heuristic, i.e. a simple backlog-sequencing rule, represents a more feasible solution than optimization for this context. The main prior study on simple sequencing rules was presented by Leu (2000), but this contribution does not reflect recent advances, such as the emergence of capacity slack-based sequencing rules. In response, we ask:

Can a backlog-sequencing rule be used to extend the applicability of ConWIP to high-variety make-to-order flow shops?

An exploratory study based on controlled simulation experiments is used to provide an answer to this question. We will show that specifically capacity-slack based backlog sequencing rules have the potential to improve performance compared to ‘classical’ ConWIP backlog-sequencing rules.

The remainder of this paper is organized as follows. The literature is first reviewed to identify the backlog-sequencing rules available in the ConWIP and Workload Control literatures in Section 2. Here we also use the capacity slack-based rules from the Workload Control literature as the basis for the design of new capacity slack-based rules that reflect the particular characteristics of ConWIP. Section 3 then outlines the simulation model that is used to examine the performance impact of improved backlog-sequencing in a high variety context (in terms of job arrival times, processing times, and routings). The results are then presented and discussed in Section 4. Finally, concluding remarks are made in Section 5, where managerial implications and future research directions are also outlined.

2. Literature Review – Backlog-Sequencing Rules

ConWIP, as illustrated in Figure 1, is arguably the simplest card-based control system available in the literature. Whenever the number of jobs in the system (or shop floor) is below a pre-established limit, a new job is released to the system. To control the number of jobs, each job in the system has to have a ConWIP card attached to it. Thus, by restricting the number of cards that can circulate in the system, the number of jobs is also restricted. Once a job leaves the system, its card is freed and can be used by a different job from the set of jobs waiting to enter the system. The place where these jobs (the 'backlog') wait is referred to as a "pool". The decision concerning which job(s) to release next is called the "backlog-sequencing decision".

[Take in Figure 1]

ConWIP is a simple means of exercising pull control, providing that product variety is restricted. Indeed, Hopp & Spearman (2001, p. 461) argued that ConWIP only works well if routings are constant and processing time variability is low. The main mean of improving the performance of ConWIP is by changing the sequence in which jobs are released to the shop floor. Other means of bringing about an improvement, such as by changing the loop structure, do not apply since they would transform ConWIP into a different system altogether; see, e.g. Thürer *et al.* (2016a) for a discussion on alternative card-based control systems. Further, this study focuses on ConWIP and on a balanced shop since in a shop with stationary bottleneck(s) load balancing across resources is less important. ConWIP extensions, such as ConWork or ConLoad (Rose, 1999), which presuppose a stationary bottleneck, are therefore not considered.

This section does not aim to present a complete review of the ConWIP (or Workload Control) literature; rather, it focuses on identifying the backlog-sequencing rules to be considered in our study. For a broader review of ConWIP, the reader is referred to Framinan *et al*. (2003) and Prakash & Chin (2015). Our review hereby focuses on the limited number of greedy heuristics available in the ConWIP and Workload Control literature. This is motivated by the fact that the backlog-sequencing rule must be suitable for high-variety make-to-order contexts where processing times, routings, and the inter-arrival times of orders follow a stochastic process. This setting means that a significant part of the literature – that assumes a given set of jobs, which are optimized for a certain set of performance measures – are omitted as not being relevant because job arrivals follow a stochastic process. Approaches based on linear/non-linear integer programming (e.g. Herer & Masin, 1997; Luh *et al*., 2000; Zhang &

Chen, 2001; Cao & Chen, 2005) or meta-heuristics (e.g. Woodruff & Spearman, 1992; Golany *et al*., 1999; Liu, 2010) are arguably not feasible in the stochastic production environment considered in our study.

Section 2.1 first reviews the most commonly applied backlog-sequencing rules that have been used with ConWIP. Section 2.2 then discusses the capacity slack-based backlog-sequencing rules from the Workload Control literature. This includes a discussion on potential refinements to adapt these rules for use with ConWIP.

2.1 Backlog-sequencing Rules from the ConWIP Literature

Many papers that apply ConWIP do not specify which backlog-sequencing rule is incorporated (e.g. Spearman *et al*., 1990; Germs & Riezebos, 2010). It appears that this aspect of the system was either not specified (or overlooked) or it was assumed that this did not have a significant impact on performance. The most widely used backlog-sequencing rules in the ConWIP literature are arguably as follows:

- *First-Come-First-Served (FCFS)*, a time-oriented rule that sequences jobs according to their time of arrival in the pool. This rule was used, e.g. by Leu (2000) and Ryan & Vorasayan (2005).
- *Earliest Due Date (EDD)*, a time-oriented rule that sequences jobs according to their due date. This rule was used, e.g. by Leu (2000).
- *Planned Release Date (PRD)*, a time-oriented rule that sequences jobs according to planned release dates given by Equation (1) below. Two variants of this rule are used in the literature, where either waiting times or operation throughput times are treated as a constant. This rule was used, e.g. by Thürer *et al*. (2012).

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

t_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

a_i = constant for estimated waiting time at the i^{th} operation in the routing of a job

b_i = constant for estimated throughput time at the i^{th} operation in the routing of a job

The three rules above are all time-oriented, i.e. they use an urgency-based measure to prioritize jobs in the pool. One load-oriented rule that has been applied in the ConWIP literature is as follows:

- *Shortest Total Work Content (STWK),* a load-oriented rule that sequences jobs according to the sum of all processing times in the routing of an order. This rule was applied, e.g. by Leu (2000).

2.2 Capacity Slack-based Sequencing from the Workload Control Literature

Fredendall *et al.* (2010) and Thürer *et al.* (2015) recently demonstrated the potential for performance improvement from using a backlog-sequencing rule developed by Philipoom *et al.* (1993) – the Capacity Slack (CS) rule – in combination with Workload Control order release. In contrast to ConWIP, which limits the workload of the shop as a whole, Workload Control limits the workload at each station. Similarly, the CS rule considers the workload imbalance at each station. CS sequences jobs according to a capacity slack ratio given by Equation (2) below – the lower the capacity slack ratio of job *j* (S_j), the higher the priority of job *j*. The rule integrates three elements into one priority measure: the *workload contribution* of a job (i.e. the processing time of job *j* at operation *i*: p_{ij}); the *load gap,* (i.e. the difference between a pre-established load norm measured as the aggregate of the workload released to station *s* corresponding to operation *i*, N_s^A, and the current aggregate of the workload released to that station W_s^A: $N_s^A - W_s^A$); and, the *routing length* (i.e. the number of operations in the routing of job *j*: n_j), which is used to average the ratio between the load contribution and load gap elements over all operations in the routing of a job.

$$S_j = \frac{\sum_{i \in R_j} \frac{p_{ij}}{N_s^A - W_s^A}}{n_j} \tag{2}$$

The major difference between alternative capacity slack-based rules that can be found in the literature concerns the workload measures applied for calculating the load gap element. For example, Philipoom *et al.* (1993) and Fredendall *et al.* (2010) use the aggregate of the full processing times of jobs on the shop floor from release to completion at a station: the so-called aggregate load. Meanwhile, Thürer *et al.* (2015) corrected the processing times by dividing the processing time of an operation at a station by the station's position in a job's routing. This *corrected* aggregate load method (Oosterman *et al.*, 2000) recognizes that an order's contribution to a station's direct load is limited to only the proportion of time that an order is at the station. For example, an order's load contribution at the second station in its routing is set at 50% of the processing time at this station; similarly, its load contribution at the third station is set at 33.33%, and so on. In contrast to Workload Control, ConWIP does not measure workloads in full processing times or corrected processing times, but in terms of the number of jobs.

In the light of the above, four alternative CS rules will be considered.

- *Capacity Slack (CS)*, which uses full processing times to calculate the capacity slack S_j (Equation 2). This rule was applied, e.g. by Philipoom *et al*. (1993) and Fredendall *et al*. (2010).
- *Capacity Slack CORrected (CSCor)*, which uses corrected processing times to calculate the capacity slack S_j^C (Equation 3). This rule was applied, e.g. by Thürer *et al*. (2015).

$$S_j^C = \frac{\sum_{i \in R_j} \left(\frac{\frac{p_{ij}}{i}}{N_s^C - W_s^C} \right)}{n_j} \tag{3}$$

- *Capacity Slack number of jobs (CSjob)*, which uses the number of operations (this is the processing time set to unity) to calculate the capacity slack S_j^u (Equation 4). This rule is a specific adaption for ConWIP to reflect the fact that ConWIP measures the workload in the system in terms of the number of jobs rather than in processing times.

$$S_j^u = \frac{\sum_{i \in R_j} \left(\frac{1}{N_s^u - W_s^u} \right)}{n_j} \tag{4}$$

- *Capacity Slack number of jobs direct load (CSjobdir)*, which uses the direct load queuing at a station measured in terms of the number of jobs (i.e. the load that queues in front of a station) instead of the released workload (which measures the load from release to completion at a station, i.e. direct and indirect load) to calculate the capacity slack S_j^d (Equation 5).

$$S_j^d = \frac{\sum_{i \in R_j} \left(\frac{1}{N_s^d - W_s^d} \right)}{n_j} \tag{5}$$

Finally, since ConWIP does not limit the workload measure W_s at each station, the workload may exceed the limit N_s resulting in a negative priority value. This means that a capacity slack-based rule may prioritize an already overloaded station. Therefore, if the workload of a station is equal to or exceeds the workload norm, that is $N_s - W_s \leq 0$, then the job is positioned at the back of the queue by replacing the components $\left(\frac{p_{ij}}{N_s^A - W_s^A}\right)$, $\left(\frac{\frac{p_{ij}}{i}}{N_s^C - W_s^C}\right)$ $\left(\frac{1}{N_s^u - W_s^u}\right)$ and $\left(\frac{1}{N_s^d - W_s^d}\right)$ related to this station in the priority value S_j with M, where M is a sufficiently large number.

2.2.1 Modified Capacity Slack-based Backlog-Sequencing Rules

The Workload Control literature suggests that capacity slack-based backlog-sequencing rules can be a powerful means of improving load balancing. Hence, they have the potential to overcome a major weakness of ConWIP – its missing load balancing capability (Germs & Riezebos, 2010). However, Thürer *et al*. (2015) recently demonstrated that a sole focus on load balancing can be detrimental to performance, since large but urgent orders may never be released. It was found that load balancing should be restricted only to specific periods when many jobs are in the pool and, as a consequence, when many jobs are at risk of becoming tardy. In response, the authors proposed a modified capacity slack rule, which combines time-oriented and load-oriented sequencing into one rule. The rule uses a load-oriented capacity slack element to speed up the process when multiple jobs become urgent and a time-oriented PRD element to ensure that the overall mix of released jobs can be produced in time to satisfy due dates. This modified capacity slack (MODCS) rule can be summarized as follows:

(i) Jobs are divided into two classes: urgent jobs, i.e. jobs with a planned release date (refer back to Equation (1)) that equals the current date or has already passed; and non-urgent jobs. Urgent jobs always receive priority over non-urgent jobs.

(ii) Within the class of urgent jobs, jobs are sequenced according to a capacity slack rule.

(iii) Then, within the class of non-urgent jobs, jobs are sequenced according to the PRD rule.

Since we have four different capacity slack rules in this study depending on how the workload is measured (Equations 2 to 5 above), four different modified capacity slack rules will be considered: MODCS, MODCSCor, MODCSjob, and MODCSjobdir. Overall, twelve different backlog-sequencing rules will be considered (i.e. four rules from the ConWIP literature, four CS rules, and four modified CS rules), as will be summarized in Section 3.2 below.

3. Simulation Model

The shop and job characteristics modeled in the simulations are first outlined in Section 3.1. Section 3.2 then describes how ConWIP and the backlog-sequencing rules have been operationalized in the simulations. The priority dispatching rules applied on the shop floor are then discussed 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

A simulation model of a general flow shop (Oosterman *et al*., 2000) has been implemented using ARENA simulation software. The model is stochastic, whereby job routings, processing times, inter-arrival times and due dates are stochastic (random) variables. The shop contains six stations, where each station is a single constant capacity resource. As in previous studies on ConWIP, we consider the output to be fixed, thereby neglecting options for adjusting capacity (and thus the output rate), although this may often be a pre-requisite for the implementation of pull systems in practice. The routing length varies uniformly from one to six operations. All stations have an equal probability of being visited and a particular station is required at most once in the routing of a job. This means that we consider a balanced shop (in the long run) to avoid the effect of a stationary bottleneck. The resulting routing vector (i.e. the sequence in which stations are visited) is sorted for the general flow shop so that the routing is directed and there are typical upstream and downstream stations.

Operation processing times follow a truncated 2-Erlang distribution with a maximum of 4 time units and a mean of 1 time unit before truncation. Set up times are assumed to be included in the operation processing time. Meanwhile, the inter-arrival time of orders follows an exponential distribution with a mean of 0.648. Based on the number of stations in the routing of an order, this inter-arrival time 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 of the due date 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 waiting or queuing times of 6 time units. While any individual high-variety shop in practice will differ in many aspects to this stylized environment, it captures the typical shop characteristics of high routing variability, processing time variability, and arrival variability. Finally, Table 1 summarizes the simulated shop and job characteristics.

[Take in Table 1]

3.2 ConWIP

As in previous simulation studies on ConWIP (e.g. Hopp & Spearman, 1991; Bonvik *et al*., 1997; Herer & Masin, 1997; Jodlbauer & Huber, 2008), it is assumed that materials are available and all necessary information regarding due date, shop floor routing and processing times is known upon the arrival of an order in the pool. On arrival, jobs enter the pre-shop pool directly and await release according to ConWIP. Whenever the number of jobs on the shop floor is below a pre-established limit (WIP-Cap), jobs in the pool are sequenced according to the applied backlog-sequencing rule, and the next job in the sequence is released to the shop floor. This is repeated until the limit is reached or there are no jobs waiting to be released to the shop floor. Six limits on the number of jobs allowed in the system are applied: 30, 35, 40, 45, 50, and an infinite number of cards or jobs allowed.

The pre-established norm limit N_s that is used when calculating the priority measure for capacity slack-based backlog-sequencing rules changes according to the limit that is applied. It is given by the pre-established limit divided by the number of stations on the shop floor (six). This division is necessary because the WIP-Cap refers to the shop load while the workload limit used in capacity slack-based sequencing rules refers to each station. Finally, since ConWIP limits the number of jobs but CS and CSCor use processing times, N_s needs to be adapted. First, N_s^A is calculated based on the average processing time, which is one time unit per operation. Hence, a limit of 30 jobs on the whole shop floor translates to an N_s^A value of 30 divided by 6 time units (at each station *s*). Second, N_s^c is calculated based on the average processing time corrected by the average routing position of a station in the routing of jobs: 2.67. Thus, a limit of 30 jobs allowed in the shop translates into an N_s^c value of 30 divided by 6 divided by 2.67 time units.

Table 2 summarizes the twelve backlog-sequencing rules considered in this study and lists the parameters used for each. All of the sequencing rules identified (or derived) from the literature in Section 2.1 and Section 2.2 are considered.

[Take in Table 2]

3.3 Priority Dispatching Rule for the Shop Floor

ConWIP controls the work released to the shop floor, but it does not control the flow of work on the shop floor. Instead, the job that should be selected for processing next from the queue in front of a particular station is determined by a shop floor dispatching rule. In addition to First in System First Served (FSFS) dispatching, which was suggested by Spearman *et al*. (1990) and Hopp & Spearman (2001) and is used as a baseline measure in this study, three alternative dispatching rules will be considered: (i) the Operation Due Date (ODD) rule; (ii) the Shortest Processing Time (SPT) rule; and, (iii) the Modified Operation Due Date (MODD) rule (Baker & Kanet, 1983), which combines the SPT and ODD rules.

The calculation of the operation due date δ_{ij} for the i^{th} operation of a job j follows Equation (6) below. The operation due date for the last operation in the routing of a job is equal to the due date δ_j, while the operation due date of each preceding operation is determined by successively subtracting an allowance c from the operation due date of the next operation. This allowance is given by the running average of the actually realized operation throughput times at each station.

$$\delta_{ij} = \delta_j - (n_j - i) \cdot c \quad i{:}1 \ldots n_j \tag{6}$$

The ODD rule prioritizes jobs with the earliest operation due date. Meanwhile, the SPT rule selects the job with the shortest processing time from the queue. Finally, the MODD rule prioritizes jobs according to the lowest priority number, which is given by the maximum of the operation due date and earliest finish time. In other words, $max(d_{ij}, t+p_{ij})$ for an operation with processing time p_{ij}, where t refers to the time when the dispatching decision is made. The MODD rule shifts between a focus on ODDs to complete jobs on time and a focus on speeding up jobs – through SPT effects – during periods of high load, i.e. when multiple jobs exceed their ODD (Land *et al*., 2015).

3.4 Experimental Design and Performance Measures

The experimental factors are: (i) the six different levels of the number of jobs (or cards) allowed in the system; (ii) the 12 different backlog-sequencing rules; and, (iii) the four dispatching rules (FSFS, ODD, SPT, and MODD). A full factorial design with 288 scenarios was used, where each scenario was replicated 100 times. Results were collected over 13,000 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: *total throughput time* – the mean of the completion date minus the pool entry date across jobs;

percentage tardy – the percentage of jobs completed after the due date; *mean tardiness* – that is, $T_j = \max(0, L_j)$, with L_j being the lateness of job *j* (i.e. the actual delivery date minus the due date of job *j*); and, the *standard deviation of lateness*. In addition to these four main performance measures, we also measure the *shop floor throughput time* as an instrumental performance variable. This approach was introduced by Oosterman *et al.* (2000) and has been adopted in many subsequent studies on load-based order release (e.g. Germs & Riezebos, 2010; Thürer *et al*. 2012). While the total throughput time includes the time that an order waits before release, the shop floor throughput time only measures the time after release to the shop floor.

4. Results

Statistical analysis has been conducted by applying ANOVA to give a first indication of the relative impact of our three experimental factors: the backlog-sequencing rule, the dispatching rule, and the number of jobs (or cards) allowed in the system. The results are summarized in Table 3; all main effects, two-way interactions and three-way interactions are shown to be statistically significant.

[Take in Table 3]

The Scheffé multiple-comparison procedure was used to examine the significance of the differences between the outcomes of the individual backlog-sequencing and dispatching rules. Due to the large size of our experimental design in terms of the number of backlog-sequencing rules considered, these results are not shown here. But note that all of the backlog-sequencing rules except EDD and FCFS were significantly different for at least one performance measure, while the results for EDD and FCFS were statistically equivalent. Further, our four dispatching rules perform statistically different for all performance measures considered. These results are further evaluated in Section 4.1 and Section 4.2. Section 4.1 first provides detailed performance results for our backlog-sequencing rules under FSFS dispatching, which is our baseline measure. The robustness of the results to changes in the shop floor dispatching rule is then assessed in Section 4.2.

4.1 Performance Assessment of Backlog-sequencing Rules

To aid interpretation, the simulation results are presented in the form of performance curves. The left-hand starting point of the curves represents the lowest number of jobs (or cards) allowed (30 jobs). The number of cards allowed increases step-wise by moving from left to right in each graph, with each data point representing one card level (30, 35, 40, 45, 50, and infinite). Increasing the number of jobs in the system increases the level of work-in-process and, as a result, increases shop floor throughput times. Meanwhile, under infinite norms, jobs are not withheld in the pool meaning the backlog-sequencing rule is inactive, which results in all backlog-sequencing rules converging on the same point. Figures 2a and 2b show the total throughput time, percentage tardy, mean tardiness and standard deviation of lateness results over the shop floor throughput time results for the 'classical' ConWIP backlog-sequencing rules (from Section 2.1) and for the capacity slack-based backlog-sequencing rules (from Section 2.2), respectively. Only results with FSFS dispatching are shown in Figure 2 as the impact of the dispatching rule will be assessed in Section 4.2

[Take in Figure 2]

By comparing the results in Figure 2a for the classical ConWIP backlog-sequencing rules with Figure 2b for the capacity slack-based rules, it can be observed that capacity slack-based sequencing rules lead to shorter shop floor throughput times and shorter total throughput times. Further, they reduce the percentage tardy compared to classical ConWIP backlog-sequencing rules at all load limit levels and reduce the mean tardiness if the limits are tight. Meanwhile, CSjobdir also leads to a standard deviation of lateness performance that is comparable with classical time-oriented ConWIP rules. The two groups of backlog-sequencing rules are analyzed further below in sections 4.1.1 and 4.1.2, respectively.

4.1.1 Classical ConWIP Backlog-sequencing Rules (Figure 2a)

PRD performs worse than EDD and FCFS. PRD considers the routing length, i.e. the number of stations in the routing of jobs. As a result, the more stations there are in the routing of a job, the higher the priority of the job among jobs with similar due dates. As a result, total throughput times (and thus mean lateness) increase. This explains the increase in the percentage tardy and mean tardiness as the standard deviation of lateness is similar across PRD, EDD, and FCFS. ConWIP's release function does not consider job characteristics – the next job is simply released regardless of its characteristics (e.g. its routing or work content) if a card is available. In contrast, Workload Control limits the workload at each station and a job has to fit this limit at all stations – thus, Workload Control's release function considers job characteristics, which attenuates the negative effect observed for ConWIP and makes PRD a better choice than FCFS or EDD for Workload Control (see, e.g. Thürer *et al*., 2015, 2016b). Finally, typical shortest processing time effects can be observed for STWK, resulting in the shortest total throughput times and lowest percentage tardy across the four classical

ConWIP rules (FCFS, EDD, PRD, and STWK). This however is achieved at the expense of a higher standard deviation of lateness.

4.1.2 Capacity Slack-based Sequencing Rules (Figure 2b)

The best performance in terms of the total throughput time, mean tardiness, and the standard deviation of lateness is realized by CSjobdir, which uses the direct load queuing at each station (measured in terms of the number of jobs) to calculate the capacity slack. Moreover, CSjobdir reduces the shop floor throughput time and thus the work-in-process on the shop floor compared to alternative capacity slack-based sequencing rules. It appears that CSjobdir, which only considers the direct load queuing at a station, is more able to realize load balancing in the context of ConWIP than rules that consider both the direct load and the indirect load (i.e. the load on its way to a station). Limiting the number of ConWIP cards restricts the number of jobs; but this does not limit the workload, since a ConWIP card is not associated to a specific type of job. A limit of 30 jobs may, on one occasion, result in 30 jobs each with only one operation and a processing time of 0.1 time units (a limit of 3 time units) and, on another occasion, result in 30 jobs each with six operations and a processing time of four time units per operation (a limit of 720 time units). Similarly, a job may or may not have a particular station in its routing. Hence, there are significant fluctuations possible in terms of the workload queuing at each station, resulting in periods during which the workload measure W_s at each station exceeds the limit N_s and where the ratio between the load contribution and load gap elements is substituted by *M*, where *M* is a sufficiently large number. As a result, capacity slack rules are less effective specifically in high load periods, even though these are the periods when improved load balancing is needed the most (Land *et al*., 2015). Another interesting result is the poor performance of the modified capacity slack-based rules. Based on the results from Thürer *et al*. (2015) in the context of Workload Control, these rules should lead to better mean tardiness performance since they confine improved load balancing to periods when many jobs are in the pool, i.e. periods of high load. Yet, in our experiments, the modified capacity slack-based rules are consistently outperformed by basic capacity slack-based rules across all performance measures. It is argued here that this is due to the poor performance of the PRD backlog-sequencing rule.

4.2 Sensitivity Analysis: The Impact of the Shop Floor Dispatching Rule

As expected, ODD dispatching improves the percentage tardy, mean tardiness, and the standard deviation of lateness performance compared to FSFS dispatching. However, similar observations on performance differences between backlog-sequencing rules as in the context of FSFS dispatching can also be made for ODD dispatching. This can be observed from Figures 3a and 3b, which show the total throughput time, percentage tardy, mean tardiness and standard deviation of lateness results over the shop floor throughput time results under ODD dispatching for the 'classical' ConWIP backlog-sequencing rules and the capacity slack-based backlog-sequencing rules, respectively.

[Take in Figure 3]

SPT dispatching however not only leads to the lowest shop floor throughput time and total throughput time performance, it also changes the relative performance of the different backlog-sequencing rules. But the SPT effects created by the dispatching rule are so strong that the performance differences across backlog-sequencing rules become arguably negligible. This can be observed from Figures 4a and 4b, which show the total throughput time, percentage tardy, mean tardiness, and standard deviation of lateness results over the shop floor throughput time results under SPT dispatching for the 'classical' ConWIP backlog-sequencing rules and the capacity slack-based backlog-sequencing rules, respectively.

[Take in Figure 4]

Finally, Figures 5a and 5b show the total throughput time, percentage tardy, mean tardiness, and standard deviation of lateness results over the shop floor throughput time results under MODD dispatching for the 'classical' ConWIP backlog-sequencing rules and the capacity slack-based backlog-sequencing rules, respectively. MODD switches between SPT dispatching and ODD dispatching. As a result, while performance differences between the backlog-sequencing rules are maintained (compared to those under ODD sequencing), the relative performance differences diminish. In general, MODD leads to the best performance across the four dispatching rules considered in this study.

[Take in Figure 5]

5. Conclusions

Constant Work-In-Process (ConWIP) is a simple card-based control system. Although it has been effective in simple flow shops, its lack of load balancing capabilities has hindered its application to more complex production environments (Germs & Riezebos, 2010; Thürer *et al*., 2016a). Load balancing is a key function of Workload Control, a production planning and control concept specifically designed for high-variety make-to-order environments. In the context of Workload Control, capacity slack-based backlog-sequencing rules – i.e. the greedy heuristics that determine the sequence in which jobs are considered for release from the pool – were recently found to significantly enhance workload balancing capabilities (Thürer *et al*., 2015). Based on this finding from the Workload Control literature, we have asked: *Can a backlog-sequencing rule be used to extend the applicability of ConWIP to high-variety make-to-order flow shops?* Using a simulated general flow shop environment with high variability in terms of job arrivals, processing times, and routings, we have shown that the backlog-sequencing decision has a significant impact on the performance of ConWIP. More specifically, it has been shown that capacity slack-based rules maintain their ability to significantly improve load balancing. They provide ConWIP with load balancing capabilities that result in lower total throughput times while simultaneously reducing the percentage of tardy orders. Thus, they provide an important means of extending the applicability of ConWIP to more complex high-variety contexts.

5.1 Managerial Implications, Limitations and Future Research

This paper has demonstrated that capacity slack-based backlog-sequencing rules have the potential to significantly improve the performance of ConWIP when compared to 'classical' ConWIP backlog-sequencing rules. A major explanation for this is the use of load information from the shop floor. This may be a factor that argues against the use of these rules in practice since this information is sometimes difficult to obtain; however, ConWIP is typically embedded in a higher level planning system that provides this data. Moreover, with the advent of new technology, such as the Internet of Things, it is argued that this data is becoming more readily available. In fact, a major future research direction may be the exploration of how increased data availability can be used to enhance the performance of simple control systems without jeopardizing their simplicity. Note that the simplicity of ConWIP is in fact not affected by changes to the backlog-sequencing decision as, on the shop floor, an operator continues to process the next job in the sequence regardless of the rule applied. In other words, any increase in sophistication is decoupled from the shop floor.

A major limitation of our study however is that, in order to keep the experimental settings reasonable, it has focused on a limited set of job and shop characteristics. This is justified by the broad set of backlog-sequencing and dispatching rules considered, i.e. we chose to extend our experimental setting in terms of the number of rules considered rather than the number of environmental variables. Nonetheless, future research could explore the impact of other environmental variables on the relative performance of the backlog-sequencing rules. Finally, our results revealed two unexpected findings. First, that CSjobdir, which only considers the direct load, performs better than capacity slack-based rules that incorporate a proportion of the indirect load. In response, future research should explore whether only considering the direct load also enhances the performance of Workload Control. Second, that the poor performance of PRD sequencing leads to poor overall performance for modified capacity slack-based rules. Future research is therefore required to identify a more appropriate time-oriented rule than PRD that unlocks the potential of modified capacity slack-based rules in the context of ConWIP.

References

Baker, K.R., and Kanet, J.J., 1983, Job shop scheduling with modified due dates, *Journal of Operations Management*, 4, 1, 11-22.

Bonvik, A.M., Couch, C.E. and Gershwin, S.B., 1997, A comparison of production-line control mechanisms, *International Journal of Production Research*, 35, 3, 789-804.

Cao, D. and Chen, M., 2005, A mixed integer programming model for a two line CONWIP-based production and assembly system, *International Journal of Production Economics*, 95, 3, 317–326.

Framinan, J. M., Ruiz-Usano, R., and Leisten, R., 2001, Sequencing CONWIP Flow-shops: Analysis and Heuristics, *International Journal of Production Research*, 39, 12, 2735–2749.

Framinan, J.M., Gonzalez, P.L., and Ruiz-Usano, R., 2003, The CONWIP production control system: Review and research issues, *Production Planning & Control*, 14, 3, 255-265.

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.

Germs, R., and Riezebos, J., 2010, Workload balancing capability of pull systems in MTO production, *International Journal of Production Research*, 48, 8, 2345-2360.

Golany, B. Dar-El, E.M., and Zeev, N., 1999, Controlling shop floor operations in a multi-family, multi-cell manufacturing environment through constant work-in-process, *IIE Transactions*, 31, 8, 771-781

Herer Y. T. and Masin M., 1997, Mathematical programming formulation of CONWIP based production lines; and relationships to MRP, *International Journal of Production Research,* 35, 4, 1067-1076.

Hopp, W.J. and Spearman M.L., 1991, Throughput of a constant working process manufacturing line subject to fails, *International Journal of Production Research*, 29, 3, 635- 655.

Hopp, W.J. and Spearman M.L., Factory Physics: Foundations of Manufacturing Management, Irwin/McGraw-Hill, 2001.

Hopp, W.J., and Spearman, M.L., 2004, To pull or not to pull: What is the question?, *Manufacturing & Service Operations Management*, 6, 2, 133 – 148.

Jodlbauer, H. and Huber, A., 2008, Service-level performance of MRP, kanban, CONWIP and DBR due to parameter stability and environmental robustness, *International Journal of Production Research*, 46, 8, 2179–2195.

Land, M.J., 2009, Cobacabana (control of balance by card-based navigation): A card-based system for job shop control, *International Journal of Production Economics*, 117, 97-103

Land, M.J., Stevenson, M., Thürer, M., and Gaalman, G.J.C., 2015, Job Shop Control: In Search of the Key to Delivery Improvements, *International Journal of Production Economics*, 168, 257-266.

Leu, B.Y., 2000, Generating a backlog list for a CONWIP production line: A simulation study, *Production Planning & Control*, 11, 4, 409-418.

Li, J.W., 2011, Comparing Kanban with CONWIP in a make-to-order environment supported by JIT practice, *Journal of the Chinese Institute of Industrial Engineers*, 28, 1, 72–88

Lingayat, S., Mittenthal, J., and O'Keefe, R.M., 1995, An order release mechanism for a flexible flow system, *International Jorunal of Production Research*, 33, 5, 1241-1256.

Liu, C.H., 2010, A coordinated scheduling system for customer orders scheduling problem in job shop environments, *Expert Systems with Applications*, 37, 7831–7837

Luh, P.B., Zhou, X. and Tomastik, R.N., 2000, An effective method to reduce inventory in job shops, *IEEE Transactions on Robotics and Automation* 16,4, 420 - 424

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.

Philipoom, P.R., Malhotra, M.K., and Jensen, J.B., 1993, An evaluation of capacity sensitive order review and release procedures in job shops, *Decision Sciences*, 24, 6, 1109-1133.

Plossl, G.W., and Wight, O.W., 1971, Capacity planning and control, *Working paper presented at the APICS International Conference in St.Louis, Missouri*.

Prakash, J., and Chin, J.F., 2015, Modified CONWIP systems: a review and classification, *Production Planning & Control*, 26, 4, 296-307.

Rose, O., 1999, CONLOAD – A New Lot Release Rule for Semiconductor Wafer Fabs, In *Proceedings of the 1999 Winter Simulation Conference*, 850–855.

Ryan, S.M., and Vorasayan, J., 2005, Allocating work in process in a multiple-product CONWIP system with lost sales, *International Journal of Production Research*, 43, 2, 223–246.

Spearman, M.L., Woodruff, D.L., and Hopp, W.J., 1990, CONWIP: a pull alternative to kanban, *International Journal of Production Research*, 28, 5, 879-894.

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., and Protzman, C.W., 2016a, Card-Based Production Control: A Review of the Control Mechanisms Underpinning Kanban, ConWIP, POLCA and COBACABANA Systems, *Production Planning & Control*, (in print)

Thürer, M., Stevenson, M., and Qu, T., 2016b, Job Sequencing and Selection within Workload Control Order Release: An Assessment by Simulation, *International Journal of Production Research*, 54, 4, 1061-1075.

Thürer, M., Land, M.J., Stevenson, M., Fredendall, L.D., and Godinho Filho, M., 2015, Concerning Workload Control and Order Release: The Pre-Shop Pool-sequencing Decision, *Production & Operations Management*, 24, 7, 1179–1192.

Thürer, M., Land, M.J., and Stevenson, M., 2014, Card-Based Workload Control for Job Shops: Improving COBACABANA, *International Journal of Production Economics*, 147, 180-188.

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.

Wight, O., 1970, Input/Output control a real handle on lead time, *Production & Inventory Management Journal*, 11, 3, 9-31.

Woodruff D.L. and Spearman M.L., 1992, Sequencing and batching for two classes of jobs with deadlines and setup times, *Production & Operations Management*, 1, 1, 87-102

Zhang, W., and Chen, M., 2001, A mathematical programming model for production planning using CONWIP, *International Journal of Production Research*, 39, 2723–2734.

Tables & Figures

Table 1: Summary of Simulated Shop and Job Characteristics

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

Table 2: Summary of the Twelve Backlog-Sequencing Rules Applied in this Study

Type	Abbr.	Full Name	Brief Description	Parameter
'Classical' ConWIP Rules	FCFS	First-Come-First-Served	Time-oriented. The job that arrived in the pool first is considered for release first.	None
	EDD	Earliest Due Date	Time-oriented. The job with the earliest due date is considered for release first.	None
	PRD	Planned Release Date	Time-oriented. The job with the earliest planned release date is considered for release first.	b[1)]
	STWK	Shortest Total work Content	Load-oriented. The job with the shortest total work content is considered for release first.	None
Capacity Slack (CS) Based Rules	CS	Capacity Slack	Load-oriented. The job with the lowest capacity slack ratio (see Eq. 2) based on the aggregate load (from release to completion) is considered for release first.	None
	CSCor	Capacity Slack Corrected	Load-oriented. The job with the lowest capacity slack ratio (see Eq. 3) based on the corrected aggregate load measure is considered for release first.	None
	CSjob	Capacity Slack Number of Jobs	Load-oriented. The job with the lowest capacity slack ratio (see Eq. 4) based on the aggregate load measured in the number of jobs is considered for release first.	None
	CSjobdir	Capacity Slack Number of Jobs direct	Load-oriented. The job with the lowest capacity slack ratio (see Eq. 5) based on the direct load (i.e. the load queuing in front of a station) measured in the number of jobs is considered for release first.	None
	MODCS	Modified Capacity Slack	Time-oriented and load-oriented. Jobs are divided into two classes: urgent, i.e. jobs with a planned release date that equals or has already passed the current date; and non-urgent. Urgent jobs are considered for release first according to the CS rule. Non-urgent jobs are then considered according to the PRD rule.	b[1)]
	MODCSCor	Modified Capacity Slack Corrected	Time-oriented and load -oriented. Jobs are divided into two classes: urgent, i.e. jobs with a planned release date that equals or has already passed the current date; and non-urgent. Urgent jobs are considered for release first according to the CSCor rule. Non-urgent jobs are then considered according to the PRD rule.	b[1)]
	MODCSjob	Modified Capacity Slack Number of Jobs	Time-oriented and load -oriented. Jobs are divided into two classes: urgent, i.e. jobs with a planned release date that equals or has already passed the current date; and non-urgent. Urgent jobs are considered for release first according to the CSjob rule. Non-urgent jobs are then considered according to the PRD rule.	b[1)]
	MODCSjobdir	Modified Capacity Slack Number of Jobs direct	Time-oriented and load-oriented. Jobs are divided into two classes: urgent, i.e. jobs with a planned release date that equals or has already passed the current date; and non-urgent. Urgent jobs are considered for release first according to the CSjobdir rule. Non-urgent jobs are then considered according to the PRD rule.	b[1)]

[1)] Running average of the realized operation throughput time

Table 3: ANOVA Results

	Source of Variance	Sum of Squares	df[1]	Mean Squares	F-Ratio	p-Value
Total Throughput Time	Sequencing Rule (S)	39267.71	11.00	3569.79	241.04	0.00
	Dispatching Rule (D)	781062.82	3.00	260354.27	17580.04	0.00
	Number ConWIP Cards (C)	140759.63	5.00	28151.93	1900.92	0.00
	S x D	15503.88	33.00	469.81	31.72	0.00
	S x C	56641.91	55.00	1029.85	69.54	0.00
	C x D	52619.15	15.00	3507.94	236.87	0.00
	S x C x D	22659.20	165.00	137.33	9.27	0.00
	Error	422252.71	28512.00	14.81		
Percentage Tardy	Sequencing Rule (S)	36.71	11.00	3.34	1403.28	0.00
	Dispatching Rule (D)	22.32	3.00	7.44	3128.67	0.00
	Number ConWIP Cards (C)	5.68	5.00	1.14	477.43	0.00
	S x D	13.14	33.00	0.40	167.50	0.00
	S x C	34.76	55.00	0.63	265.80	0.00
	C x D	2.21	15.00	0.15	61.86	0.00
	S x C x D	12.82	165.00	0.08	32.68	0.00
	Error	67.80	28512.00	0.00		
Mean Tardiness	Sequencing Rule (S)	6457.02	11.00	587.00	80.83	0.00
	Dispatching Rule (D)	9592.39	3.00	3197.46	440.30	0.00
	Number ConWIP Cards (C)	83017.91	5.00	16603.58	2286.36	0.00
	S x D	3130.80	33.00	94.87	13.06	0.00
	S x C	12741.41	55.00	231.66	31.90	0.00
	C x D	34632.53	15.00	2308.84	317.93	0.00
	S x C x D	6606.63	165.00	40.04	5.51	0.00
	Error	207054.57	28512.00	7.26		
SD Late	Sequencing Rule (S)	1129576.40	11.00	102688.77	332.11	0.00
	Dispatching Rule (D)	278447.82	3.00	92815.94	300.18	0.00
	Number ConWIP Cards (C)	3518943.80	5.00	703788.76	2276.15	0.00
	S x D	459650.38	33.00	13928.80	45.05	0.00
	S x C	1695157.70	55.00	30821.05	99.68	0.00
	C x D	1414437.60	15.00	94295.84	304.97	0.00
	S x C x D	736826.31	165.00	4465.61	14.44	0.00
	Error	8815969.50	28512.00	309.20		

[1]) degrees of freedom

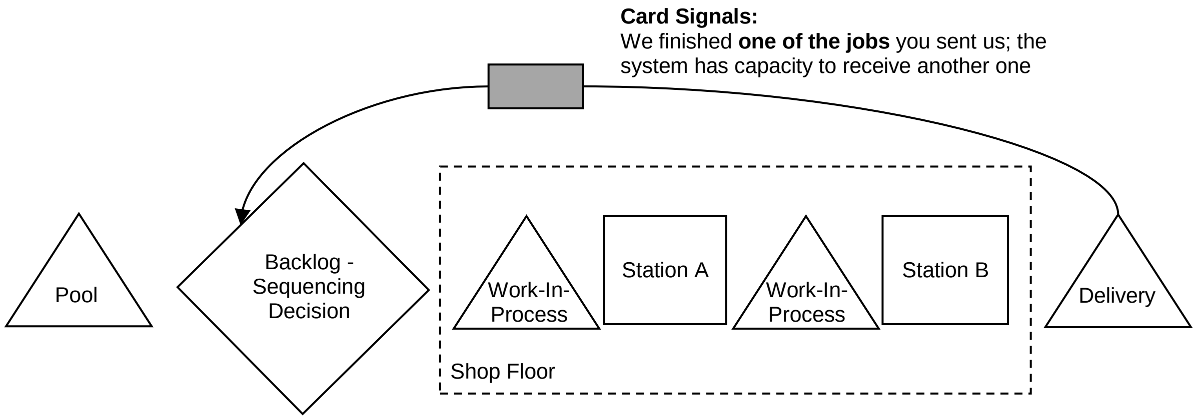
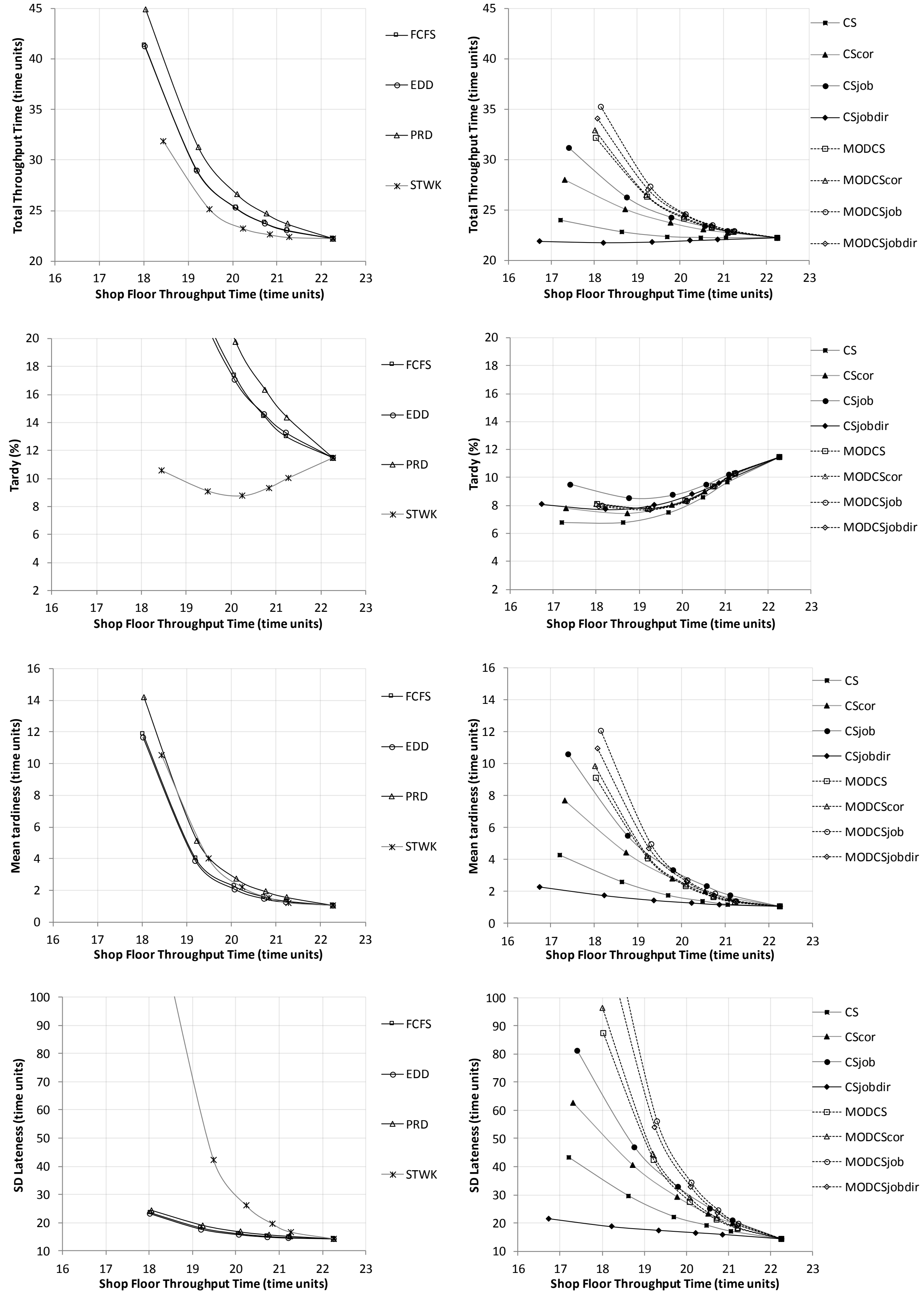
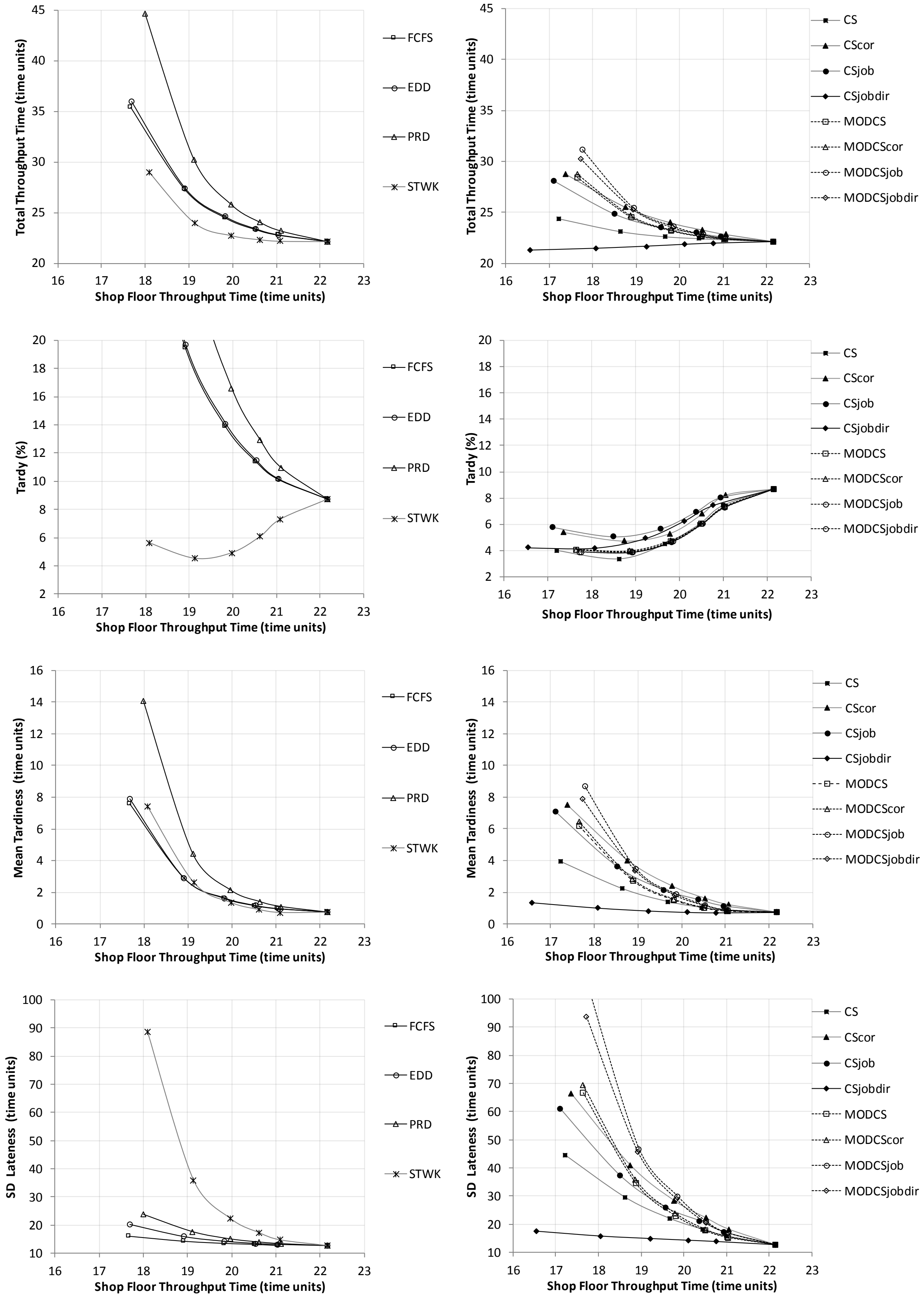


Figure 1: Illustration of a 2-Station ConWIP System with Backlog-Sequencing Decision



(a) (b)

Figure 2: Performance Results for Alternative Backlog-Sequencing Rules under FSFS Shop Floor Dispatching



(a) (b)

Figure 3: Performance Results for Alternative Backlog-Sequencing Rules under ODD Shop Floor Dispatching

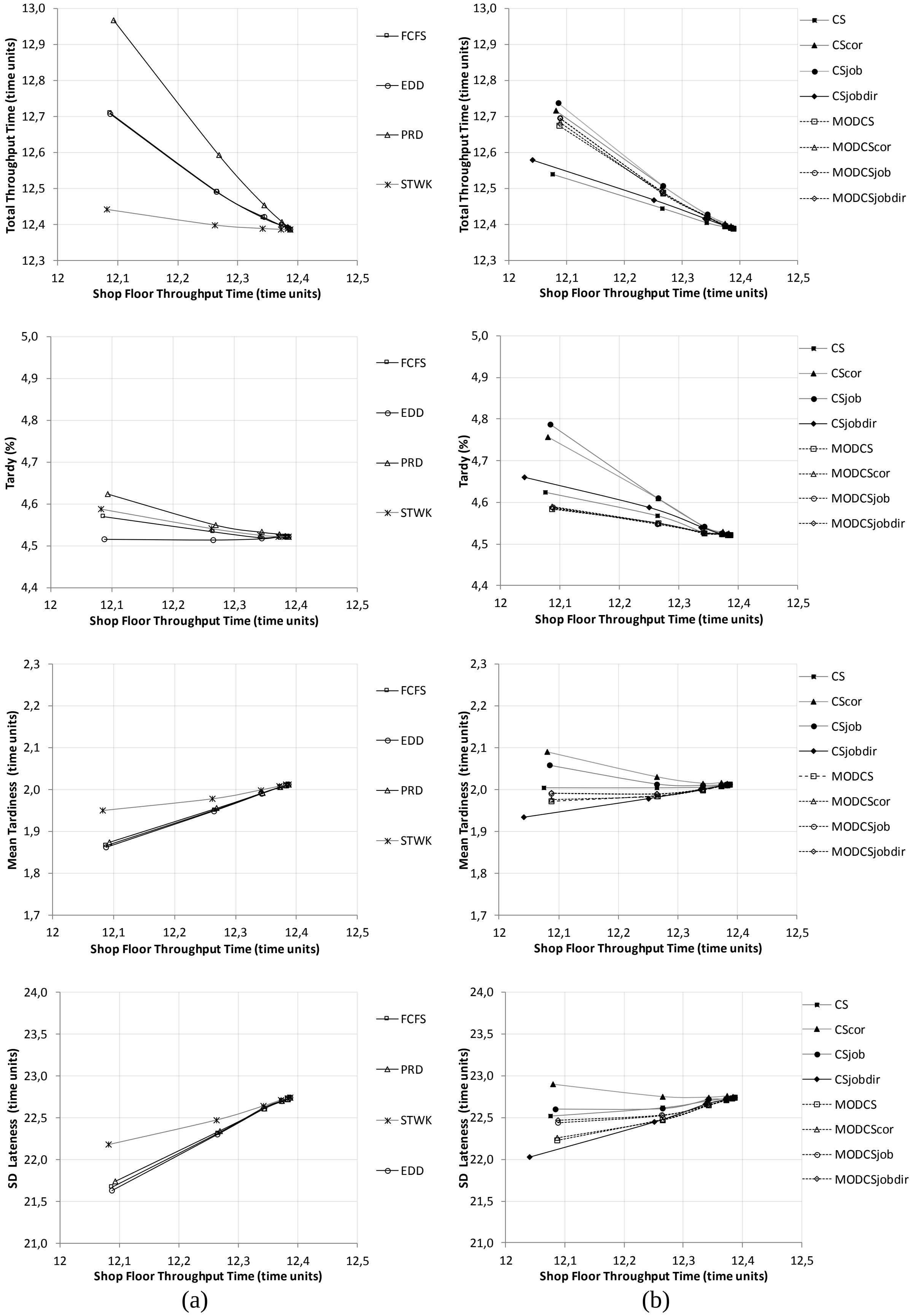
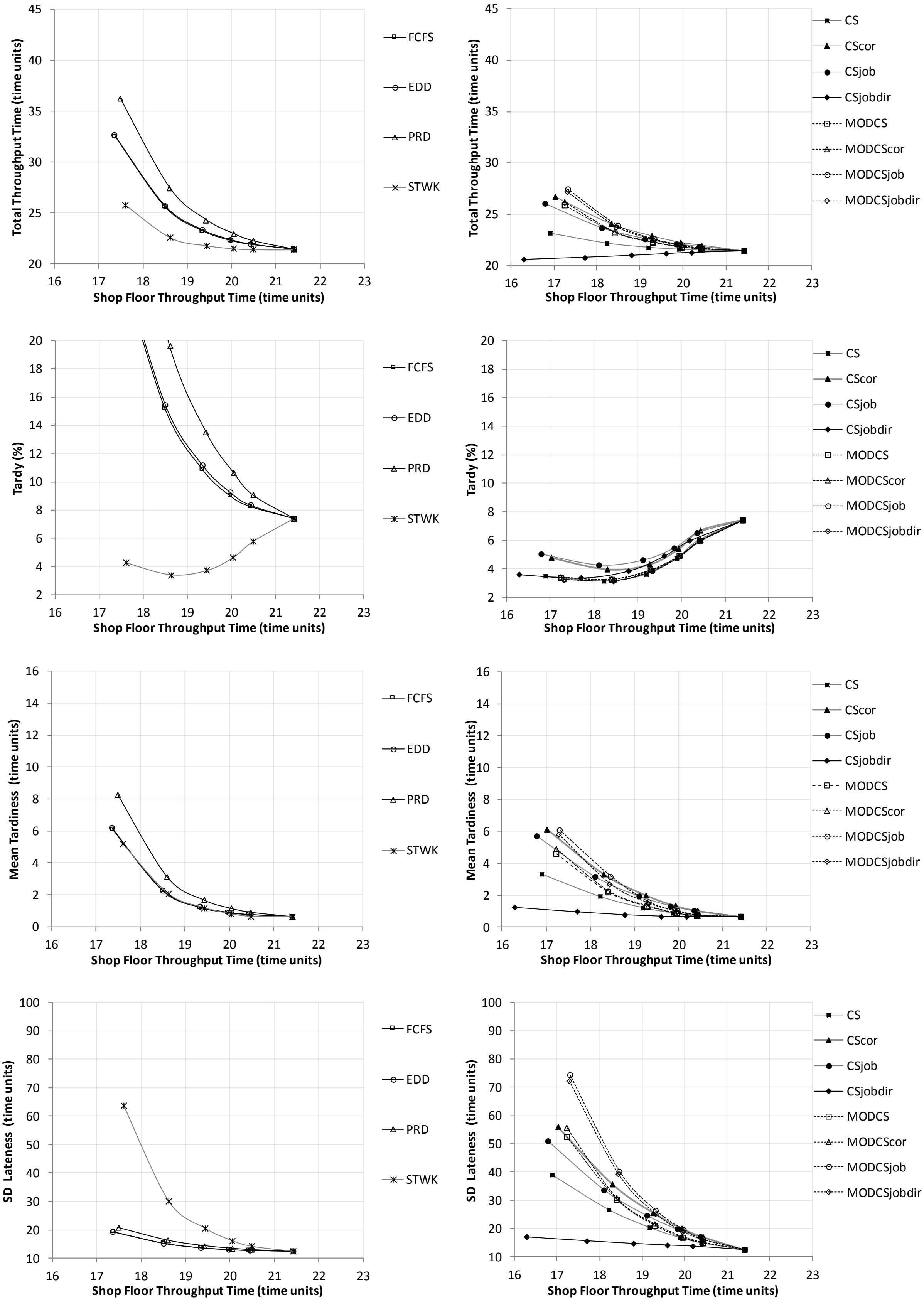


Figure 4: Performance Results for Alternative Backlog-Sequencing Rules under SPT Shop Floor Dispatching



(a) (b)

Figure 5: Performance Results for Alternative Backlog-Sequencing Rules under MODD Shop Floor Dispatching