Load-based ConWIP: An Assessment by Simulation

Matthias Thürer (matthiasthurer@workloadcontrol.com)
Jinan University, PR China

Nuno Fernandes
Instituto Politécnico de Castelo Branco, Portugal

Nick Ziengs
Groningen University, The Netherlands

Mark Stevenson
Lancaster University, United Kingdom

Ting Qu
Jinan University, PR China

Abstract

Constant Work-in-Process (ConWIP) is a simple production control system. There are arguably only two major search directions to improve the concept: (i) to alter the meaning of cards; and, (ii) to adopt alternative backlog sequencing rules. In this study, we follow the first search direction. We argue that changing the meaning of cards away from anonymous jobs to a workload contribution can address load balancing issues caused by processing time variability. Simulation results demonstrate the positive performance impact of limiting the total shop load instead of the number of jobs.

Keywords: ConWIP, Workload Control, Simulation

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 that uses a Work-In-Process (WIP) limit or cap (WIP-Cap) to realize input/output control. In accordance with input/output control, the output of work from the shop floor determines the input of work to the shop floor. Jobs are only permitted to enter the shop floor if the WIP-Cap, which is pre-established by management, is not violated; otherwise, they have to wait in a so-called 'backlog' (Spearman *et al*., 1990) until a job on the shop floor has been completed. Cards circulate between the exit from the shop floor and the backlog or entry point. The return of a card signals that one job has been completed (output) and another can be released (input).

ConWIP is an effective means of exercising pull control providing that product variety is restricted – its applicability to high-variety make-to-order environments is therefore rather limited. There are two key reasons for this: (i) ConWIP's simple loop structure, which contains all possible stations in the routing of jobs within one loop, requires short routings and complete routing homogeneity to ensure effective control; and (ii) ConWIP's lack of load balancing capabilities requires low levels of processing time variability. These two weaknesses have been a key focus of the extant literature on ConWIP. For example, a backlog sequencing rule has been used to enhance ConWIP's ability to balance the workload across resources. In this study, we extend this literature by arguing that further improvement can be obtained by changing the meaning of cards such that they represent a certain contribution to the workload.

The original ConWIP cards were job anonymous, i.e. they signal that a job can be released but they did not indicate what kind of job (Spearman *et al.*, 1990). The motivation behind this was that product specific cards, as used in *kanban* systems, require a large number of cards to be managed and maintained when product variety is high. Thus by making cards job anonymous, only a single card type was needed. This unique characteristic of ConWIP was questioned by Duenyas (1994) who introduced m-ConWIP. In m-ConWIP, cards are again product specific (like *kanban* cards). This overcomes the restrictions on routing variability for the original ConWIP system and even led to improvements in terms of load balancing in Germs & Riezebos (2010). However, it re-introduces the limitation on product variety. Moreover, it is argued here that this adaptation only addressed the lack of load balancing capability that is caused by routing variability and not in terms of processing time variability. Load balancing is here defined as the balancing of the workload across resources and is thus also influenced by variety in the workload of jobs and not just the routing.

We argue that changing the meaning of cards away from anonymous jobs to a workload contribution can address load balancing issues caused by processing time variability while at the same time maintaining the advantage of ConWIP, allowing for high product variety. The objective of this study is twofold. First, we outline a simple, practical load-based ConWIP system that changes the meaning of cards. Second, we use controlled simulation experiments to assess the potential of this refinement to improve the performance of ConWIP in a general flow shop that produces to-order; i.e. a type of shop environment characterized by high product variety, high routing variety, and high processing time variability.

Background – The ConWIP Production Control System

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 on the shop floor) is below a pre-established limit, a new job is released to the system.

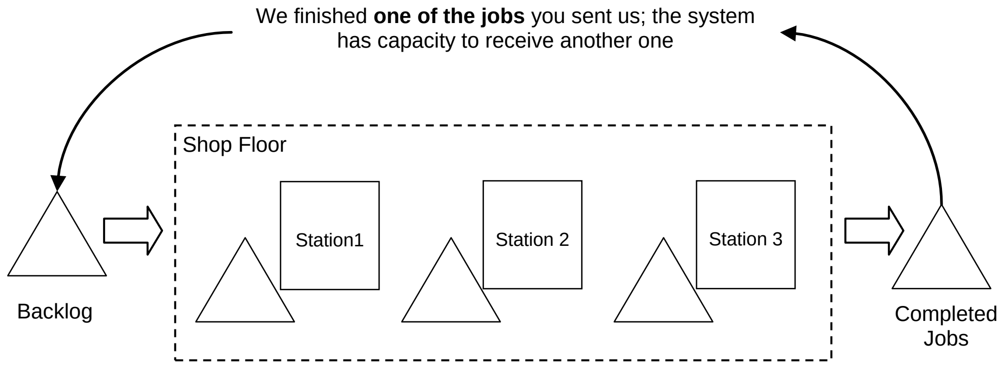

We finished **one of the jobs** you sent us; the system has capacity to receive another one
Shop Floor
Station1
Station 2
Station 3
Backlog
Completed Jobs

Figure 1 – Constant Work-in-Process (ConWIP)

Since the loop structure of ConWIP cannot be changed without creating a different card-based control system, there are arguably only two major search directions for performance improvement: (i) to alter the meaning of cards away from controlling jobs; and (ii) to adopt alternative backlog sequencing rules when considering jobs for release. These search directions will be discussed next.

Altering the Meaning of cards

ConWIP uses a single loop to control the input of work to the shop floor. This has two important consequences for the routing characteristics that can be accommodated by ConWIP (Hopp & Spearman, 2001): (i) there should not be too many stations contained in the loop; and, (ii) the routing of jobs should not differ (in other words, lines should not be split). An alternative ConWIP system designed to overcome the latter shortcoming is the m-ConWIP system that makes ConWIP loops product specific (Duenyas, 1994; Framinan *et al*., 2000). Product specific means that the system signals the requirement for a specific component. In other words, if there are four different types of jobs, then a specific m-ConWIP card is associated with each job type and, as a consequence, four independent m-ConWIP loops exist.

While the switch from job anonymous cards to product specific cards allows routing variability to be accommodated and improves load balancing capability (Germs & Riezebos, 2010), there are at least two weaknesses. First, m-ConWIP does not work in high-variety contexts as jobs cannot typically be grouped into a restricted number of specific job types; as a result, a large number of m-ConWIP cards and associated loops must be maintained, and this leads to the same criticisms as those leveled on the *kanban* system that triggered the development of ConWIP in the first place. Second, m-ConWIP does not address processing time variability. Both ConWIP and m-ConWIP neglect the actual workload contributions of jobs, which hinders effective load balancing if work content varies.

Backlog Sequencing Rules

One means of realizing load balancing in high-variety contexts is via the backlog sequencing decision, which determines the sequence in which orders are released to the system. Previous studies on the 'backlog pool-sequencing problem' have often focused on complex optimization algorithms. In this body of work, a fixed set of orders has been assumed and the sequence in which those orders should be released by a ConWIP system has been determined to optimize a certain set of performance parameters. However, in a make-to-order system, job arrivals follow a stochastic process and jobs may arrive at any moment in time. In response, Thürer *et al*. (2017) assessed the impact of a simple greedy heuristic, i.e. a simple backlog sequencing rule. Thürer *et al*. (2017) showed that a capacity slack-based sequencing rule, which averages the capacity slack (i.e. the difference between a target workload and the actual workload released to a station) across stations in the routing of a job, has the potential to enhance ConWIP's load balancing capability.

Capacity slack-based sequencing is however rather complex and requires a significant amount of feedback from the shop floor. It remains to be established whether there are other simple means of improving load balancing in the context of ConWIP. While the loop structure itself cannot be changed without creating a different card-based control system altogether, the meaning of cards can effectively be changed – and this will be discussed next.

Load-based ConWIP: Changing the Meaning of Cards

In this study, we propose that a ConWIP card should be adapted such that it represents a measure of workload rather than a job. We ask: Can ConWIP performance be improved by associating ConWIP cards with a workload? Figure 2 illustrates how our refinement can be operationalized in practice.

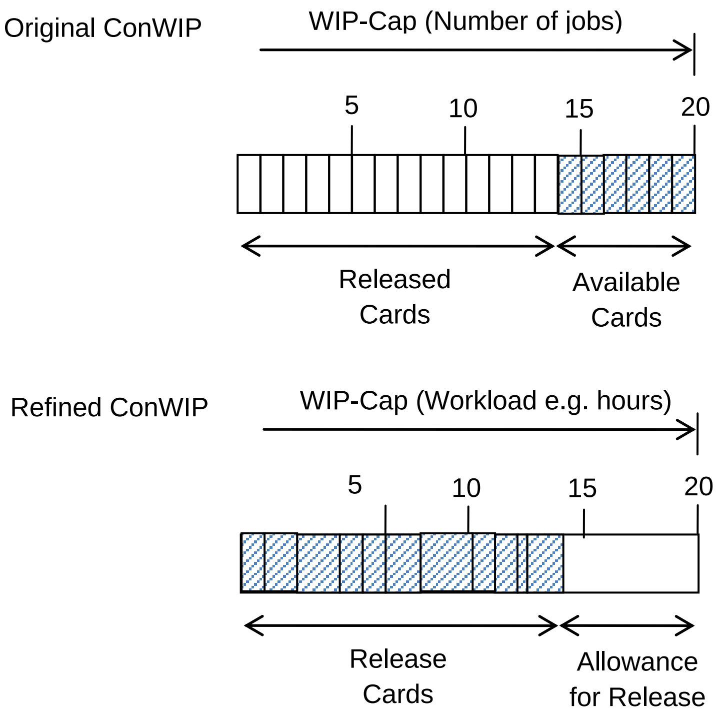


Figure 2 – Changing the Meaning of Cards

Based on the refinement to COBACABANA proposed in Thürer *et al.* (2014), we invert the meaning of cards. While in the original ConWIP system having a card available at release signals that a job can be released, in our refined ConWIP system a card represents a workload. As a consequence, cards need to be duplicated. One card, the release card, is used to represent the released workload while the second card, the operations card, travels with the order and signals its completion. Once a job is completed, the release card is withdrawn. Release cards are cut to the size of a job's workload contribution. The stack of release cards then represents the workload on the shop floor. A new job can only be released if its workload contribution does not violate the WIP-Cap.

But the question remains – which type of workload measure should be applied? Workload Control is an alternative production control concept that focuses on the workload (see, e.g. Thürer *et al.* (2011) for a review). We therefore refer to Workload Control theory to identify suitable workload measures to embed within our refined version of ConWIP. Four workload measures will be considered in this study as follows:

- *The number of jobs*: this is the original ConWIP system;
- *The shop load*: this is the total workload of all jobs on the shop floor;
- *The shop load corrected by the routing length*: this is the workload of all jobs on the shop floor where the load contribution of each job is divided by its routing length; and,

- *The shop load corrected by the routing position*: this is the workload of all jobs on the shop floor where the load contribution of each job's operation(s) is divided by its routing position.

Controlled simulation experiments will next be used to assess the performance impact of these different workload measures in the context of ConWIP. The following section outlines the simulation model used.

Simulation Model

Overview of Modeled Shop and Job Characteristics

A simulation model of a general flow shop has been implemented using ARENA simulation software. Our 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. 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. 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 considered part of the operation processing time. Meanwhile, the inter-arrival time of orders follows an exponential distribution with a mean of 0.648, which – based on the number of stations in the routing of an order – 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. These settings have been chosen to facilitate comparisons with earlier studies on ConWIP (e.g. Thürer *et al.* 2017). While any individual high-variety shop in practice will differ in many aspects from this stylized environment, it captures the typical shop characteristics of high routing variability, processing time variability, and arrival variability. Finally, Table 2 summarizes the simulated shop and job characteristics.

ConWIP

As in previous simulation studies on ConWIP, it is assumed that materials are available and all necessary information regarding shop floor routing, processing times, etc. is known upon the arrival of an order at the shop. On arrival, jobs directly enter the backlog and await release. Six limits are applied if the WIP-Cap is the number of jobs: 30, 35, 40, 45, 50 and an infinite number of cards or jobs allowed. The same WIP-Cap, but in terms of work content, can also be applied for the shop load corrected by the routing length. However, for the shop load and the shop load corrected by the routing position, the limit has to be multiplied by the average work content of jobs.

Finally, in this study four backlog sequencing rules are applied: First Come First Served (FCFS), Shortest Total Work Content (STWK), Capacity Slack (CS) and Capacity Slack number of jobs direct load (CSjobdir). The choice of rules is based on recent results in Thürer *et al.* (2017).

Priority Dispatching Rule for the Shop Floor

ConWIP controls the work released to the shop floor; 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 this study, the Modified Operation Due Date (MODD) rule is used since it was arguably the best performing rule in Thürer *et al*. (2017).

Experimental Design and Performance Measures

The experimental factors are: the four different backlog sequencing rules; the four different measures of the workload; and the six levels of WIP-Cap. A full factorial design was used with 96 (4*4*6) scenarios, where each scenario was replicated 100 times. All results were collected over 13,000 time units following a warm-up period of 3,000 time units. These parameters allow us to obtain stable results while keeping the simulation run time to a reasonable level. Three main performance measures are considered in this study as follows: the *total throughput time* – the mean of the completion date minus the pool entry date across jobs; the *percentage tardy* – the percentage of jobs completed after the due date; and the *mean tardiness*. In addition, we also measure the average shop floor throughput time as an instrumental performance variable. While the total throughput time includes the time that an order waits before being released, the shop floor throughput time only measures the time after an order is released to the shop floor.

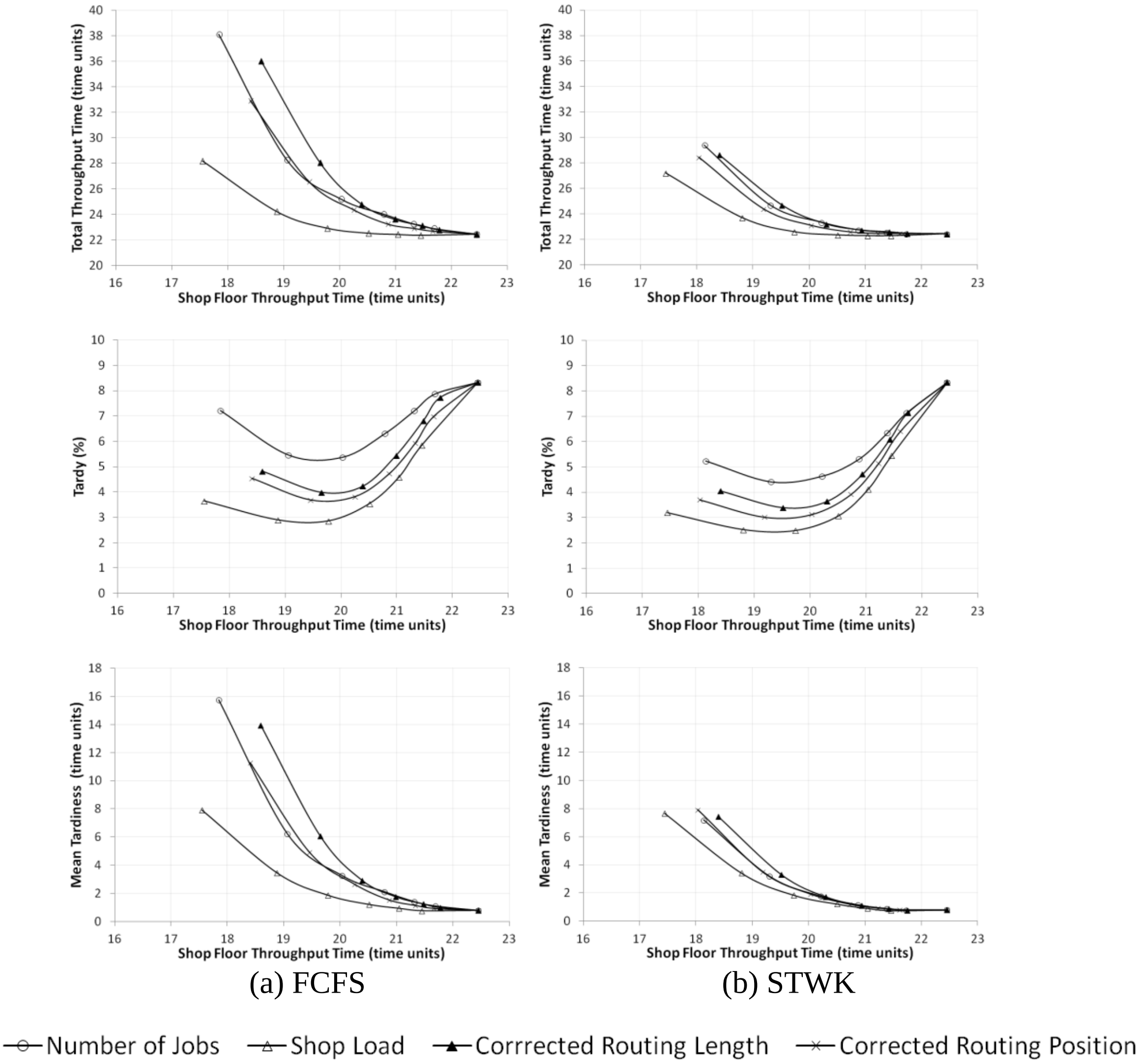
Results

Detailed results are presented in Figure 3a and Figure 3b for FCFS and STWK. Meanwhile, Figure 4a and Figure 4b present the results for our capacity slack based sequencing rules, CS and CSjobdir, respectively. Rather than comparing one specific parameter setting, parameters are varied for each policy and the results presented in the form of performance curves. These performance or operating characteristic curves are an important means of obtaining a ‘fair’ comparison across different control policies.

The relative positioning of the different curves (where each curve represents one policy) allows the performance of each policy to be compared. The left-hand starting point of each curve represents the tightest WIP-Cap. The WIP-Cap increases step-wise by moving from left to right, with each data point representing one level of WIP-Cap. Loosening the cap increases the workload level and, as a result, throughput times on the shop floor become longer. On the far right are the results for infinite load norms or no limit. This single point is located to the right of the curves as it leads to the longest throughput times on the shop floor.

In terms of the direct impact of our workload measures and their interaction with the backlog sequencing rule, the following can be observed from the results:

- *Direct Impact of the Workload Measure (within Figures)*: Changing the meaning of cards and controlling the shop load rather than the number of jobs leads to significant performance improvements for all performance measures considered in our study. Meanwhile, the use of either correction (dividing by the routing length or routing position) does not lead to any performance improvement compared to simply using the shop load; both measures appear to rely on the use of a limit for each station (as in Workload Control). The shop load can therefore be considered to be the best workload measure to be used within our load-based ConWIP system.



(a) FCFS

(b) STWK

Figure 3 – Performance Curves for FCFS and STWK

- *Interaction between the Workload Measure and Backlog Sequencing Rule (across Figures)*: The impact of the backlog sequencing rule when the number of jobs is controlled confirms results in Thürer *et al*. (2017). FCFS is outperformed by STWK in terms of the percentage tardy and both FCFS and STWK are outperformed by capacity slack-based sequencing rules, with CSjobdir leading to the best performance. However, there are significant two-way interactions between the workload measures and backlog sequencing rules. Load balancing improves if the workload of the shop rather than the number of jobs in the system is controlled; and, as a result, total throughput times are reduced. This effect – obtained by changing the meaning of cards – diminishes performance differences between the different backlog sequencing rules. Still, the combination of limiting the shop load (rather than the number of jobs) and using a capacity slack-based backlog sequencing rule leads to the best performance in terms of all three performance measures. It is therefore this combination that should be applied in practice.

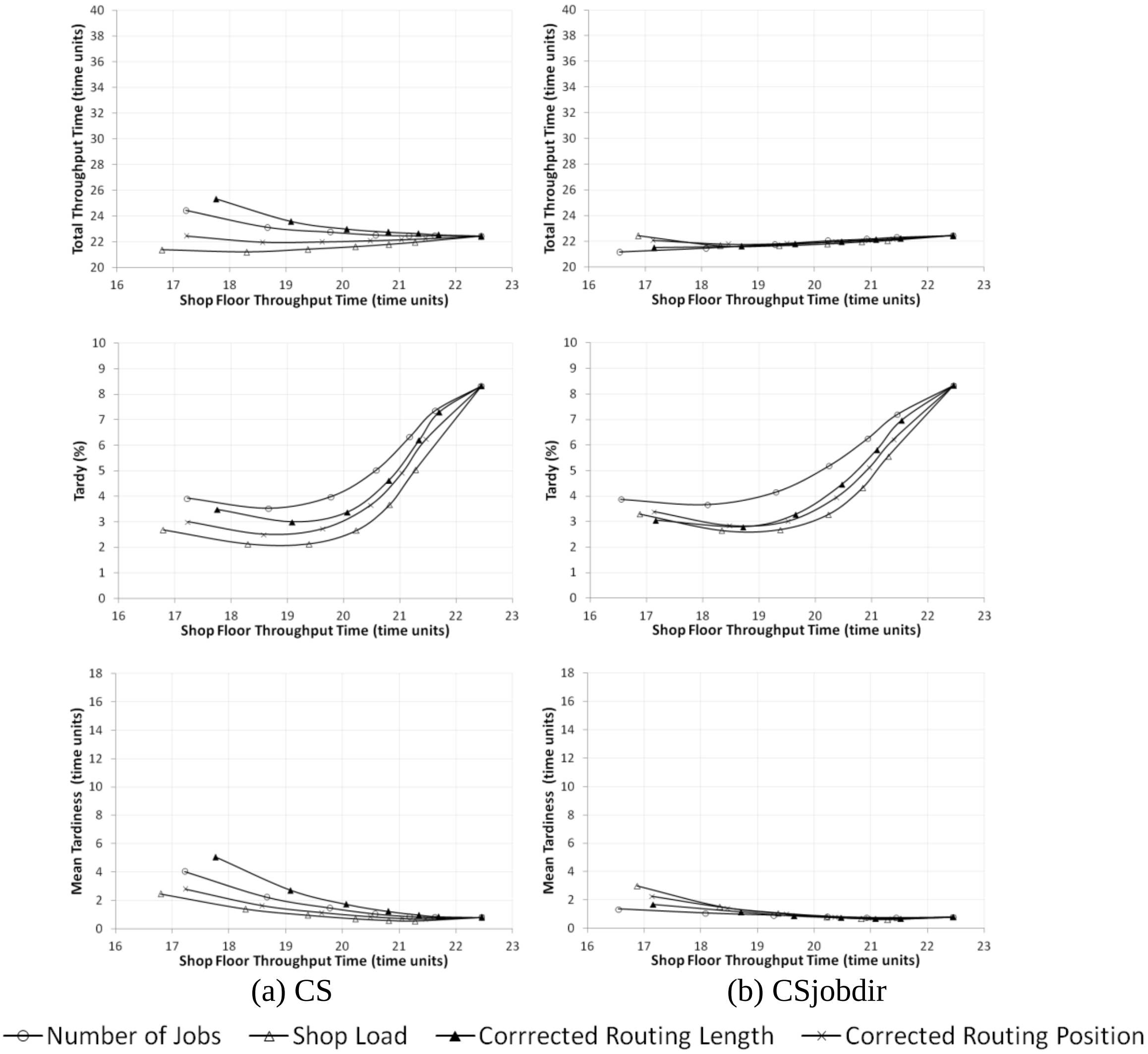


Figure 4 – Performance Curves for CS and CSjobdir

Conclusions

ConWIP is a simple yet effective means of implementing pull production – jobs are only allowed to enter the shop floor if the number of jobs on the shop floor is below a certain limit (the WIP- Cap). As a consequence, ConWIP has received much research attention, where a core focus has been on overcoming ConWIP's main shortcoming – a lack of load-balancing capability that hinders its use in high-variety contexts. While a major advantage of ConWIP is its simplicity, this simplicity also limits the opportunities available to improve the concept. There are arguably only two major search directions: (i) to alter the meaning of cards away from controlling jobs; and (ii) to adopt alternative backlog sequencing rules for considering jobs for release. In this study, we propose that a ConWIP card should be adapted such that it represents a measure of workload rather than a job, and we present a simple, practical solution for implementing this load-based ConWIP system in practice. Using controlled simulations, we asked: *Can ConWIP performance be improved by associating ConWIP cards with a workload?* Our results have demonstrated the positive performance impacts of limiting the shop load instead of the number of jobs in the system. But using a correction to the shop load, as suggested in the Workload Control literature, leads to worse performance when compared to the shop load.

The major limitation of our study is the narrow set of environmental and control variables considered. For example, we have only considered one level of processing time variability. Similarly, only one dispatching rule for controlling the progress of jobs on the shop floor has been considered. While these choices are arguably justified by results from prior studies and the need to keep the study focused, future research could extend our research by exploring the performance of ConWIP and its contingency factors in a broader context.

References

Duenyas, I. (1994), "A simple release policy for networks of queues with controllable inputs", *Operations Research*, Vol. 42, pp. 1162-1171.

Framinan, J. M., Ruiz-Usano, R., and Leisten, R. (2000), "Input control and dispatching rules in a dynamic CONWIP flow-shop", *International Journal of Production Research*, Vol. 38, No. 18, pp. 4589-4598.

Germs, R., and Riezebos, J. (2010), "Workload balancing capability of pull systems in MTO production", *International Journal of Production Research*, Vol. 48, No. 8, pp. 2345-2360.

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

Spearman, M.L., Woodruff, D.L., and Hopp, W.J. (1990), "CONWIP: a pull alternative to kanban", *International Journal of Production Research*, Vol. 28, No. 5, pp. 879-894.

Thürer, M., Stevenson, M., and Silva, C. (2011), "Three decades of workload control research: a systematic review of the literature", *International Journal of Production Research*, Vol. 49, No. 23, pp. 6905-6935.

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

Thürer, M., Fernandes, N.O., Stevenson, M., and Qu, T. (2017), "On the Backlog-sequencing Decision for Extending the Applicability of ConWIP to High-Variety Contexts: An Assessment by Simulation", *International Journal of Production Research*, Vol. 55, No. 16, pp. 4695-4711