

COBACABANA (Control of Balance by Card Based Navigation): An Alternative to *Kanban* in the Pure Flow Shop?

Matthias Thürer (corresponding author), Mark Stevenson and Charles W. Protzman

Name: Dr. Matthias Thürer
Institution: Jinan University
Address: Jinan University
No 601, Huangpu Road
510632 Guangzhou, PR China
E-mail: matthiasthurer@workloadcontrol.com

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

Name: Charles W. Protzman
Institution: BIG – Business Improvement Group LLC.
Address: 1403 Midmeadow Road
Towson, MD 21286, USA
Email: charlieprotzman@comcast.net

COBACABANA (Control of Balance by Card Based Navigation): An Alternative to *Kanban* in the Pure Flow Shop?

Abstract

Kanban systems are widely applied in practice as they represent a simple yet effective means of controlling production. But they suffer from a lack of load balancing capabilities, which hinders their application even to pure flow shops if there is variability. In response, this study focuses on COBACABANA (Control of Balance by Card Based Navigation), a card-based production control approach based on the Workload Control concept that was recently introduced in the literature. COBACABANA was developed for high-variety job shop contexts, but we argue it can also provide an important control alternative to *kanban* systems in pure flow shops. We first show that, in the pure flow shop, the control loop structure of COBACABANA resembles that of a *kanban* system when the flow of jobs is controlled. But a distinct difference is COBACABANA's unique focus on load balancing. Using simulation, we then demonstrate the potential of COBACABANA to improve performance in a pure flow shop with high demand and processing time variability. Results show that a fixed gateway station – inherent to a pure flow shop – presents a structural constraint that makes COBACABANA's original starvation avoidance mechanism, which injects work to a starving station, dysfunctional. An alternative is prioritizing jobs with short processing times at upstream stations to ensure quick replenishment takes place at downstream stations threatened by starvation. This has important implications not only for COBACABANA but for priority dispatching. Although card-based systems are typically combined with first-come-first-served dispatching, our results suggest this may be inappropriate in flow shops with processing time variability.

Keywords: *Kanban; COBACABANA; Workload Control; Order Release; Dispatching.*

1. Introduction

In recent decades, repetitive manufacturers have simplified production control through the use of card-based control systems, particularly *kanban* systems that are an important element of just-in-time and lean production (e.g. Sugimuri *et al.*, 1977; Ohno, 1988; Shingo, 1989; Monden, 2011). *Kanban* systems are used to connect production stages or operations to one another to improve coordination, thereby regulating work-in-process and eliminating overproduction. They are commonly used in practice by a wide range of organizations that differ in size and production strategy. For example, White *et al.* (1999) – who surveyed companies that had implemented at least 1 of 10 so-called just-in-time practices – reported that about 50% of small & medium-sized manufacturers and 70% of large manufacturers had adopted *kanban* systems. Similarly, White & Prybutok (2001) reported that about 60% of non-repetitive and 70% of repetitive manufacturers that they surveyed had adopted *kanban* systems.

Kanban systems were originally applied to repetitive manufacturing, but their spread to non-repetitive shops, including pure flow shops with demand and processing time variability, has motivated much recent research. This recent body of work has sought to refine *kanban* systems so they can cope with variability (Lage Junior & Godinho Filho, 2010). But while some studies have attempted to address the problems that variability introduces by adjusting the number of *kanbans* (e.g. Takahashi & Nakamura, 1999; Dallery & Liberopolous, 2000) or by establishing different loop structures (e.g. Gaury *et al.*, 2001), Germs & Riezebos (2010) suggested that the poor performance of card-based systems in high-variety contexts may be explained by a lack of load balancing capabilities. Thus, adding load balancing capabilities may be a potential key to improving the performance of *kanban*-like systems in shops that feature variability.

In response, this study presents an alternative to a *kanban* system that has recently emerged in the job shop literature and incorporates load balancing capabilities: Control of Balance by CARD Based NAVigation or COBACABANA (see Land, 2009; Thüerer *et al.* 2014a). COBACABANA is based on the (non-card based) Workload Control concept, which has been shown to significantly improve the performance of job shops both through simulation (e.g. Thüerer *et al.*, 2012, 2014b and 2015) and, on occasions, in practice (e.g. Hendry *et al.*, 2013). Workload Control and, consequently, COBACABANA were designed to achieve the same leveling of workload to capacity that is achieved in repetitive manufacturing using lean tools, but while also allowing the company to offer highly customized products to its customers. It reduces the variability of the incoming workload that

results from product customization rather than limiting variation in the product mix itself (Thürer *et al.*, 2014b).

Although COBACABANA was developed for job shops, it is argued here that it can also provide an important control alternative to *kanban* systems in pure flow shops with high variability in terms of the occurrence of demand and/or processing times. This is a common shop type, e.g. for companies that focus on producing prototypes and making small runs, e.g. of 1 to 4 units, sometimes referred to as “one offs”. More specifically, the two main objectives of this study are:

1. To compare the control structure of *kanban* systems and COBACABANA in a pure flow shop, i.e. where all jobs have to visit all stations in the same sequence; and,
2. To use simulation to demonstrate the potential of COBACABANA to improve performance in contexts where *kanban* systems typically fail: pure flow shops with high variability in terms of the occurrence of demand and processing times.

The remainder of this paper is organized as follows. *Kanban* systems and COBACABANA are first introduced and compared in Section 2 where we show that, in the pure flow shop, the control loop structure of the two systems resemble one another when the flow of jobs is controlled. The difference is COBACABANA’s unique focus on load balancing. A simulation model of a pure flow shop, which we use to evaluate two variants of COBACABANA, is then outlined in Section 3. The simulation results are presented and discussed in Section 4 before the paper concludes in Section 5, which includes managerial implications and future research directions.

2. Kanban vs. COBACABANA

This section provides an overview and comparison of the *kanban* and COBACABANA systems. *Kanban* systems are first discussed in Section 2.1 before the new COBACABANA system is introduced in Section 2.2. A reflection on the *kanban* literature is then provided in Section 2.3.

2.1. Kanban Systems

The Japanese word “*kanban*” translates literally to “watch over board for a period of one” – it is a common term that means signboard or billboard (Protzman *et al.*, 2010). A *kanban* system is an order release system that withholds work from the shop. Work is not released immediately to the shop floor – it has to wait for authorization (through *kanban* cards/signals

that trigger release). Since the number of *kanban* cards in the system is restricted, it is essentially an input/output control system, where the output rate determines the input rate.

While the original *kanban* system is mainly described as an ‘inventory control system’, we focus on its use as an ‘order control system’. Most production processes involve both systems, with the two separated at the inventory/order interface (Hopp & Spearman, 2004) or order penetration point (Olhager, 2003). The inventory/order interface emerges at the point where a job is linked to a customer; linked, according to our definition, means receiving a ‘mark’ or “genidentity” in the sense of Reichenbach (Reichenbach, 1999). This essentially means that jobs are interchangeable in the inventory control system while, in the order control system, they are not. The *kanban* system – as an inventory control system – was developed to curb overproduction (Ohno, 1988; Shingo, 1989). *Kanbans* were used to signal from the downstream to the upstream station that parts are needed, which ensured that parts would only be produced if they would actually be used at a later stage of production. Since jobs are interchangeable, *kanban* cards are independent from individual jobs. Rather, they are dedicated to job classes. This decouples the control loop operating between two stages from all preceding stages, with production control exercised through a chain of interlinked pairs of stages (see Figure 1).

[Take in Figure 1]

In this study, we focus on a pure flow shop, where the flow of individual jobs is controlled – i.e. an order control system. In an order control system, the nature of the control problem is different from that in an inventory control system. Since jobs are no longer interchangeable, *kanban* cards have to be dedicated to individual orders. But, if it is dedicated to an individual order, the *kanban* card (and thus production) is blocked until the order belonging to that *kanban* arrives at the station. In other words, *kanban* cards now not only represent the direct load queuing at a station but also the indirect load that is still upstream. The further downstream a station is, the higher its indirect load. For example, the *kanban* card of the 3rd station stays about three times longer at the station than a *kanban* card for the first station (see also Land, 2009). This may explain why, in studies that sought to identify the optimum number of *kanban* cards in circulation for each station (e.g. Gstettner & Kuhn, 1996), downstream stations had a much higher number of cards. This phenomenon significantly changes the *kanban* system. Production is no longer controlled by the last station, which should have the highest number of *kanban* cards in circulation, but by the first station, which should have the lowest number of *kanban* cards in circulation.

The above change to the control problem was recognized by the introduction of so-called job-order *kanbans*, which are issued for each job and different from other types of *kanban* cards used for recurrent production (see, e.g. Monden, 2011). Similarly, the card-based Constant Work-in-Process or ConWIP system (e.g. Spearman *et al.*, 1990) establishes only one feedback loop from the final to the first station (or gateway station) and triggers release whenever a job is completed. In general, however, ConWIP suffers from the same weakness as *kanban* systems – a lack of load balancing capabilities (Germs & Riezebos, 2010). Even the Paired cell Overlapping Loops of Cards with Authorization (POLCA; e.g. Suri, 1998; Riezebos, 2010) system, which was designed to cope with more variability than *kanban* and ConWIP, only accounts for routing variability. POLCA represents an extension of a *kanban* system that allows a station to enter into a control loop with more than one station but, in a pure flow shop, POLCA and *kanban* systems are the same.

This lack of load balancing hinders the application of all well-known card-based systems – *kanban*, ConWIP and POLCA – to pure flow shops with variability in terms of the occurrence of demand and processing times. It is this type of flow shop for which COBACABANA – with its focus on load balancing – is proposed as an alternative.

2.2 Introduction to COBACABANA Systems

This section discusses three key aspects of a COBACABANA system: the overall structure (Section 2.2.1), when release decisions should be taken (Section 2.2.2), and how the release decision should be taken (Section 2.2.3).

2.2.1 COBACABANA's Overall Structure

If *kanban* cards are dedicated to specific jobs, they typically fulfill a different function. They no longer function to curb overproduction – instead, they act as a place holder to reserve capacity. So, essentially, the *kanban* system releases orders by backward checking whether there is capacity available to start production without violating a target level of Work-In-Process – the so-called WIP-Cap (Hopp & Spearman, 2004). The *kanban* system transforms into the general *kanban* system proposed by Chang & Yih (1994a, 1994b). Here, an order is only released to the shop if it acquires a *kanban* card for each station. Once the order is completed at a station, the *kanban* card is dropped and can be acquired by a different order.

COBACABANA – as illustrated in Figure 2 – has a structure that is similar to the general *kanban* system. But, rather than obtaining the authorization individually at each station, control is centralized. Orders arriving to the shop are not released immediately but have to wait in a so-called pre-shop pool. This pool is typically positioned at the inventory/order

interface that separates the production process into two and decouples the inventory control and order control systems. Orders are then released from the pool to the shop floor to meet certain performance targets. Direct card loops between the planner performing the order release decision and each station on the shop floor are established, which provides a global rather than local view of the shop and allows for load balancing as part of this release decision. COBACABANA, as presented here, follows the refinements proposed by Thüerer *et al.* (2014a) to Land's (2009) original card-based concept. This means two types of cards are used: (i) operation cards, which travel with the order and signal when an operation is complete; and, (ii) release cards, which visualize the shop floor's workload situation on the planner's display (as will be described in Section 2.2.3 below).

[Take in Figure 2]

2.2.2 COBACABANA's Timing of Release: When the Decision Should be Taken

Kanban systems are based on either constant quantity or constant time. In a constant quantity system, releases are triggered *continuously*, i.e. can occur at any moment in time whenever certain pre-established upper or lower quantities (represented by *kanban* cards) are violated. In a constant time system, releases occur *periodically* at fixed time intervals. COBACABANA, according to Thüerer *et al.* (2014a), combines both a continuous and a periodic release time element. It executes the main release decision (see Section 2.2.3 below) at periodic time intervals to allow for workload balancing, but it also uses a continuous trigger to release work to a station whenever it is starving. In this study, we will use two different variants of COBACABANA to account for this difference in the timing of release:

- *Original COBACABANA* (Land, 2009; Thüerer *et al.*, 2014a): a system that triggers the main release decision at fixed (periodic) time intervals and also releases an order without load considerations whenever the first gateway station is starving; and
- *Continuous COBACABANA*: a system that is similar to a general *kanban* system, which triggers the main release decision whenever an operation is completed or a new order arrives in the pre-shop pool.

2.2.3 COBACABANA's Main Release Decision: How the Decision Should be Taken

When the main release decision takes place, orders in the pool are sorted according to their due date. The subset of orders to be released from the pool is then determined by considering all orders in the pool for release once, beginning with the first order in the sequence. Each operation in a job's routing has one *release card* and one *operation card*. The size of the

release card represents the corrected workload of the operation. The corrected workload divides the processing time of the operation at a station by the station's position in the job's routing (Oosterman *et al.*, 2000). This recognizes that the card for the second operation stays on the shop floor about twice as long as the card for the first operation. So the use of cards in COBACABANA is different from the use of cards in any other card-based system, including *kanban*, since cards represent the workload of each operation by their adjustable size. This allows load balancing to be integrated into the release decision.

To consider an order for release, the planner places the release card(s) at each station's area on the planning board (see Figure 3). The planner then compares the workload of each station with the predetermined workload limits or norms used to control order release. If, for any station in the routing of an order, the workload represented by the release cards on the planning board (the existing workload plus the new order's workload) exceeds its workload limit, the order is retained in the pool and the order's release cards are removed from the planning board. Otherwise, the order's release cards remain on the planning board, the planner attaches the corresponding operation cards to an order guidance form that travels with an order through the shop, and the order is released. This release process continues until all orders in the pool have been considered for release. The shop floor returns each operation card to the planner as soon as an operation is completed. This closes the information loop and signals to the planner to remove the release card that matches the operation card from the planning board. This process can be simplified by color-coding the cards such that each station is represented by a different color, similar to in the POLCA system (e.g. Riezebos, 2010).

[Take in Figure 3]

Figure 3 illustrates an example of how the planning board is used when making a release decision. Since the job's workload contribution to the second station violates the workload norm in this example, the order is not released and remains in the pool. The planning board bears a similarity to a *heijunka* box (e.g. Marchwinski *et al.*, 2008) and gives a quick overview of the shop floor status. The release cards clearly show the workload released to each station whereas the empty space left on the planning board shows the remaining release possibilities. But while the *heijunka* box typically levels the mix and volume of production by assigning capacity to product types, COBACABANA contributes the workload of individual jobs towards the total workload allowed for release to each station.

Finally, cards are physically stored in an order guidance form, which accompanies an order through the whole process. This guidance form can be used to summarize basic job information and, in the absence of electronic information feedback, to collect order progress information for later diagnosis (see, e.g. Soepenber *et al.*, 2008). For example, operators can write realized operation completion dates or quality problems on the form for subsequent analysis.

2.3 Reflection on the Literature

While COBACABANA's control structure resembles a *kanban* system when the flow of jobs is controlled, it differs in two important aspects that allow it to extend beyond the benefits and scope (or applicability) of traditional *kanban* systems:

- First, the centralization of control, which allows for a global rather than local view of the load situation on the shop floor, supports load balancing through the creation of a mix of jobs on the shop floor that balances the workload across resources; and
- Second, the use of adjustable card sizes to represent the workload contribution to a station, which makes workloads visible and, in combination with the centralized planning board, simplifies load balancing calculations.

To the best of our knowledge, only one study in the *kanban* literature has focused on load balancing. Driven by the need to accommodate variation in processing times, Gupta & Al-Turki (1997) developed a rather complex centralized algorithm. They fit the workload contribution of parts (measured in processing time) to the capacity available during a day by adjusting the number of *kanbans*. For example, if parts have a processing time of 0.2 days, then the number of *kanbans* should be equal to 5; if parts have a processing time of 0.5 days, then the number of *kanbans* should be equal to 2; and so on. An advantage of COBACABANA over this approach is that it significantly simplifies load balancing calculations and, most importantly, makes workloads visible through the use of cards.

Finally, it should be noted that there are other studies that focus on dynamically adjusting the number of *kanbans* in the system. But these studies do not focus on load balancing and do not maintain the WIP-Cap, which is what defines a pull system according to Hopp & Spearman (2004). Rather, most other studies suggest an increase in the input level in response to high demand. For example, Dallery & Liberopolous (2000) state that ‘..., *in a situation of high demand variability, one would like to have a ‘large’ number of kanbans at times of high demand, to quickly respond to demand.*’ Similarly, Takahashi & Nakamura (1999) and

Takahashi (2003) use forecasting techniques to predict changes in demand – with the number of *kanbans* increased if the forecast inter-arrival time falls below a threshold level – and Tardif & Maaseidvaag (2001) add *kanbans* if demand increases. It is argued here that this not only runs counter to the original idea that drove the development of *kanban* systems – a low and stable inventory buffer on the shop floor – but should also have no effect on output (except that the system is under-utilized). Since it increases the input rate (e.g. in response to demand) without increasing the output rate it is in direct conflict with the basic idea of input/output control, i.e. that the input rate to a shop must be equal to the output rate (Wight, 1970). Any positive effect observed from these changes is likely to have been caused by an increase in the inventory buffer, which compensates for starvation caused by the poor balancing capabilities of the original *kanban* system.

COBACABANA is different since it uses load balancing to avoid starvation while simultaneously reducing and stabilizing work-in-process levels, thus reducing the inventory buffer and aligning input with output. Simulation will next be used to assess the performance of the two variants of COBACABANA identified in Section 2.2.2 in a pure flow shop with variability in terms of the occurrence of demand and processing times.

3. Simulation Model

The shop and job characteristics modeled in the simulations are first outlined in Section 3.1. The order release rules applied are then summarized in Section 3.2 before the dispatching rules for the shop floor are summarized 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 pure flow shop (Oosterman *et al.*, 2000) has been implemented in Python[®] using the SimPy[®] module. Each job visits all stations in the same sequence in order of increasing station number. The shop contains six stations, where each station is a single resource with constant capacity.

Operation processing times follow a truncated lognormal distribution (Trietsch *et al.*, 2012) with a mean of 1 time unit after truncation and a maximum of 4 time units. Three levels of processing time variability are modeled, with a coefficient of variation, $cv^2 = 0.25$, 0.5 and 1 (after truncation). The level of 1 is equal to the variability of an exponential distribution, which is typically considered to represent “high” processing time variability. The “medium” level of 0.5 is equal to the variability of a 2-Erlang distribution. Finally, the level of 0.25 has been chosen for “low” processing time variability. This level is still

sufficient to avoid unrealistic, nearly symmetric distributions, as observed for lower cv^2 levels.

Set-up times are considered to be part of the operation processing time. Meanwhile, the inter-arrival time of orders follows an exponential distribution with a mean of 1.111, which – based on the number of stations in the routing of an order – deliberately results in a utilization level of 90%. Finally, due dates are set exogenously by adding a random allowance factor, uniformly distributed between 40 and 60 time units, to the job entry time.

3.2 Order Release

As in previous simulation studies on *kanban* systems (Chang & Yih, 1994a; Gupta & Al-Turki, 1997; Gaury *et al.*, 2001) and on COBACABANA (Thürer *et al.*, 2014a), 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. Orders flow into a pre-shop pool to await release according to one of the two variants of COBACABANA: Original COBACABANA or Continuous COBACABANA (see Section 2.2.2). Seven workload norm levels are applied, ranging from 4 to 10 time units of corrected load and, for the periodic release time element of COBACABANA, the time interval between releases is set to 4 time units. As a baseline measure, experiments without controlled order release have also been executed, i.e. where orders are released onto the shop floor immediately upon arrival. As a further reference, the general *kanban* system (Chang & Yih, 1994a, 1994b) has also been included in the experimental design.

3.3 Shop Floor Dispatching Rule

COBACABANA only controls the release of jobs and not the flow of jobs on the shop floor, e.g. between work stations, which remains under the control of the dispatching rule. COBACABANA creates a mix of jobs on the shop floor that balances the workload across stations, but the decision concerning which job to choose from the queue in front of a station still needs to be taken. This allows for the use of different dispatching rules according to the idiosyncratic needs of a shop. The dispatching rule typically applied in card-based systems is first-come-first-served (also known as first-in-first-out). There appear to be two reasons that justify this choice of dispatching rule. First, there is typically no or little processing time variability in the environments where card-based systems were originally developed, which makes the application of load-based dispatching rules (e.g. the shortest processing time rule) not meaningful. Second, card-based systems were generally developed for to-stock

production environments, and first-come-first-served is a suitable time-based dispatching rule for shops where jobs do not have individual due dates.

Both reasons are questioned in the production environment considered here. First, there is high processing time variability, which allows for the use of a load-oriented dispatching rule. Second, jobs have individual due dates and thus a due date-oriented dispatching rule is more suitable than first-come-first-served dispatching to indicate the urgency of jobs. Consequently, the following three dispatching rules will be considered in this study for controlling the flow of jobs on the shop floor: (i) the Operation Due Date (ODD) rule, a time-based rule that considers the urgency of jobs; (ii) the Shortest Processing Time (SPT) rule, a load-based rule that has been previously shown to reduce throughput times in flow shops (e.g. Conway, 1967); and (iii) the Modified Operation Due Date (MODD) rule, which combines the SPT and ODD rules (e.g. Baker & Kanet, 1983, Baker, 1984).

The calculation of the operation due date d_{ij} for the i^{th} operation of a job j follows Equation (1). The operation due date for the last operation with index n_j in the routing of a job is equal to the due date d_j , while the operation due date of each preceding operation is determined by successively subtracting a constant allowance c from the operation due date of the next operation. The allowance has been set to 6 time units in this study. This value has been chosen based on preliminary simulation experiments that indicated it resulted in the best overall performance.

$$d_{ij} = d_j - (n_j - i) \cdot c \quad i: 1..n_j \quad (1)$$

The ODD rule prioritizes jobs with the earliest operation due date. Meanwhile, the SPT rule selects the job from the queue with the shortest processing time at a given station. 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, i.e. $\max(d_{ij}, t + p_{ij})$ for an operation with processing time p_{ij} , where t refers to 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 – when multiple jobs exceed their ODD.

3.4 Experimental Design and Performance Measures

The experimental factors are: (i) the two variants of COBACABANA (Original and Continuous); (ii) the seven levels of workload norm (4 to 10 time units); (iii) the three dispatching rules (ODD, SPT and MODD); and, (iv) the three levels of processing time variability (low, medium and high). A full factorial design with 126 cells was used, 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 performance measures considered in this study are as follows:

- *Mean throughput time*: the mean of the completion time minus the release time across jobs. This indicates the mean time that a job spends on the shop floor and, by Little's law (Little, 1961), reflects the workload level on the shop floor.
- *Mean lead time*: the mean of the completion time minus the arrival time across jobs. This measure – also given by the mean of the throughput time plus the time that an order has to wait in the pool before release – reflects the time that a customer has to wait for the product.
- *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 mean of the tardiness $T_j = \max(0, L_j)$, with L_j being the lateness of job j (i.e. its actual delivery time minus its due date). This indicates the extent to which jobs are tardy.

Finally, the significance of the differences between the outcomes of individual experiments have been verified by paired t-tests, which comply with the use of common random number streams to reduce variation across experiments. Whenever we discuss a difference in outcomes between two experiments, the significance can be proven by a paired t-test at a level of 97.5%.

4. Results

The presentation of results is structured as follows. First, the performance of our two variants of COBACABANA in combination with ODD dispatching is assessed in Section 4.1. The impact of SPT and MODD dispatching is then evaluated in Section 4.2 and Section 4.3, respectively. Finally, a discussion of results is presented in Section 4.4.

4.1 The Performance of COBACABANA in Combination with ODD Dispatching

Figure 4a, 4b and 4c show the lead time, percentage tardy and mean tardiness results over the throughput time results, respectively with ODD dispatching. Only results for medium processing time variability are shown since the results for our release methods were qualitatively equivalent across the three levels of this factor. Results are presented in the form of performance curves, where the left-hand starting point of the curves represents the tightest

workload norm level (4 time units). The workload norm increases step-wise by moving from left to right in each graph, with each data point representing one norm level (from 4 to 10 time units). Loosening the norms increases the workload levels and, as a result, the throughput times on the shop floor. In addition, the result obtained when orders are released immediately is also included. This is referred to as IMM (IMMEDIATE release) – see the single point “X” in the figures – and represents the outcome with no order release control. It is located to the right of the curves as it leads to the highest level of throughput times on the shop floor. Finally, the performance curve of the general *kanban* system is depicted by the dashed curve.

[Take in Figure 4]

First, it can be observed from Figure 4 that using the general *kanban* system is detrimental to performance and is consequently outperformed by both variants of COBACABANA. Second, the Original COBACABANA system is outperformed by Continuous COBACABANA, as explained below.

The starvation trigger incorporated in the Original COBACABANA system releases jobs regardless of whether they violate one or more norms. The trigger was introduced by Thüerer *et al.* (2012) to avoid so-called premature idleness (see Kanet, 1988; Land & Gaalman, 1998), which may result from the strict enforcement of norms in combination with periodic release. Without this trigger, since a job has to fit the norm at all stations in its routing, it is possible that a station may be starving even though a job that can be processed directly at this station is waiting in the pool. This job may be being held up in the pool because its release would violate the norm of another station – hence, the first station becomes idle prematurely. The trigger itself however becomes dysfunctional in a pure flow shop or if all jobs enter the shop via a particular gateway station. If the percentage of jobs that are released by the continuous element is high, control becomes very localized on the gateway station. Thus, although the workload of the gateway station may be very tightly controlled, the workload of other stations is not. This limits the impact on performance of the Original COBACABANA system and leads to its poor performance.

An alternative solution for postponing starvation at a station is to give priority to jobs with the shortest processing times at upstream stations since this will create the quickest replenishment of successive queues. Using an upper bound as part of a continuous release method, as in Continuous COBACABANA – which constantly attempts to fill up the workload norms as much as possible – leads to SPT effects if many jobs are waiting to be

released (Land *et al.*, 2010). This is because, for a job to fit the norm, its workload contribution has to be equal to or smaller than the workload contribution freed up by the job completed. It is argued that the combination of load balancing – which postpones the starvation of stations for as long as possible by reducing the variability of queues in front of a station – and the creation of SPT effects leads to the superior performance of Continuous COBACABANA over the Original COBACABANA system.

4.2 The Performance Impact of SPT Dispatching

The positive impact of SPT effects noted above for Continuous COBACABANA can also be observed from Figure 5a, 5b and 5c, which show the lead time, percentage tardy and mean tardiness results over the throughput time results, respectively with SPT dispatching. Again, only results for medium processing time variability are shown since the results for our release methods were qualitatively equivalent across the three levels of variability evaluated in this study.

[Take in Figure 5]

As expected from the scheduling literature, SPT dispatching significantly reduces the throughput time and lead time compared to ODD dispatching. This is because processing the smallest orders first typically involves the least sacrifice to achieve a reduction in the average throughput time: many small orders can usually be processed quickly at the expense of delaying just a few large orders. The big reduction in lead times and, consequently, in average lateness also results in a lower percentage tardy. Moreover, the strong effect of SPT dispatching diminishes the impact of our release methods – and the resulting performance difference – on percentage tardy and mean tardiness performance.

Overall, Continuous COBACABANA remains the best-performing release method when compared to the Original COBACABANA system and general *kanban*. Further, it allows for significant performance improvements in terms of throughput times and mean tardiness compared to immediate release (i.e. when control is only exercised via the SPT dispatching rule).

4.3 The Performance Impact of MODD Dispatching

Figure 6a, 6b and 6c present the lead time, percentage tardy and mean tardiness results over the throughput time results, respectively with MODD dispatching. MODD combines time-oriented and load-oriented dispatching into one rule. It uses its SPT element to speed up the progress of jobs and postpone starvation when multiple jobs become urgent. Meanwhile, it

uses its ODD element to ensure non-urgent jobs are released, so the mix of released jobs can be produced in time, thereby reducing tardiness.

[Take in Figure 6]

Compared to SPT dispatching, it can be observed that MODD leads to a lower mean tardiness while maintaining a similar percentage tardy performance. Since SPT effects are restricted to certain periods, the performance of MODD is realized with longer lead times and throughput times when compared to SPT. As for ODD and SPT dispatching, Continuous COBACABANA is the best-performing release method when compared to Original COBACABANA and general *kanban*. It leads to performance improvements in terms of both throughput time and tardiness performance when compared to immediate release.

Finally, the results in Figure 6 are for medium processing time variability. While the performance differences between the card-based systems are again qualitatively equivalent across the three levels of processing time variability, the tardiness performance of MODD and SPT is affected by this factor. Both dispatching rules always outperform ODD dispatching. But SPT gains an advantage over MODD when processing time variability is high; and it is outperformed by MODD on both percentage tardy and mean tardiness performance if processing time variability is low. However, since this effect was anticipated from the previous literature (e.g. Thüerer *et al.*'s (2015) study on the pre-shop pool sequencing decision), the results are not presented here.

4.4 Discussion of Results

Our results have shown that the Original COBACABANA system, as introduced by Land (2009) and Thüerer *et al.* (2014a), has shortcomings in the pure flow job shop. Since the percentage of jobs triggered by the starvation trigger is similar to that previously recorded in the pure job shop and general flow shop (see, e.g. Thüerer *et al.*, 2012) – where its performance is significantly better – the differences in its performance must be explained by the structure of the shop. In a pure flow shop, all continuous job releases are triggered by the first station. This makes the trigger, which was designed to avoid premature idleness, dysfunctional and restricts all control to the gateway station if the percentage of jobs released by the continuous release time element is high.

Another structural constraint is that all work arriving at a given station comes from one particular station. Hence, the work arriving is dependent on the processing times of orders at the preceding upstream station. Imagine an order that, according to our model, has the

maximum processing time of 4 time units at a station. If this order is now processed, no work will arrive at the next station for 4 time units. An inventory buffer of 4 time units is therefore required to avoid the second station from running out of work. But if there is another order with a processing time of 4 time units, we already require an inventory buffer of 8 time units minus the processing time of the first order at the second station, and so on. Creating SPT effects at the upstream station will postpone the processing of these orders with large processing times and avoid starvation at downstream stations. It is thus the combination of load balancing and SPT effects that makes Continuous COBACABANA an important alternative to *kanban* systems in pure flow shops with high variability in terms of the occurrence of demand and processing times.

Finally, our results also highlight the important role that a load-based dispatching rule such as SPT can play if the input of work to a station depends on processing times at upstream stations. Previous literature and practice has typically assumed a first-come-first-served rule. But such a rule is only appropriate if jobs do not have individual due dates and there is no processing time variability. Time-oriented rules such as ODD (and first-come-first-served) performed the worst in our experiments. Meanwhile, and as expected, load-oriented rules – in the form of SPT – performed best in terms of the average throughput time and lead time. However, since load-oriented rules neglect the urgency of jobs, they may lead to high tardiness, specifically for jobs with large processing times. MODD presents a means of combining the advantage of time-oriented rules, that ensure the timely completion of jobs, with a focus on speeding up jobs – through SPT effects – when multiple jobs exceed their ODD. This leads to the lowest tardiness performance across jobs.

5. Conclusions

While *kanban* systems are widely applied in practice (e.g. White *et al.*, 1999; White & Prybutok, 2001), it has been noted that they suffer from a lack of load balancing capabilities (Germs & Riezebos, 2010). This hinders their application even to pure flow shops if there is variability, e.g. in terms of the occurrence of demand and processing times. In contrast, load balancing is an essential part of COBACABANA, an alternative card-based production control system that was specifically developed for high-variety contexts such as pure job shops (e.g. Land, 2009; Thürer *et al.*, 2014a). The simulation results presented in this paper for a high-variety pure flow shop demonstrate COBACABANA's potential to significantly improve performance compared to the general *kanban* system and immediate release across all four performance measures considered: throughput time, lead time, percentage tardy and

mean tardiness. This makes COBACABANA an important alternative to *kanban* systems in the pure flow shop when there is high variability. It allows managers of companies competing in non-repetitive production environments to obtain the same benefits as achieved through traditional *kanban* systems in repetitive environments.

Finally, performance improvements observed for COBACABANA were independent of the dispatching rule applied although significant performance differences between dispatching rules were evident. Yet there is no dispatching rule that performs the best on all of our metrics. Therefore, the dispatching rule chosen to combine with COBACABANA will depend on what is most important to a particular shop. Since COBACABANA only controls the release of jobs to the shop floor, it can be combined with any dispatching rule.

5.1 Managerial Implications

Our analysis has revealed a structural constraint that is inherent to the modeled pure flow shop environment, i.e. where all jobs have to visit all stations in the same sequence. We have observed that high processing times at upstream stations may lead to starvation at downstream stations. This structural constraint is independent of the production control system applied – but, uniquely, COBACABANA provides a means of mitigating the negative effect of this constraint. There are three main practical solutions:

- *Improve load balancing to reduce the variability of queues in front of a station.* Indeed, load balancing is an integral part of COBACABANA, which controls the mix of jobs on the shop floor such that the workload is balanced across stations.
- *Prioritize the release of jobs with short processing times so loads at downstream stations threatened by starvation can be replenished quickly.* Continuous COBACABANA – which constantly attempts to fill up the workload norms as much as possible – leads to SPT effects specifically during periods when many jobs arrive at the shop. The process of balancing the positive and negative impact (such as an increased risk of tardiness for orders with large processing times) of SPT effects can be facilitated by the planner's display in COBACABANA. This display provides an overview of the load situation at each resource in the shop. Moreover, COBACABANA only controls the release of jobs, which allows for the use of load-based dispatching rules on the shop floor.
- *Allow downstream stations to pull in work if they are starving.* This solution first requires a switch in configuration away from a strict pure flow shop; hence, a change in the routing of jobs is needed. This can be achieved, e.g. through product (re)design or by enhancing the mix of jobs produced. A switch in configuration would be supported by

COBACABANA since all routing permutations for an order can be accommodated by its existing, flexible card loop structure. This structure makes COBACABANA – to the best of our knowledge – the most suitable card-based production control solution for high-variety contexts, where routings vary (see, e.g. Thüerer *et al.*, 2012, 2014a and 2015).

5.1.1 Implementation Issues for Managers

Our results suggest that Continuous COBACABANA is the best variant of COBACABANA for a pure flow shop. However, to achieve the best overall balance in performance, it is argued that managers should not just release jobs ‘blindly’ according to this approach – judgment is also required to allow for the timely release of large jobs.

Based on previous applications of Workload Control (e.g. Stevenson, 2006; Hendry *et al.*, 2013), it is recommended that COBACABANA is initially implemented with no or ‘infinite’ workload norms. This means that, at first, the workload norm will not have an impact on performance; however, the workload will be visible and fluctuations will still be evident from the planner’s display. Once feedback loops and workload calculations have been established, the workload norms can be gradually tightened. Workload norms, however, should not be tightened too rapidly. Instead, this should be done gradually, with norms lowered stepwise until the desired balance between throughput time reduction (i.e. work-in-process) and tardiness performance has been reached. Once work-in-process has been reduced, other types of improvement opportunities may present themselves. For example, quality problems hidden by high work-in-process may become evident. Then, once such problems have been resolved, it may be possible to tighten the workload norms even further.

5.2 Limitations and Future Research

The main limitation of our study is that it has focused on a simple sequential production line. Future research should investigate how COBACABANA can be used in divergent/convergent flow shops. Further, the above suggestion of switching the configuration away from a strict pure flow shop towards a general flow shop (or even a pure job shop) also needs further investigation in terms of how this change can be achieved and how it would affect performance. Finally, another important avenue for future research is to explore how COBACABANA can enhance production control in shops with multiple production cells, which are often created as part of a lean implementation, e.g. with each cell responsible for a particular product family. This may involve, for example, introducing a nested COBACABANA system whereby individual COBACABANA systems are used to control the workload within each cell and another is used to control the workload across cells.

References

- Baker, K.R., 1984. Sequencing rules and due-date assignments in a job shop. *Management Science*, 30 (9), 1093-1104.
- Baker, K.R., and Kanet, J.J., 1983. Job shop scheduling with modified operation due-dates. *Journal of Operations Management*, 4 (1), 11-22.
- Chang, T.M., and Yih, Y., 1994a, Generic kanban systems for dynamic environments, *International Journal of Production Research*, 32, 4, 889–902.
- Chang, T.-M., and Yih, Y., 1994b, Determining the number of kanbans and lot sizes in a generic kanban system: a simulated annealing approach, *International Journal of Production Research*, 32, 8, 1991–2004.
- Conway, R.W., Maxwell, W.L., and Miller, L.W., 1967, *Theory of scheduling*, Addison-Wesley, Reading, Mass.
- Dallery, Y., and Liberopoulos, G., 2000, Extended kanban control system: combining kanban and base stock, *IIE Transactions*, 32, 4, 369–386.
- Gaury, E.G.A., Kleijnen, J.P.C., Pierreval, H., 2001, A methodology to customize pull control systems, *Journal of the Operational Research Society*, 52, 7, 789–799.
- Germes, R., and Riezebos, J., 2010, Workload balancing capability of pull systems in MTO production, *International Journal of Production Research*, 48, 8, 2345-2360.
- Gstettner, S., and Kuhn, H., 1996, Analysis of production control systems kanban and CONWIP, *International Journal of Production Research*, 34, 11, 3253–3273.
- Hopp, W. J., and Spearman, M. L., 2004, To pull or not to pull: What is the question?, *Manufacturing and Service Operations Management*, 6, 2, 133-148.
- Kanet, J.J., 1988, Load-limited order release in job shop scheduling systems, *Journal of Operations Management*, 7, 3, 44-58.
- Lage Junior, M. and Godinho Filho, M., 2010, Variations of the kanban system: Literature review and classification, *International Journal of Production Research*, 125, 13-21.
- 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., and Gaalman, G., 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., 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.
- Little, J., 1961, A proof of the theorem $L = \lambda W$, *Operations Research* 8, 383-387.

- Marchwinski C., Shook, J., and Schroeder, A., 2008. *Lean lexicon: A graphical glossary for lean thinkers*, 4th Edition, Cambridge, MA: The Lean Enterprise Institute.
- Monden, Y., 2011, *Toyota Production System: An Integrated Approach to Just-In-Time*, 4th Ed., Productivity Press.
- Hendry, L.C., Huang, Y., and Stevenson, M., 2013, Workload control: Successful implementation taking a contingency-based view of production planning & control, *International Journal of Operations & Production Management*, 33, 1, 69-103.
- Olhager, J., 2003, Strategic positioning of the order penetration point, *International Journal of Production Economics*, 85, 319-329.
- Ohno, T., 1988, *Toyota Production System: Beyond Large-Scale Production*, 1st Ed., Productivity Press.
- Oosterman, B., Land, M.L., and Gaalman, G., 2000, The influence of shop characteristics on workload control, *International Journal of Production Economics*, 68, 1, 107-119.
- Protzman, C., Mayzell, G., and Kerpchar, J., 2010, *Leveraging Lean in Healthcare: Transforming Your Enterprise into a High Quality Patient Care Delivery System*, 1st Ed. Productivity Press.
- Reichenbach, H., 1999, *The direction of time*, Dover Publications.
- Riezebos, J., 2010, Design of POLCA material control systems, *International Journal of Production Research*, 48, 5, 1455-1477.
- Shingo, S., 1989, *A Study of the Toyota Production System from an Industrial Engineering Viewpoint*, Cambridge, MA: Productivity Press.
- Soepenbergh, G. D., Land, M.J., and Gaalman, G., 2008, The order progress diagram: A supportive tool for diagnosing delivery reliability performance in make-to-order companies, *International Journal of Production Economics*, 112, 1, 495-503.
- 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., 2006, Refining a workload control (WLC) concept: A case study, *International Journal of Production Research*, 44, 4, 767 – 790.
- Sugimori, Y., Kusunoki, K., Cho., F., and Uchikawa, S., 1977, Toyota production system and Kanban system Materialization of just-in-time and respect-for-human system, *International Journal of Production Research*, 15, 6, 553-564.
- Suri, R., 1998, *Quick Response Manufacturing: A Companywide Approach to Reducing Lead Times*, Productivity Press, Portland, OR.
- Takahashi, K., 2003, Comparing reactive kanban systems, *International Journal of Production Research*, 41, 18, 4317–4337.
- Takahashi, K., and Nakamura, N., 1999, Reacting JIT ordering systems to the unstable changes in demand, *International Journal of Production Research*, 37, 10, 2293–2313.

- Tardif, V., and Maaseidvaag, L., 2001, An adaptive approach to controlling kanban systems, *European Journal of Operational Research*, 132, 2, 411–424.
- 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.
- Thürer, M., Land, M.J., and Stevenson, M., 2014a, 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., Fredendall, L.D., and Melnyk, S.A., 2014b, Lean Control for Make-to-Order Companies: Integrating Customer Enquiry Management and Order Release, *Production & Operations Management*, 23, 3, 463-476.
- 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*, (in print)
- Trietsch, D., Mazmanyán, L., Gevorgyan, L., Baker, K.R., 2012, Modeling activity times by the Parkinson distribution with a lognormal core: Theory and validation, *European Journal of Operational Research* 216, 2, 386-396.
- Wight, O., 1970, Input/Output control a real handle on lead time, *Production and Inventory Management Journal*, 11, 3, 9-31.
- White, R.E., and Prybutok, V., 2001, The relationship between JIT practices and type of production system, *Omega*, 29, 113-124.
- White, R.E., Pearson, J.N., and Wilson, J.R., 1999, JIT manufacturing: A survey of implementation in small and large US manufacturers, *Management Science*, 45, 1, 1-15.

List of Figures

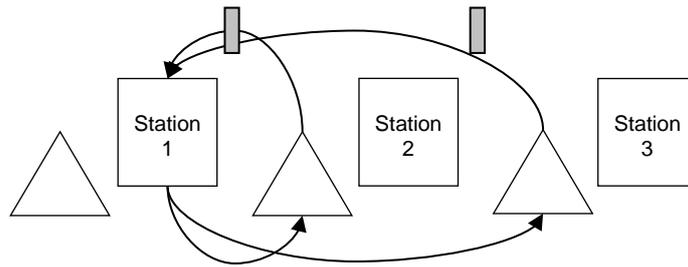


Figure 1: Kanban System – Loops between Queues and Preceding Stations that build a Chain of Interlinked Pairs

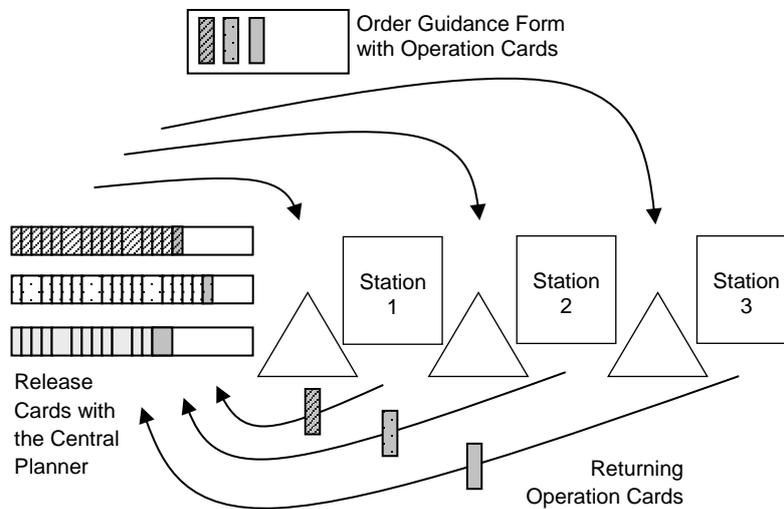


Figure 2: COBACABANA System – Card-based Order Release with Loops between the Central Planner and the Stations on the Shop Floor

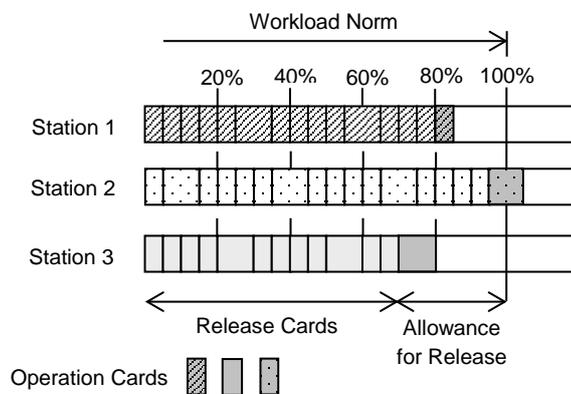
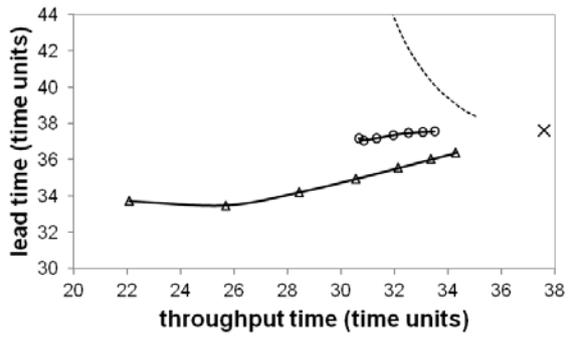
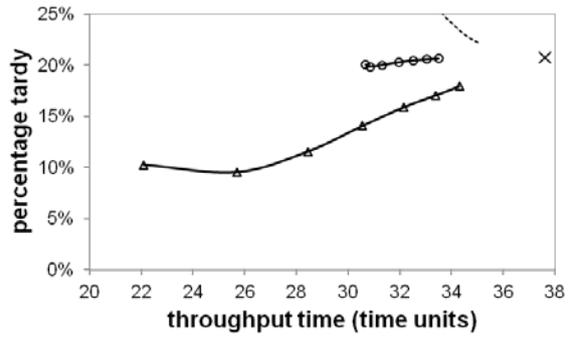


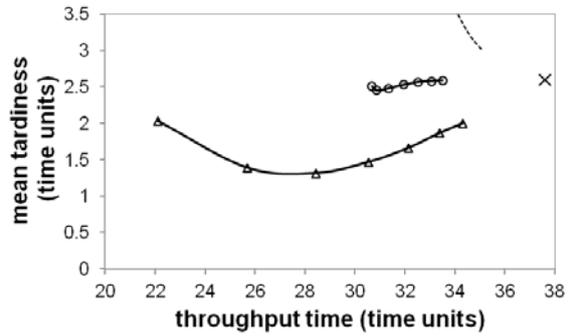
Figure 3: The Planner's Planning Board for Order Release (with an Example Release)



(a)



(b)



(c)

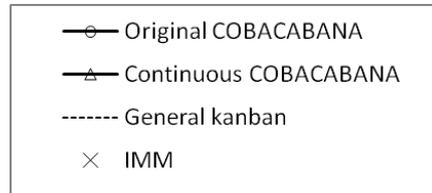


Figure 4: Performance Curves for ODD Dispatching under Medium Processing Time Variability: (a) Lead Time; (b) Percentage Tardy; and, (c) Mean Tardiness over the Throughput Time

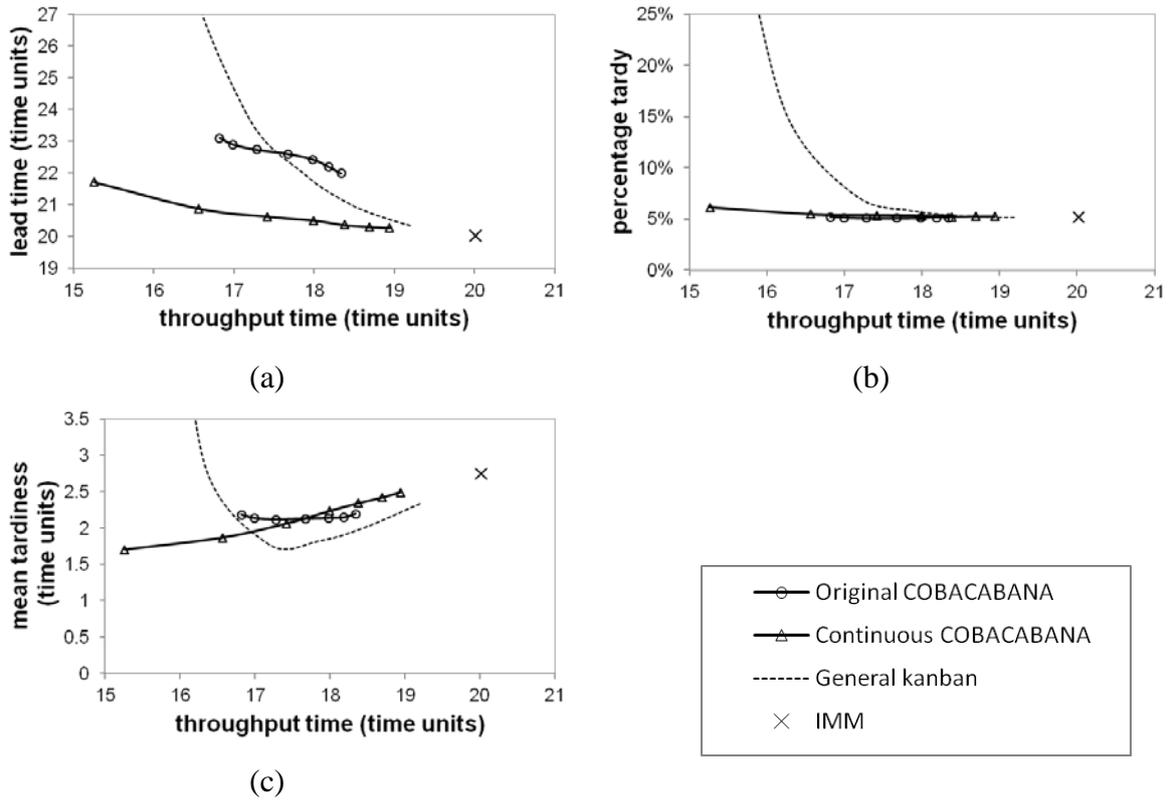


Figure 5: Performance Curves for SPT Dispatching under Medium Processing Time Variability: (a) Lead Time; (b) Percentage Tardy; and, (c) Mean Tardiness over the Throughput Time

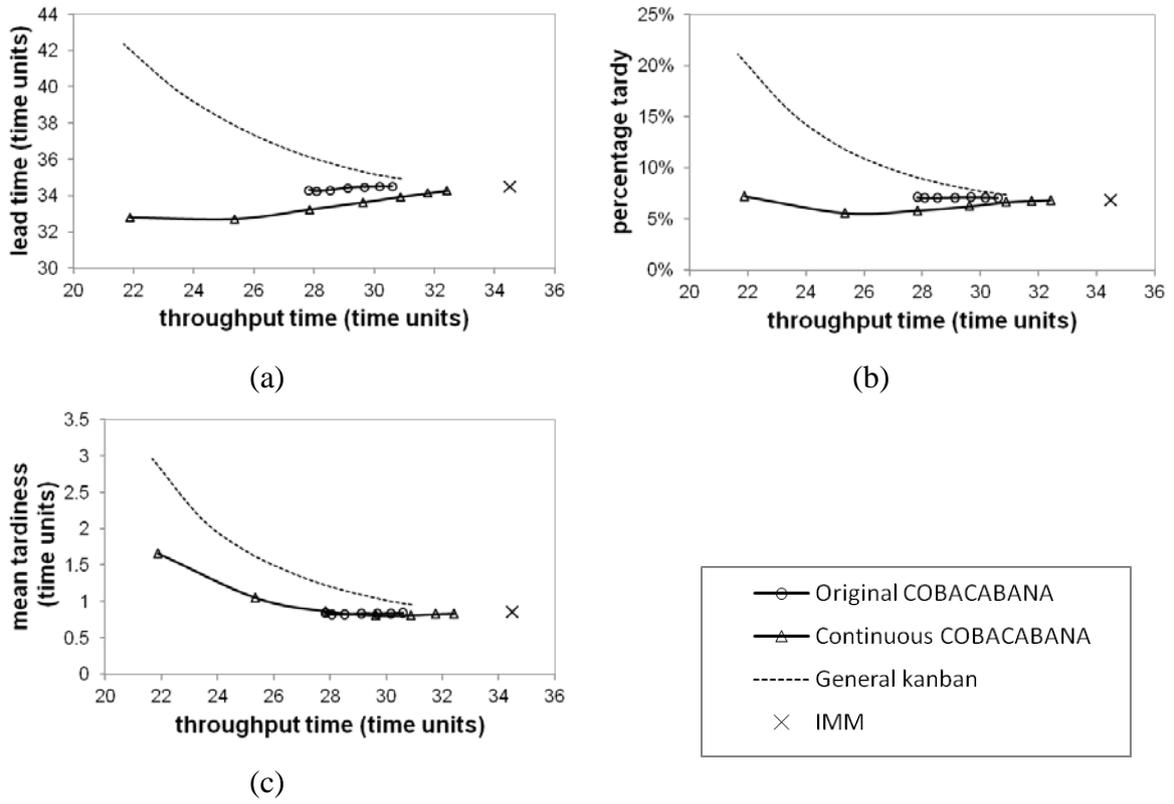


Figure 6: Performance Curves for MODD Dispatching under Medium Processing Time Variability: (a) Lead Time; (b) Percentage Tardy; and, (c) Mean Tardiness over the Throughput Time