Centralized *vs*. Decentralized Control Decision in Card-Based Control Systems: Comparing *kanban* Systems and COBACABANA

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

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

Name: Prof. Nuno O. Fernandes
Institution: Instituto Politécnico de Castelo Branco & ALGORITMI Research Unit (University of Minho)
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

Name: Prof. Cong Dong Li
Institution: Jinan University
Address: Institute of Physical Internet
Jinan University (Zhuhai Campus)
519070, Zhuhai, PR China
E-mail: licd@jnu.edu.cn

Keywords: *Kanban; Workload Control; Order Release; Shop Floor Control; Simulation*

Centralized *vs*. Decentralized Control Decision in Card-Based Control Systems: Comparing *kanban* Systems and COBACABANA

Abstract

Kanban systems are simple yet effective means of controlling production. Production control is decentralized or exercised locally on the shop floor, i.e. a downstream station signals to an upstream station that an item is needed. If items are always the same and known, then demands can be satisfied instantaneously from stock; but if items differ and are unknown, demands must first be propagated backwards from station to station before being satisfied. The former is defined as an inventory control problem and the latter as an order control problem. Handling the order control problem via *kanban* involves a decentralized card acquisition process (during which information is propagated from station to station) that is separated from the actual production process. COBACABANA (Control of Balance by Card-based Navigation), an alternative card-based solution, shares *kanban*'s control structure but centralizes the card acquisition process. Evaluating the two systems therefore provides a unique opportunity to compare decentralized and centralized control. Using simulation, we demonstrate that it is specifically the centralized card acquisition process that allows COBACABANA to balance the workload across resources and thus to outperform *kanban* in an order control problem. This has major implications for research and practice.

Keywords: *Kanban; Workload Control; Order Release; Shop Floor Control; Simulation.*

1. Introduction

In this study, we use the similarities and differences between *kanban* systems (e.g. Sugimuri *et al*., 1977; Monden, 1983; Ohno, 1988; Shingo, 1989) and Control of Balance by Card Based Navigation (COBACABANA; see Land, 2009; Thürer *et al*., 2014) to determine the impact of decentralizing the release decision. While *kanban* systems are characterized by a decentralized card acquisition process, COBACABANA, a card-based production control system that originates from the (non-card based) Workload Control concept, centralizes the card acquisition and thus the release decision. While our study draws heavily on previous literature on *kanban* systems and COBACABANA, it is to the best of our knowledge the first to explicitly model *kanban*'s card acquisition process.

The material flow in a *kanban* system is typically described by moving through the production process in sequence, from the operations at the beginning of the process required to complete a job through to the end of the process. But it is important to note that there is also an information flow in the other direction – a *kanban* signals or transmits information from the end to the beginning of a process. A defining characteristic of a *kanban* system is that control is decentralized and only exercised between production stages. The only centralized decision is determining the number of cards contained in each loop; and by controlling the number of cards in each loop, work-in-process on the shop floor is regulated.

If the same item(s) are always produced, then these items can be stored in an inventory decoupling point, typically referred to as a 'supermarket' (Ohno, 1988), that allows demand to be satisfied instantaneously. Since only inventory needs to be controlled, this scenario can be referred to as an **inventory control problem**. In an inventory control problem, the *kanban* system functions as an inventory replenishment system and the *kanban* card signals 'an item X **was used**, please replenish it'. But when each item differs, the item cannot be stored in advance and simply withdrawn from the supermarket to satisfy demand. In this case, the need for the item must first be propagated upstream from station to station until it reaches the first station where the need can be met; only then can processing of the job commence. Since, in this scenario, the flow of the whole order needs to be controlled, it can be referred to as an **order control problem**. Note that the distinction between an inventory and an order control problem is different from the distinction between make-to-stock and make-to-order, which captures when demand is placed, i.e. before or after production. For example, both inventory and order control problems can occur in a make-to-order context where production only occurs after demand is placed.

In an order control problem, a job has to acquire a *kanban* card at each station by moving backwards from the last station until it reaches the station where the need can be met – the so-called inventory/order separation point (Hopp & Spearman, 2004) or order penetration point (Olhager, 2003). Only then can the physical processing actually start. In other words, information on which item to start needs to be propagated backwards to the inventory/order separation point. The need can be met at the inventory/order separation point since the job's customer-specificity reduces as we move upstream in the creation process. At a certain point, it becomes 'inventory'. During this card acquisition process, the *kanban* card signals 'an item X **will be used**, please create it'.

A similar acquisition process to that incorporated in Ohno's *kanban* system is executed by COBACABANA. However, in contrast to *kanban* and its decentralized control process, COBACABANA centralizes the acquisition process, establishing card loops between each station and a centralized release function. Thus, while in a *kanban* system the acquisition process is decentralized and each card has to be acquired at each station individually, in COBACABANA the centralized release function supports the acquisition process. Evaluating the two systems thus provides a unique opportunity to compare decentralized with centralized control.

It was recently argued by Thürer *et al*. (2015a) that the difference in the acquisition process (centralized *vs*. decentralized) should significantly enhance the performance of COBACABANA compared to a *kanban* system for the order control problem. But this assertion has not been tested; and, in fact, the actual performance impact of *kanban*'s decentralized acquisition process has never been determined. Thürer *et al*. (2015a) did not include *kanban* in their experimental design, only assessing the performance of different COBACABANA systems. Meanwhile, to the best of our knowledge, there are only two papers in the *kanban* literature that model *kanban* in the context of an order control problem where items are unknown and cannot be stocked beforehand: Chang & Yih (1994a and 1994b). Yet, while these authors recognized the need for a card acquisition process that precedes the actual production process, they did not explain how their acquisition process works or whether and how it differs from the acquisition process in Ohno's (1988) original *kanban* system. In response, this study assesses the performance differences between a decentralized acquisition process (Ohno's *kanban* system) and a centralized acquisition process (COBACABANA) through simulation. It is hoped that the insights gained provide guidance to managers and researchers for the design of control systems.

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 the control structures of the two systems resemble one another. The main difference between the two is the degree of centralization of the acquisition, and thus control, process. The simulation model used to assess performance is then outlined in Section 3. The 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 (Decentralized *vs*. Centralized Control)

This section provides an overview and comparison of the *kanban* and COBACABANA systems. *Kanban* systems are first discussed in Section 2.1 before the COBACABANA system is introduced in Section 2.2. A comparison between *kanban* and COBACABANA is then provided in Section 2.3. We do not aim to present a comprehensive review of all of the literature here – for this, the reader is referred to Berkley (1992), Lage Junior & Godinho Filho (2010), and Thürer *et al*. (2016). Rather, literature is only discussed if it is of relevance to the argument put forward in our study.

Before reviewing the relevant literature, it is important to first clarify some key terms. *Kanban* systems use a work-in-process cap to limit the work-in-process on the shop floor. This means orders are not directly released to the shop floor but are withheld. The place where orders are withheld (often in paper form) is referred to here as a pre-shop pool given the negative connotations associated with the term 'backlog' that is used, e.g. in Spearman *et al*. (1990). In the original *kanban* system, orders were released from this pre-shop pool according to the sequence in which they were stored in the *heijunka* box. This sequence itself was determined at Toyota using simple meta-heuristics (Monden, 1983).

2.1. *Kanban* Systems

Kanban systems are used to connect production stages or operations to one another to improve coordination, thereby regulating work-in-process and eliminating overproduction. While there have been some studies exploring the use of *kanban* systems in an assembly context (Faccio *et* *al*., 2013; Faccio *et al*., 2014, Khojasteh & Sato, 2015; Lolli *et al*., 2016), most studies focus on multi-stage production lines. Further, much of the extant literature on *kanban* systems has viewed them as inventory replenishment systems, where the same, known items are used, often realized in the form of *kanban* containers (e.g. Spearman *et al*., 1990; Berkley, 1992; Buzacott & Shanthikumar, 1992; Graves *et al*., 1995; Frein *et al*., 1995; Dallery & Liberopolous, 2000; Liberopoulos & Dallery, 2000; Diaz & Ardalan, 2010; Lage Junior & Godinho Filho, 2010; Gonzalez *et al*., 2012; Khojasteh & Sato, 2015; Chen & Sarker, 2015).

A widely recognized weakness of *kanban* systems is that they do not implicitly balance capacity and demand (workload). In the original *kanban* system, excess capacity was used to overcome this weakness (Kimura & Terada, 1981; Hopp & Spearman, 2004). Meanwhile, a broad body of research exists that instead sought to determine the optimal number of cards in the systems and/or to design rules for dynamically adjusting the number of *kanban* cards in response to variation in demand (see, e.g. Takahashi & Nakamura, 1999; Dallery & Liberopolous, 2000; Tardif & Maaseidvaag, 2001; Takahashi, 2003; Faccio *et al*., 2013; Renna *et al*., 2013; Xanthopoulos, *et al*., 2017).

In this study we focus on the card acquisition process. Kimura & Terada (1981) had already considered two different flow times: (i) the flow time between the moment when a *kanban* is removed from the container and the moment when production begins; and (ii) the flow time between the moment when production begins and the moment when the operation is completed. However, most research does not explicitly consider delays associated with the handling of *kanbans*. An exception is Lolli *et al*. (2016) who considered delays in *kanban* handling, but these delays were due to the *kanban* batch time and capacity constraints, i.e. when the need signaled by the *kanban* cannot be fulfilled directly. Meanwhile, electronic *kanban* systems are discussed in Kotani (2007) and Hofmann & Rüsch (2017); a major advantage of these systems is the instantaneous transfer of information that overcomes delays associated with information processing, as discussed in Gong *et al*. (2014). But all of these studies focused on delays in an inventory control problem. In contrast, we focus on the card acquisition process, and associated delays in information transfer, as it occurs in an order control problem.

A so-called common *kanban* system (Thürer *et al*., 2016) is illustrated in Figure 1 for a context where items are always the same and known beforehand. This context is called an inventory control problem as only inventory stocked at inventory decoupling points needs to be controlled. Since jobs are interchangeable in an inventory control problem, *kanban* cards are independent from individual jobs. Instead of referring to particular items or jobs, 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.

[Take in Figure 1]

In an order control problem, jobs are no longer interchangeable since they differ. In other words, *kanban* cards are dedicated to individual orders. As a consequence, the need cannot typically be directly satisfied, since the physical job associated with the *kanban* still has to be processed at upstream stations. Thus, two different processes – as illustrated in Figure 2 – occur:

1. *A (backwards) card acquisition process*: A job is released from the pool of orders when it acquires a card for the last station in its process. At this moment, it begins to move backwards through the shop, acquiring a *kanban* card at each station. During the acquisition process, the job only exists as a 'need' or on paper – this is a precursor to the actual production process.
2. *A (forwards) production process*: When the job arrives at its inventory/order separation point (via the above backwards acquisition process), the need can begin to be satisfied. The production process now starts and the job moves forwards from station to station, being processed at each one in turn (retracing in reverse the steps in the acquisition process). At each station, the physical job is now matched with its corresponding *kanban* card.

[Take in Figure 2]

Similar to Kimura & Terada (1981), there are two flow times: (i) the flow time between the moment when a *kanban* is designated to the job and the moment when production begins; and (ii) the flow time between the moment when production begins and the moment when the operation is completed. However, the former may be significantly longer in an order control problem since production cannot begin before the actual physical job arrives at the station.

2.2 Introduction to COBACABANA Systems

COBACABANA is the card-based version of a production control system known as Workload Control (Hendry & Kingsman, 1991; Bechte, 1994; Land & Gaalman, 1996, Bergamaschi *et al*., 1997; Cigolini & Portioli-Staudacher, 2002). Workload Control has been developed separately from card-based control systems such as *kanban* for over 30 years (Thürer *et al*., 2011). The separate research streams may have hidden the fact that COBACABANA (and consequently Workload Control) has a structure that is similar to the *kanban* system. A key difference is that, rather than acquiring cards individually at each station, control is centralized – card loops are established between each station and a centralized release function. This is illustrated in Figure 3.

In the original COBACABANA system (Land, 2009), the availability of cards authorized the planner to release new orders onto the shop floor. In other words, an order could only be released from the pool if it could acquire sufficient cards for each station in its routing from the centralized release function. COBACABANA, as presented here, follows the refinements proposed by Thürer *et al*. (2014) to Land's (2009) original card-based concept. This means two types of cards are used: (i) operation cards, which travel with an order and signal when an operation is complete; and, (ii) release cards, which visualize the shop floor's workload situation on a centralized planning board, as described below. Thus, rather than acquiring cards (from a limited amount of cards available per station), release cards have to be placed on a planning board that has a limited area. This allows the size of cards to represent the workload of an operation and supports workload balancing across stations.

[Take in Figure 3]

When the release decision takes place (which may either be periodically at fixed intervals or continuously whenever the system status changes), orders in the pool are sorted according to a so-called pool sequencing rule (Thürer *et al*., 2015b). 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*. To consider an order for release, the planner places the release card(s) at each station's area on the planning board (see Figure 4). The planner then compares the workload of each station with the predetermined workload limits. 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 floor, and the order is released. This release process continues until all orders in the pool have been considered for release once. The shop floor returns each operation card to the planner as soon as an operation has been 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.

[Take in Figure 4]

Figure 4 illustrates how the planning board is used when making a release decision. In this example, a new order with an operation at each station is considered for release. Since the second operation cannot be loaded into the station without exceeding the workload limit, the order is not released. The stack of release cards in each station's area on the planning board summarizes the workload released but not yet completed at a station. Thus, the planning board can also be understood as a real-time *Yamazumi* board.

2.3 Discussion: *Kanban* vs. COBACABANA

The *kanban* and COBACABANA systems were developed independent from each other. Yet, although hidden, they share a similar structure. There are only two key differences between the *kanban* and COBACABANA systems:

1. *The acquisition process*: While, in a *kanban* system, a card has to be acquired individually at each station; in a COBACABANA system, card acquisition (Land, 2009) or the posting of release cards (Thürer *et al*., 2014) is centralized and supported by a planning board.
2. *The meaning of cards*: While, in a *kanban* system, a card refers to a job (or better an operation of a job); in a COBACABANA system, a card represents the workload (e.g. in hours) of an operation.

In a recent comparison of major card-based control systems, Thürer *et al*. (2016) identified *kanban* systems as the preferred choice for an inventory control problem and COBACABANA as the preferred choice for an order control problem. But Thürer *et al*. (2016) based their judgment on a conceptual analysis of the structure underlying card-based control systems. They did not provide any results to back up their conclusion or to quantify the potential negative effect. A major issue put forward by Thürer *et al*. (2016) concerned delays in the card acquisition process; but jobs also have to wait until cards are acquired in a centralized card-acquisition process. In response, this study explores the impact of a decentralized *vs*. a centralized release decision in an order control problem, thereby addressing the following research question:

> *What is the best choice for an order control problem: a decentralized (kanban) or centralized (COBACABANA) release decision?*

Using controlled simulation experiments, we compare *kanban* with COBACABANA. This provides a unique insight into the difference between a centralized and decentralized release decision since this (and the meaning of cards) is the only difference between the two systems. In order to control for the second difference – the meaning of cards – a *kanban* system that limits the workload instead of the number of jobs at each station is evaluated in addition to the original *kanban* system. This system will be referred to as a load-based *kanban,* with its adapted acquisition process illustrated in Figure 5.

[Take in Figure 5]

3. Simulation Model

The shop and job characteristics modeled in the simulations are first outlined in Section 3.1 before the order release and dispatching rules applied are summarized in sections 3.2 and 3.3, respectively. 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 has been implemented in ARENA simulation software. In the pure flow shop, each job visits all stations in the same sequence in order of increasing station number. The pure flow shop was chosen since it is this environment in which *kanban* is typically applied in the literature. The shop contains six stations, where each station is a single resource with constant capacity. Since we model an order control problem, jobs are not interchangeable, which means that a job belongs to a certain customer from its arrival at the shop until delivery. Meanwhile, our model is stochastic, whereby processing times, inter-arrival times, and due dates are stochastic (random) variables.

Operation processing times follow a truncated lognormal distribution (Trietsch *et al*., 2012) with a truncated mean of 1 time unit and a maximum of 4 time units. Processing time variability is a factor that is likely to influence the card acquisition time of *kanban* systems and thus the performance difference between COBACABANA and *kanban*. Therefore, three levels of processing time variability are modeled, with a squared 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 to represent “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 part of the operation processing time.

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 and Acquisition Rule

As in previous simulation studies on *kanban* systems (Chang & Yih, 1994a; Gupta & Al-Turki, 1997; Gaury *et al*., 2001) and COBACABANA (Thürer *et al*., 2014, 2015a), 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 three release methods: *kanban*, load-based *kanban*, or COBACABANA. Since *kanban* systems take the release decision continuously, COBACABANA has also been implemented as a continuous release method; COBACABANA’s release mechanism is triggered whenever an operation is completed or a new job arrives at the shop.

Seven workload limits are applied, ranging from 4 to 10 jobs for *kanban* and from 4 to 10 time units for load-based *kanban* and COBACABANA. These limits apply to the first station. In *kanban* systems and COBACABANA, a card is linked to a job until the corresponding operation is completed. This means cards represent the direct load (if the job arrived at the station related to the card) and the indirect load (if the job is still upstream from the station related to the card). As a consequence, the workload limit should be increased for downstream stations (the further downstream a station, the higher its indirect load). The limit imposed at each station is accordingly increased by multiplying the limit for the first station by the station number (from 2 to 6). As a baseline measure, experiments without controlled order release have also been executed, i.e. where jobs are released onto the shop floor immediately upon arrival.

Finally, COBACABANA uses a pool sequencing rule to determine the sequence in which orders are considered for release from the pool. A similar rule is needed for our two *kanban* systems. Three different rules will be applied:

- *First in System First Served (FSFS)*: The job that started the acquisition process first receives the card first. This rule is applied as a baseline measure.
- *Earliest Due Date (EDD)*: The job that has the earliest due date receives the card first. This rule ensures that the most urgent job is considered first.
- *Shortest Processing Time (SPT)*: The job that has the shortest processing time at the upstream station receives the card first for our two *kanban* systems. For COBACABANA, the sequence is determined based on the processing time at the first station. This is a simple load-based rule.

3.3 Dispatching Rule

The dispatching rule determines the priority of jobs during the actual production process. In this study, the Modified Operation Due Date (MODD; e.g. Baker & Kanet, 1983) rule is used since it was recently identified as the best solution in pure flow shops by Thürer *et al*. (2015a). The MODD rule prioritizes jobs according to the lowest priority number, which is given by the maximum of the operation due date δ_{ij} and earliest finish time, i.e. $max(\delta_{ij}, t+p_{ij})$ 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 Shortest Processing Time (SPT) effects – during periods of high load, i.e. when multiple jobs exceed their ODD (Land *et al*., 2015). Meanwhile, the calculation of the operation due date δ_{ij} for the i^{th} operation of a job j follows Equation (1) 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 a constant allowance c_s from the operation due date of the next operation. This allowance is based on the actually realized operation throughput times at each station s corresponding to the operation.

$$\delta_{ij} = \delta_j - (n_j - i) \cdot c_s \;\; i{:}1 \ldots n_j \qquad (1)$$

3.4 Experimental Design and Performance Measures

The experimental factors are: (i) the three release methods (*kanban*, load-based *kanban,* and COBACABANA); (ii) the seven workload limits (4 to 10 jobs or time units); (iii) the three levels of pool sequencing/acquisition rule (FSFS, EDD and SPT); and, (iv) the three levels of processing time variability (low, medium, and high). A full factorial design with 189 cells was used, where each cell 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 four principal performance measures considered in this study are as follows: (i) the *(shop floor) throughput time* – the mean of the completion date minus the start date of the production process across jobs; (ii) the *total throughput time* – the mean of the completion date minus the job entry date across jobs; (iii) the *percentage tardy* – the percentage of jobs completed after the due date; and, (iv) the *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*). In addition – and to evaluate the impact of *kanban*'s card acquisition process – we also consider the acquisition time at each station and its standard deviation as auxiliary performance measures (see Section 4.2).

4. Results

Statistical analysis of our results was conducted using an ANOVA, with the results presented in Table 1. The ANOVA is here based on a block design with the workload limit as the blocking factor, i.e. the seven levels for the workload limit were treated as different systems. A block design allowed the main effect of the workload limit and the main and interaction effects of the release method, pool sequencing/acquisition rule, and processing time variability to be captured.

[Take in Table 1]

All main effects, two-way interactions, and three-way interactions were shown to be statistically significant (α = 0.05). The Scheffé multiple-comparison procedure was used to further examine the significance of the differences between the outcomes of the individual release methods and the pool sequencing/acquisition rules. Table 2 and Table 3 summarize the 95% confidence intervals for the release methods and pool sequencing/acquisition rules, respectively. Differences are considered not significant if this interval includes zero. Significant differences between the outcomes of all release methods can be identified from Table 2. Meanwhile, from Table 3, significant differences for most performance measures can be identified except for between FSFS and EDD, which perform statistically equivalent.

[Take in Table 2 & Table 3]

Detailed performance results are presented next in Section 4.1 for medium processing time variability. The card acquisition process is then examined in more detail in Section 4.2 where our auxiliary performance measures on card acquisition time are presented. Finally, Section 4.3 assesses the robustness of our results to processing time variability.

4.1 Performance Assessment

Figures 6a, 6b, and 6c show the total throughput time, percentage tardy, and mean tardiness results over the throughput time results, respectively. Only results for medium processing time variability are presented here, being the impact of processing time variability assessed in Section 4.3. Results are presented in the form of performance curves, where the left-hand starting point of the curves represents the tightest workload limit (4 jobs or time units). The limit increases step-wise by moving from left to right in each graph, with each data point representing one limit (from 4 to 10 jobs or time units). Loosening the limits increases the workload on the shop floor and, as a result, the shop floor throughput times. In addition, and as a reference point, the result obtained when orders are released immediately is also included. This result 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 shop floor throughput times.

[Take in Figure 6]

The following can be observed from our results:

- *Release Method:* COBACABANA outperforms *kanban* and load-based *kanban* for all experimental settings. Meanwhile, load-based *kanban* realizes shorter throughput times, and thus work-in-process levels, for a given norm level than *kanban* since: load-based *kanban* can not completely fill up the norm due to the granularity of the workload and *kanban* only controls the number of jobs, not imposing a strict upper bound on the workload. With regards to our research question, it can be seen that centralizing the card acquisition process leads to significant performance improvements compared to a decentralized acquisition process for an order control problem. Whether this is really due to the card acquisition process will be examined in more detail in the next section.

- *Pool sequencing/Acquisition Rule:* As expected from our multiple-comparison results, the performance difference between FSFS and EDD is negligible. In fact, curves for FSFS and EDD mostly overlap for each release method. Meanwhile, SPT leads to the expected effect of reduced total throughput times and a lower percentage tardy at the expense of mean tardiness performance.

4.2 Performance Analysis: The Card Acquisition Process

Our results above demonstrated the superior performance of COBACABANA compared to *kanban* and load-based *kanban*. It is argued that the performance difference between the two *kanban* systems and COBACABANA is explained by the card acquisition process. To better understand this card acquisition process, the acquisition time and the standard deviation of the acquisition time for each experimental setting of the workload limit were collected at each station. Results for *kanban* are presented in Table 4. Only results for EDD and SPT sequencing are presented since the FSFS rule performed statistically equivalent to EDD. As a reference, the card acquisition time for COBACABANA's centralized acquisition process is also given. This is equivalent to the pool waiting time, i.e. the total throughput time minus the shop floor throughput time.

[Take in Table 4]

The following can be observed from the results:

- *EDD Acquisition Rule:* If the limits are loose, the acquisition time at upstream stations (e.g. Station 1) is higher than that at downstream stations (e.g. Station 5) since upstream stations are more tightly controlled (given that the number of cards allowed is dependent on station position). In contrast, if the limits are tight, fewer jobs are allowed to enter the acquisition process; the time to acquire a card at Station 5 increases while the time to acquire a card at an upstream station (e.g. at Station 1) decreases. A similar pattern can be observed for the standard deviation of the acquisition time. As expected, the acquisition of the card at the 6th station (the beginning of the acquisition process) determines performance since it dictates when a job is released from the pool. Hence, this acquisition time reflects the pool waiting time. It is much higher at tighter norms when compared to COBACABANA's pool waiting time because COBACABANA incorporates a load balancing capability that smoothes the workload across stations, which is lacking in *kanban*.

- *SPT Acquisition Rule:* Compared to EDD, a general increase in acquisition times can be observed for the SPT rule at tighter norms, except for the 6th station and a limit of 4 jobs. This result however must be interpreted with care given the literal explosion in the standard deviation of acquisition time results. As somewhat expected for the SPT rule, the standard deviation increases significantly for all settings. These SPT effects are only active during the acquisition process. If we refer back to Figure 6 and compare the shop floor throughput time results for the SPT and EDD rules then we observe a negligible effect.

The above highlights that *kanban*'s decentralized card acquisition process is not able to balance the workload. On the other hand, load balancing is realized effectively by COBACABANA, where the increase in pool waiting time is offset by a reduction in shop floor throughput times. But is this detrimental effect under *kanban* really due to its decentralized release decision? Or could it be due to the second difference between *kanban* systems and COBACABANA, i.e. the meaning of cards. In COBACABANA, cards are of an adjustable size to support load balancing. Rather than controlling the number of cards, the size of the stack of cards released to a station – and thus the workload represented by this stack – is controlled.

To control for this second difference, a similar change in the meaning of cards has been introduced in the form of load-based *kanban*. The acquisition time and the standard deviation of the acquisition time obtained for load-based *kanban* are given in Table 5. This is again provided for each limit level and each station together with COBACABANA's pool waiting time as a reference. The results show a similar pattern to the results for *kanban*, but both the acquisition time and the standard deviation of the acquisition time increase significantly. A major reason is that load-based *kanban* further strengthens SPT effects since jobs with large operations find it more difficult to fit within the limit. This identifies the decentralization of the card acquisition process as the primary cause of the poor performance of *kanban* compared to COBACABANA.

[Take in Table 5]

Load-based *kanban* decentralizes the load balancing mechanism inherent to COBACABANA, but it does so whilst only 'seeing' one station, e.g. Station 6. Yet, a good balance at Station 6 does not necessarily lead to a set of jobs that creates a good balance at all other stations. A vital element of load balancing in an order control problem is that the mix of jobs has to balance the workload across stations. On the contrary, in an inventory control problem, each station is decoupled from the others by an inventory buffer – so each inventory buffer can be controlled locally (or independently). In this context, no load balancing is required.

Our results demonstrate that the order control problem requires a centralized release decision that takes a global view of the shop. Meanwhile, the inventory control problem favors a decentralized control decision; stations are decoupled by inventory, which simplifies the problem since load balancing is typically not required. This allows for local control and avoids the need for a more complex, centralized release decision.

4.3 Assessment of the Robustness of Results: Processing Time Variability

To assess the impact of processing time variability on performance, Figure 7a and Figure 7b show the lead time, percentage tardy, and mean tardiness results over the throughput time results for low and high processing times, respectively. As somewhat expected, SPT effects and load balancing become more important with higher processing time variability (see also Thürer *et al*., 2015b). As a result:

- *SPT Effects:* The performance differences between the EDD/FSFS and the SPT pool sequencing/acquisition rules increase with high processing time variability (Figure 7b) while they diminish with low processing time variability (Figure 7a) when compared to the results obtained with medium processing time variability (Figure 6, in Section 4.1 above).
- *Load Balancing:* The performance differences between our two *kanban* systems and COBACABANA increase when there is high processing time variability (Figure 7b) and diminish with low processing time variability (Figure 7a) compared to the results obtained with medium processing time variability (Figure 6, in Section 4.1 above).

[Take in Figure 7]

In general, the main conclusion of our study is that COBACABANA, with its centralized card acquisition process, outperforms *kanban* systems that decentralize the card acquisition process in the order control problem. This conclusion is not affected by processing time variability.

5. Conclusions

Kanban systems have gained in popularity since first being introduced in the 1960s. They are simple yet effective means of controlling production and are consequently widely applied in practice. However, while there is an extensive literature on *kanban* systems for inventory control problems, the inner workings of a *kanban* system in the context of an order control problem have never been analyzed. Our analysis reveals that, in an order control problem, a card needs to be acquired by a job at each station in its routing before the physical job can start to be processed. Therefore, for *kanban* in the context of an order control problem, there needs to be a decentralized card acquisition process (during which information is propagated backwards from station to station) that is separated from the actual production process. The same control structure as *kanban* underpins Control of Balance by Card-based Navigation (COBACABANA), a card-based control system based on the Workload Control concept that has been developed separately from card-based control systems such as *kanban* for over 30 years. A key difference between *kanban* and COBACABANA is that COBACABANA centralizes the control decision and consequently the card acquisition process. Comparing the performance of *kanban* and COBACABANA thus provides a unique opportunity to compare decentralized and centralized control decisions.

In answer to our research question concerning the best choice for an order control problem – a decentralized (*kanban*) or centralized (COBACABANA) release decision – we have demonstrated that COBACABANA outperforms *kanban* across all important performance measures in our simulated pure flow shop environment. Further analysis revealed that it is the centralized card acquisition process that allows COBACABANA to balance the workload across stations, leading to superior performance. We can therefore conclude that a centralized decision process is required in an order control problem since it allows for load balancing.

5.1 Managerial Implications

The main implication from our study is that if load balancing is to be realized it has to take all stations into account. While this finding was obtained by comparing two quite specific production control systems (i.e. *kanban* and COBACABANA), we would argue that it is applicable to a broader range of production control approaches in practice. The decentralized mechanism incorporated in *kanban* only considers one station, which prohibits effective load balancing. An alternative would be to provide the workload information from all stations at each station. While this is, in our opinion, not realizable with cards, it could be embedded in an electronic *kanban* system. But this system would create huge information redundancies compared to COBACABANA, where all the information is gathered at only one point (instead of at each station). Finally, in a simpler inventory control problem, where load balancing is typically not required, centralized control is not necessary and control can be exercised locally through a decentralized approach.

5.2 Limitations and Future Research

The main limitation of our study is the environmental setting. For example, we chose a simple pure flow shop since this is the environment where *kanban* is arguably most often applied in the academic literature. However, Hopp & Spearman (2001, p 470) have highlighted that *kanban* systems naturally provide a mechanism for sharing a resource among different routings. Thus, future research could explore the performance differences between COBACABANA and *kanban* systems for more complex routings, potentially including different inventory/order separation points. This leads to the important issue of nested systems. Card-based control systems (or control systems in general) are typically applied separately. But if different control problems exist, different solutions need to be nested or combined. For example, COBACABANA could be nested inside a *kanban* system where *kanban* is used to control the assembly order (by controlling the inventory points of different shop floors that produce sub-assemblies) and COBACABANA is used to control the shop floor (since it represents an order control problem). Finally, our results demonstrate that an order control problem requires a centralized release decision. COBACABANA is one way to realize this, but future research could design different release methods, potentially improving the performance of COBACABANA or enhancing its applicability.

References

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

Bechte, W., 1994, Load-oriented manufacturing control just-in-time production for job shops, *Production Planning &Control*, 5, 3, 292 – 307.

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

Berkley, B.J., 1992, A review of the kanban production control research literature, *Production & Operations Management*, 1, 4, 393-411.

Buzacott, J.A., and Shanthikumar, 1992, A general approach for coordinating production in multiple-cell manufacturing systems, *Production & Operations Management*, 1, 1, 34-52.

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.

Chen, Z and Sarker, B.R., 2015, Optimisation of multi-stage JIT production-pricing decision: centralised and decentralised models and algorithms, *International Journal of Production Research*, 53, 20, 6210-6230.

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

Dallery, Y., and Liberopoulos, G., 2000, Extended kanban control system: combining kanban and base stock, *IIE Transactions*, 32, 4, 369–386.

Diaz, R., and Ardalan, A., 2010, An Analysis of Dual-Kanban Just-In-Time Systems in a Non-Repetitive Environment, *Production & Operations Management*, 19, 2, 233–245.

Faccio, M., 2014, The impact of production mix variations and models varieties on the parts-feeding policy selection in a JIT assembly system, *International Journal of Advanced Manufacturing Technology*, 72, 543-560.

Faccio, M., Gamberi, M., and Persona, A., 2013, Kanban number optimisation in a supermarket warehouse feeding a mixed-model assembly system, *International Journal of Production Research*, 5, 10, 2997-3017.

Frein, Y., Di Mascolo, M., Dallery, Y., 1995, On the design of generalized kanban control systems, *International Journal of Operations & Production Management*, 15, 9, 158 – 184.

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.

Gong, Q., Yang, Y., and Wang, S., 2014, Information and decision-making delays in MRP, KANBAN, and CONWIP, *International Journal of Production Economics*, 156, 208-213.

Gonzalez, P.L., Framinan, J.M, Pierreval, H., 2012, Token-based pull production control systems: an introductory overview, *Journal of Intelligent Manufacturing*, 23, 5-22.

Graves, R.J., Konopka, J.M., and Milne, R.J., 1995, Literature review of material flow control mechanisms, *Production Planning & Control*, 6, 5, 395 – 403.

Gupta, S.M., and Al-Turki, Y.A.Y, 1997, An algorithm to dynamically adjust the number of Kanbans in stochastic processing times and variable demand environment, *Production Planning & Control*, 8, 2, 133-141.

Hendry, L.C., and Kingsman, B.G., 1991, A decision support system for job release in make to order companies, *International Journal of Operations & Production Management*, 11, 6-16.

Hofmann, E., and Rüsch, M, 2017, Industry 4.0 and the current status as well as future prospects on logistics, *Computers in Industry*, 89, 23–34.

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.

Hopp, W.J., and Spearman, M.L., 2001, *Factory Physics: Foundations of Manufacturing Management*, McGrawHill, 2nd Edition.

Kimura, O., and Terada, H., 1981, Design and analysis of Pull System, a method of multi-stage production control, *International Journal of Production Research*, 19, 3, 241-253.

Kotani, S., 2007, Optimal method for changing the number of kanbans in the e -Kanban system and its applications, *International Journal of Production Research*, 45, 24, 5789-5809.

Khojasteh, Y. and Sato R., 2015, Selection of a pull production control system in multi-stage production processes, *International Journal of Production Research*, 53, 14, 4363-4379.

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

Liberopoulos, G., and Dallery, Y., 2000, A unified framework for pull control mechanisms in multi-stage manufacturing systems, *Annals of Operations Research*, 93, 325 – 355.

Lolli, F., Gamberini, R., Giberti, C., Rimini, B., and Bondi, F., 2016, A simulative approach for evaluating alternative feeding scenarios in a kanban system, *International Journal of Production Research*, 54:14, 4228-4239.

Monden, Y., 1983, *Toyota Production System: Practical Approach to Production Management*, Industrial Engineering and Management Press, Norcross, Georgia.

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.

Renna, P., Magrino, L., Zaffina, R., 2013, Dynamic card control strategy in pull manufacturing systems, *International Journal of Computer Integrated Manufacturing*, 26, 9, 881-894.

Schonberger, R.J, 1983, Application of Single-Card and Dual-Card kanban, *Interfaces*, 13, 4, 56-67.

Shingo, S., 1989, *A Study of the Toyota Production System from an Industrial Engineering Viewpoint*, Cambridge, MA: Productivity Press.

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.

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.

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., and Silva, C., 2011, Three Decades of Workload Control Research: A Systematic Review of the Literature, *International Journal of Production Research*, 49, 23, 6905-6935.

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., and Protzman, C.W., 2015a, COBACABANA (Control of Balance by Card Based Navigation): An Alternative to *Kanban* in the Pure Flow Shop?, *International Journal of Production Economics*, 166, 143-151.

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

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

Trietsch, D., Mazmanyan, 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.

Xanthopoulos, A.S., Koulouriotis, D.E., and Gasteratos, A., 2017, Adaptive card-based production control policies, *Computers & Industrial Engineering*, 103, 131–144.

Table 1: ANOVA Results

	Source of Variance	Sum of Squares	Degree of Freedom	Mean Squares	F-Ratio	p-Value
Shop Floor Throughput Time	Workload Limit	344651.09	6	57441.85	7538.23	0.00
	Release Method (R)	22933.34	2	11466.67	1504.80	0.00
	Sequen/Acquisition (S)	55.22	2	27.61	3.62	0.03
	Proc Time Variability (V)	377001.46	2	188500.73	24737.41	0.00
	R x S	1591.76	4	397.94	52.22	0.00
	R x V	10428.00	4	2607.00	342.12	0.00
	S x V	82.54	4	20.64	2.71	0.03
	R x S x V	777.68	8	97.21	12.76	0.00
	Residual	143767.81	18867	7.62		
Total Throughput Time	Workload Limit	39559112.00	6	6593185.40	676.11	0.00
	Release Method (R)	26444388.00	2	13222194.00	1355.90	0.00
	Sequen/ Acquisition (S)	9183027.70	2	4591513.80	470.85	0.00
	Proc Time Variability (V)	70707419.00	2	35353709.00	3625.44	0.00
	R x S	6530461.50	4	1632615.40	167.42	0.00
	R x V	33488942.00	4	8372235.40	858.55	0.00
	S x V	15507843.00	4	3876960.90	397.57	0.00
	R x S x V	10506443.00	8	1313305.30	134.68	0.00
	Residual	184000000.00	18867	9751.58		
Percentage Tardy	Workload Limit	29.94	6	4.99	228.43	0.00
	Release Method (R)	250.37	2	125.19	5731.55	0.00
	Sequen/ Acquisition (S)	117.79	2	58.89	2696.34	0.00
	Proc Time Variability (V)	816.11	2	408.06	18682.38	0.00
	R x S	37.81	4	9.45	432.79	0.00
	R x V	154.77	4	38.69	1771.54	0.00
	S x V	59.60	4	14.90	682.14	0.00
	R x S x V	17.74	8	2.22	101.55	0.00
	Residual	412.09	18867	0.02		
Mean Tardiness	Workload Limit	38681059.00	6	6446843.20	672.61	0.00
	Release Method (R)	23445072.00	2	11722536.00	1223.03	0.00
	Sequen/ Acquisition (S)	8138634.80	2	4069317.40	424.56	0.00
	Proc Time Variability (V)	55293678.00	2	27646839.00	2884.44	0.00
	R x S	6248487.50	4	1562121.90	162.98	0.00
	R x V	32264464.00	4	8066115.90	841.55	0.00
	S x V	15040936.00	4	3760234.10	392.31	0.00
	R x S x V	10414193.00	8	1301774.10	135.82	0.00
	Residual	180800000.00	18867	9584.82		

Table 2: Results for Scheffé Multiple Comparison Procedure:
Release Method

Release Method (x)	Release Method (y)	Shop Floor Throughput Time		Total Throughput Time		Percentage Tardy		Mean Tardiness	
		lower[1]	upper	lower	upper	lower	upper	lower	upper
L-*Kanban*	*Kanban*	-2.78	-2.54	38.28	46.90	0.03	0.04	37.63	46.17
COBA	*Kanban*	-1.05	-0.81	-53.27	-44.65	-0.23	-0.22	-48.63	-40.09
COBA	L-*Kanban*	1.60	1.85	-95.86	-87.24	-0.27	-0.25	-90.53	-81.99

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

Table 3: Results for Scheffé Multiple Comparison Procedure:
Pool Sequencing/Acquisition Rule

Sequencing Rule (x)	Sequencing Rule (y)	Shop Floor Throughput Time		Total Throughput Time		Percentage Tardy		Mean Tardiness	
		lower[1]	upper	lower	upper	lower	upper	lower	upper
FSFS	EDD	-0.19*	0.05	-5.61*	3.01	-0.01*	0.01	-5.52*	3.02
SPT	FSFS	-0.25	-0.01	-51.70	-43.09	-0.17	-0.16	-48.90	-40.36
SPT	EDD	-0.18	0.06*	-50.40	-41.79	-0.17	-0.16	-47.65	-39.11

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

Table 4: Analysis of Card Acquisition Time: COBACABANA vs. Kanban

			Limit 4	Limit 5	Limit 6	Limit 7	Limit 8	Limit 9	Limit 10
EDD	COBACABANA	Pool	9.91	6.49	4.69	3.52	2.72	2.14	1.70
	Kanban Acquisition Time	Station 6	63.16	17.58	8.13	3.87	2.12	1.18	0.66
		Station 5	1.42	1.25	0.76	0.77	0.58	0.42	0.31
		Station 4	1.04	0.99	0.82	0.72	0.60	0.48	0.37
		Station 3	0.82	0.83	0.76	0.70	0.62	0.54	0.45
		Station 2	0.65	0.71	0.71	0.70	0.66	0.61	0.55
		Station 1	0.49	0.58	0.64	0.68	0.70	0.69	0.67
	Kanban SD Acquisition Time	Station 6	34.29	18.81	12.72	8.37	5.75	3.92	2.57
		Station 5	1.79	2.06	1.73	2.05	1.88	1.65	1.40
		Station 4	1.61	1.87	1.94	1.99	1.93	1.80	1.62
		Station 3	1.46	1.74	1.88	1.96	1.97	1.92	1.81
		Station 2	1.31	1.60	1.80	1.94	2.02	2.03	2.01
		Station 1	1.13	1.43	1.68	1.88	2.03	2.12	2.18
SPT	COBACABANA	Pool	7.58	5.10	3.76	2.86	2.22	1.75	1.41
	Kanban Acquisition Time	Station 6	44.98	28.62	12.24	4.97	2.40	1.17	0.61
		Station 5	1.69	1.80	1.39	0.92	0.63	0.42	0.28
		Station 4	1.27	1.46	1.22	0.87	0.65	0.48	0.35
		Station 3	1.05	1.27	1.12	0.87	0.68	0.53	0.42
		Station 2	0.89	1.15	1.07	0.88	0.74	0.61	0.51
		Station 1	0.77	1.11	1.14	1.05	0.95	0.82	0.73
	Kanban SD Acquisition Time	Station 6	284.13	177.95	76.11	34.45	18.32	10.55	6.13
		Station 5	3.74	4.85	5.00	4.62	4.03	3.43	2.78
		Station 4	3.10	4.19	4.52	4.31	4.08	3.71	3.23
		Station 3	2.79	3.84	4.25	4.12	3.99	3.82	3.52
		Station 2	2.50	3.52	3.92	3.99	3.94	3.85	3.66
		Station 1	2.07	3.08	3.77	4.18	4.34	4.25	4.07

Table 5: Analysis of Card Acquisition Time: COBACABANA vs. Load-Based Kanban

			Limit 4	Limit 5	Limit 6	Limit 7	Limit 8	Limit 9	Limit 10
EDD	COBACABANA	Pool	9.91	6.49	4.69	3.52	2.72	2.14	1.70
	L-*Kanban* Acquisition Time	Station 6	221.84	44.10	12.20	5.33	2.76	1.47	0.80
		Station 5	3.18	2.80	1.90	1.32	0.92	0.64	0.43
		Station 4	1.93	1.83	1.42	1.09	0.85	0.67	0.51
		Station 3	1.48	1.44	1.21	1.02	0.86	0.71	0.58
		Station 2	1.23	1.22	1.10	0.98	0.88	0.80	0.71
		Station 1	1.10	1.13	1.08	1.05	1.01	0.97	0.93
	L-*Kanban* SD Acquisition Time	Station 6	82.44	29.60	15.86	10.16	6.86	4.61	3.07
		Station 5	4.19	4.33	3.92	3.45	2.93	2.46	1.96
		Station 4	3.05	3.26	3.17	2.97	2.74	2.50	2.18
		Station 3	2.55	2.77	2.81	2.77	2.66	2.50	2.30
		Station 2	2.16	2.38	2.51	2.57	2.58	2.56	2.50
		Station 1	1.80	2.05	2.27	2.43	2.55	2.63	2.68
SPT	COBACABANA	Pool	7.58	5.10	3.76	2.86	2.22	1.75	1.41
	L-*Kanban* Acquisition Time	Station 6	50.54	41.28	21.82	9.39	3.70	1.43	0.67
		Station 5	3.06	3.38	2.79	1.80	1.07	0.62	0.38
		Station 4	1.83	2.22	1.99	1.48	0.98	0.64	0.44
		Station 3	1.36	1.74	1.70	1.34	0.97	0.69	0.49
		Station 2	1.09	1.49	1.57	1.33	1.02	0.78	0.61
		Station 1	0.87	1.17	1.19	1.08	0.94	0.82	0.72
	L-*Kanban* SD Acquisition Time	Station 6	311.59	249.76	130.35	59.31	26.68	11.95	6.51
		Station 5	8.46	10.01	9.89	8.28	6.36	4.79	3.58
		Station 4	5.30	6.76	7.09	6.62	5.64	4.71	3.87
		Station 3	4.07	5.44	6.04	5.88	5.26	4.52	3.89
		Station 2	3.23	4.50	5.31	5.46	5.16	4.73	4.19
		Station 1	2.92	4.26	5.06	5.48	5.57	5.57	5.72

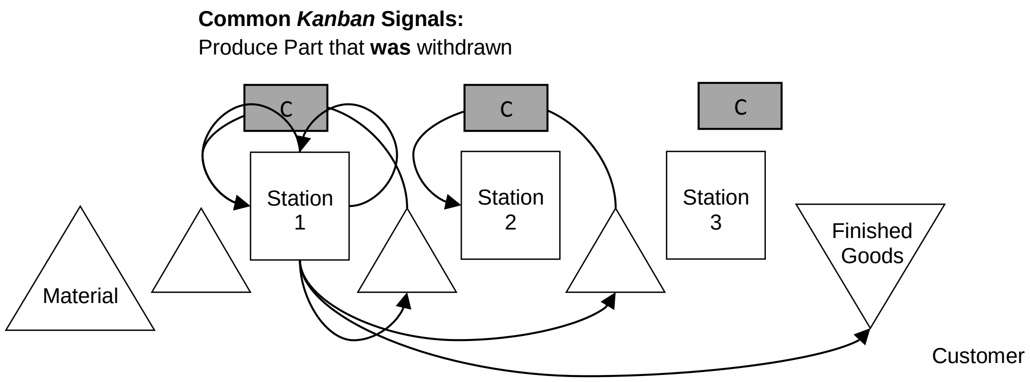

Common *Kanban* Signals:
Produce Part that **was** withdrawn
C
C
C
Station 1
Station 2
Station 3
Material
Finished Goods
Customer

Figure 1: A Common Kanban System used for Coordinating Three Stations in an Inventory Control Problem

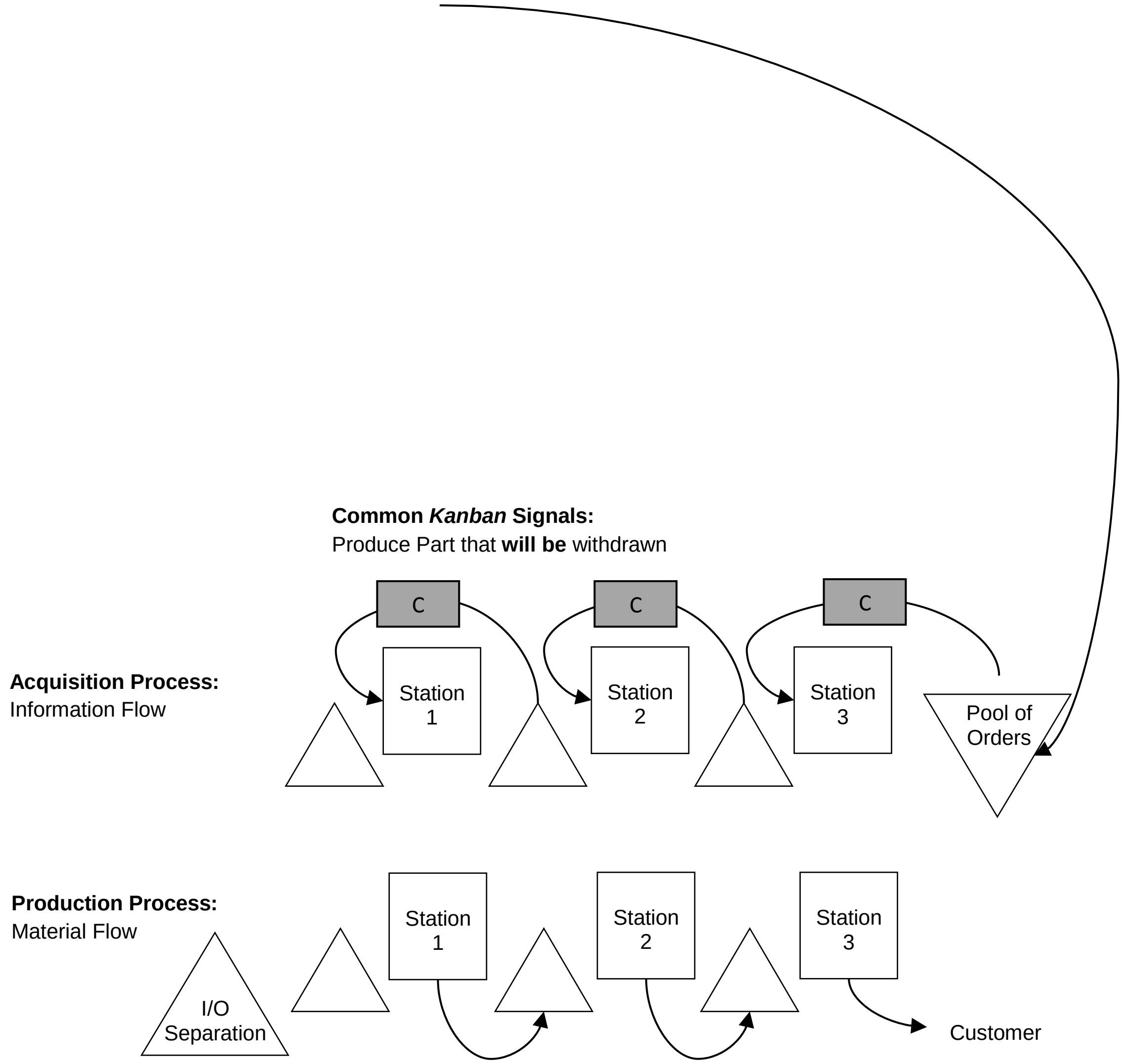


Figure 2: A Common Kanban System used for Coordinating Three Stations in an Order Control Problem

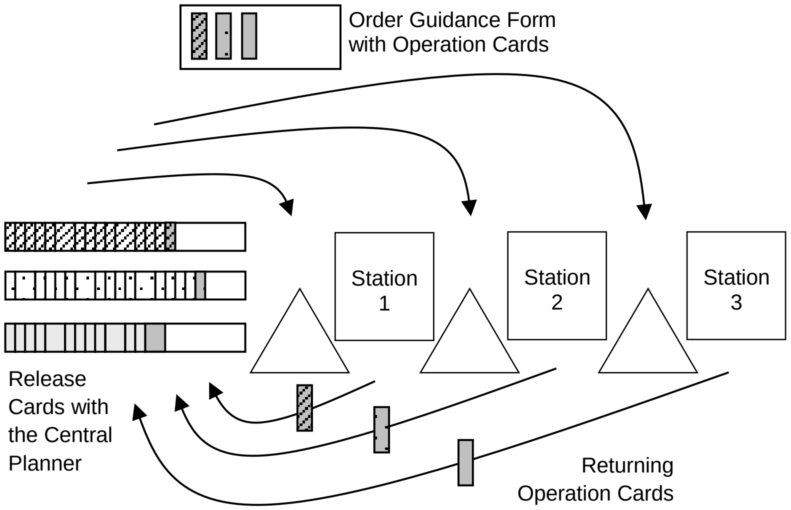

Order Guidance Form with Operation Cards
Station 1
Station 2
Station 3
Release Cards with the Central Planner
Returning Operation Cards

Figure 3: COBACABANA System – Card-based Order Release with Loops between the Central Planner and the Stations on the Shop Floor

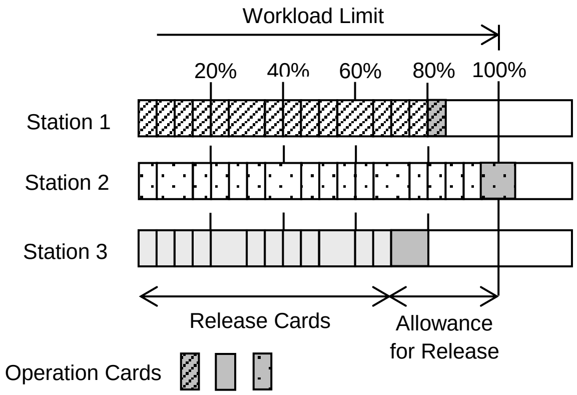


Figure 4: The Planner's Planning Board for Order Release (with an Example Release Decision)

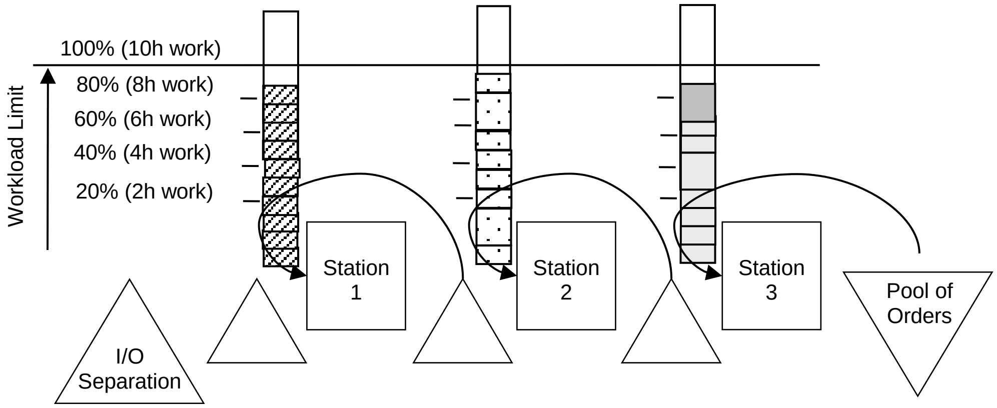
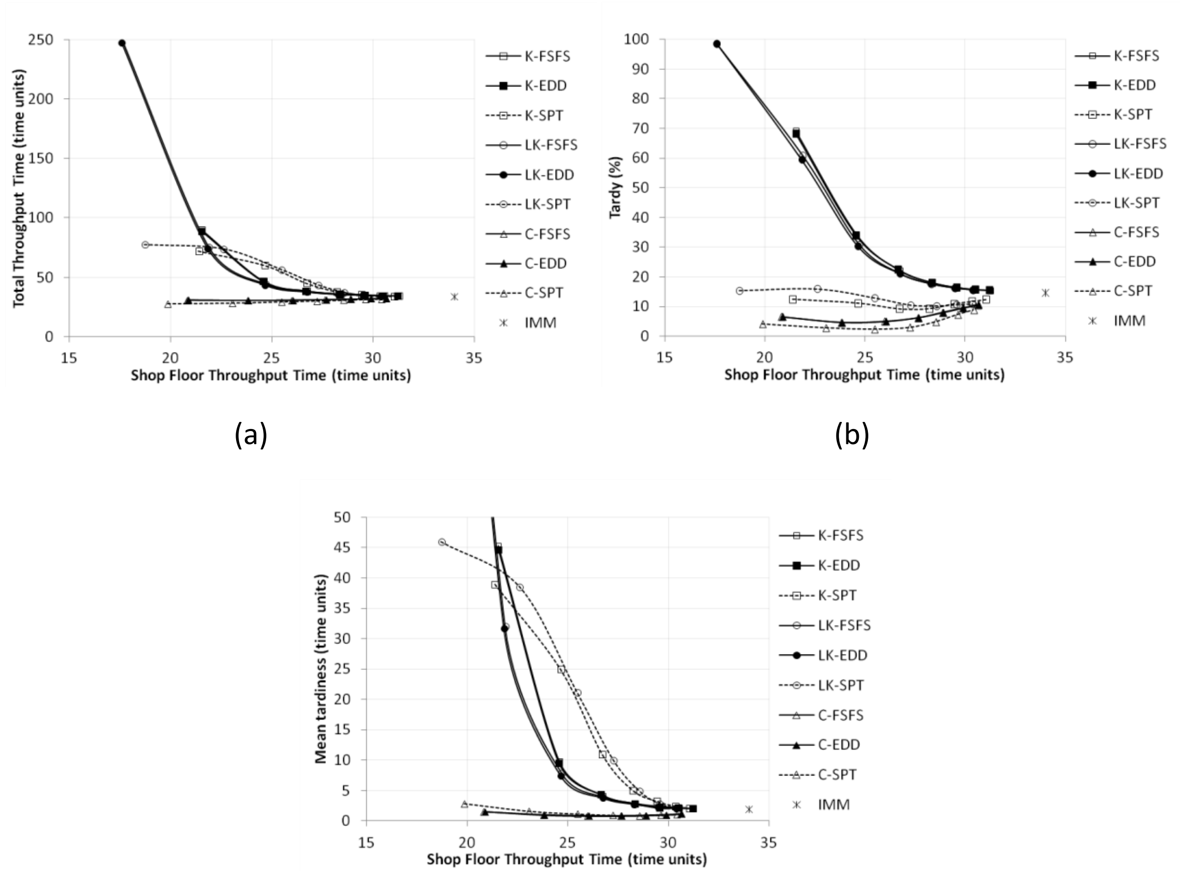


Figure 5: Acquisition Process of a Load-based Kanban System used for Coordinating Three Stations



(a)

(b)

(c)

Figure 6: Performance Assessment with Medium Processing Time Variability:
(a) Total Throughput Time; (b) Percentage Tardy; and, (c) Mean Tardiness over the Shop Floor Throughput Time

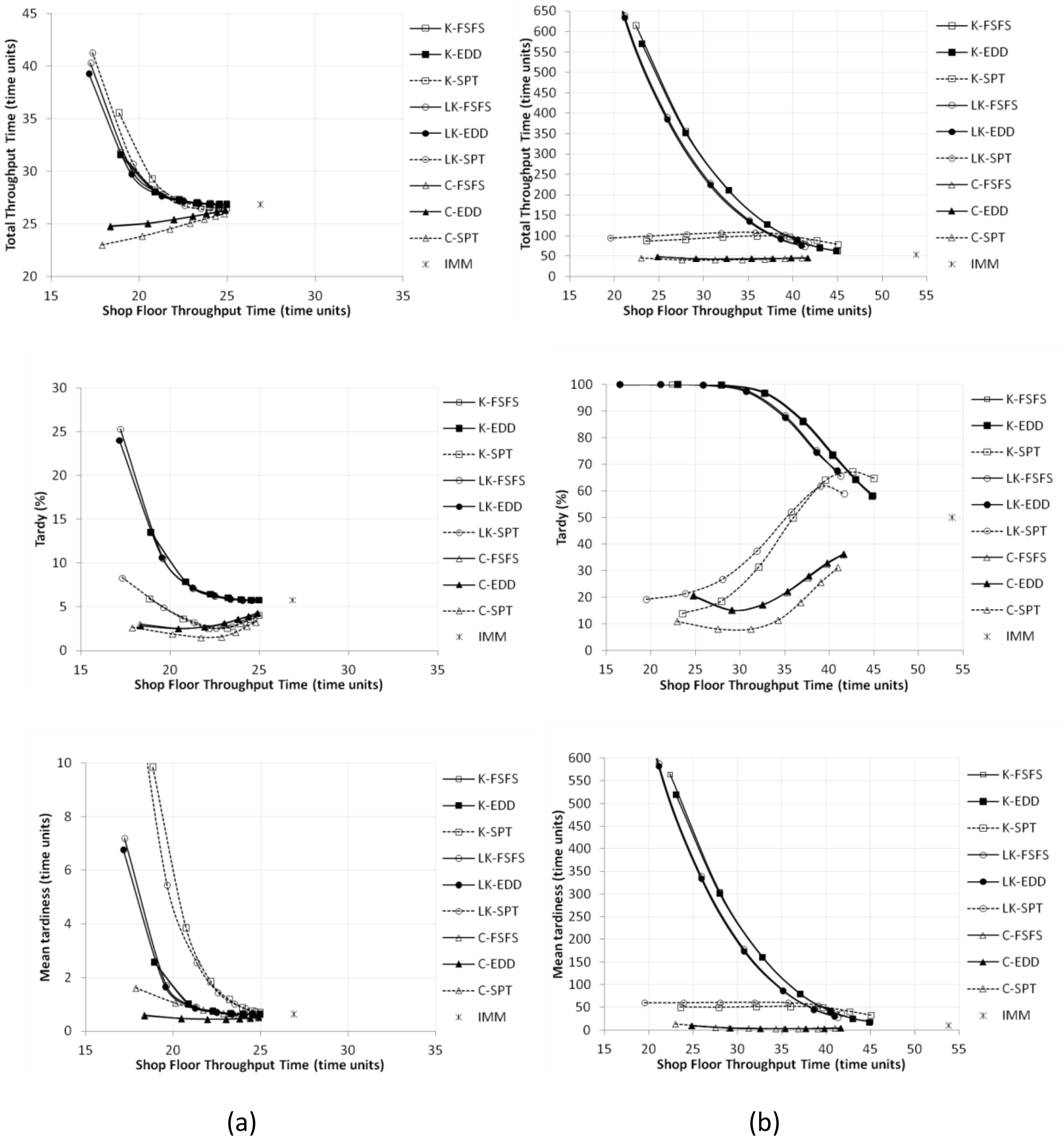


Figure 7: Performance Assessment with:
(a) Low Processing Time Variability; and, (b) High Processing Time Variability