Self-Organizing Material Flow Control using Smart Products: An Assessment by Simulation

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

Name:	Prof. Matthias Thürer
Institution:	Jinan University
Address:	School of Intelligent Systems Science and Engineering Jinan University (Zhuhai Campus) 519070, Zhuhai, PR China
E-mail:	matthiasthurer@workloadcontrol.com

Name:	Prof. Nuno O. Fernandes
Institution 1:	Instituto Politécnico de Castelo Branco
Address:	Av. do Empresário, 6000-767 Castelo Branco - Portugal
Institution 2:	ALGORITMI Research Unit, University of Minho
Address:	Campus de Gualtar, 4710-057 Braga - 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:	School of Intelligent Systems Science and Engineering Jinan University (Zhuhai Campus) 519070, Zhuhai, PR China
E-mail:	quting@jnu.edu.cn

Name:	Prof. George Q. Huang
Institution:	University of Hong Kong
Address:	Department of Industrial Manufacturing Systems Engineering Haking Wong Building, Pokfulam Road
E-mail:	gqhuang@hku.hk

Keywords: *Material Flow Control; Smart Product; Internet of Things; POLCA; Dispatching.*

Self-Organizing Material Flow Control using Smart Products: An Assessment by Simulation

Abstract

Material Flow Control (MFC) mechanisms control the movement of jobs through a set of stationery capacity resources on the shop floor. Although the objective of MFC is item-centric, i.e. to control the flow of individual jobs, most existing MFC mechanisms are resource-centric, i.e. focus on managing the capacity resources. While this was justified by technical constraints on real-time information feedback, advances in technology allow for new designs. In particular, smart products are cognizant of their local context and can communicate with one another through the Internet of Things, thereby enabling self-organized control of individual jobs. Despite this potential most applications of smart products and the Internet of Things, including multi agent systems for scheduling and holonic control, continue to focus on hierarchical, centralized data and control structures. In response, this study develops a simple item-centric MFC mechanism and uses simulation to proof the feasibility of self-organized control.

Keywords: *Material Flow Control; Smart Product; Internet of Things; POLCA; Dispatching.*

1. Introduction

This study outlines a new item-centric Material Flow Control (MFC) mechanism for high-variety make-to-order shops and assesses its performance through discrete event simulation. MFC mechanisms decide: (i) *whether* to release a job onto the shop floor; and, (ii) *whether* a job is authorized to be produced at a given station (Graves *et al.* 1995). Both MFC decisions are item-centric, i.e. are directly concerned with the flow of individual items through the shop. However, existing MFC mechanism are typically resource-centric, i.e. focus on the control of capacity resources. Well-known pull approaches to MFC include: Kanban (e.g. Ohno, 1988; Lage Junior & Godinho Filho, 2010); Drum-Buffer-Rope (DBR, e.g. Goldratt & Cox, 1984; Watson *et al*., 2007); Constant Work-in-Process (ConWIP, e.g. Spearman *et al*., 1990; Jaegler *et al*., 2018); Paired-cell Overlapping Loops of Cards with Authorization (POLCA, e.g. Suri, 1998; Riezebos, 2010); WorkLoad Control (WLC, e.g. Land & Gaalman, 1996; Thürer *et al*., 2012); and, Control of Balance by Card-Based Navigation (COBACABANA, e.g. Land, 2009; Thürer *et al*., 2014). For all of these MFC mechanisms MFC decisions are typically executed by a planner (in the case of centralized MFC mechanisms) or worker (in the case of decentralized MFC mechanisms) according to the capacity-related information available locally at the point where a decision is made.

Until recently, technical constraints on real-time information feedback meant this resource-centric approach was a necessity. But this changed with advances in technology that allow for the continual tracing and tracking of individual products (Peng *et al*., 2018), which calls for the design of MFC mechanisms to be revisited (Hsu, 2015; Grundstein *et al*., 2017). In particular, new technologies have created smart products where the information content is permanently bound to material content and where the products are able to influence decisions (McFarlane, 2003). Smart products are cognizant of their local context and can negotiate with local manufacturing resources through the Internet of Things enabling decentralized monitoring and control (Meyer *et al*., 2011). In fact, one of the first smart product applications in manufacturing was a self-organizing manufacturing control system where a job auctions off its operation to stations that bid to undertake the process (Meyer *et al*., 2009). However, existing literature on the use of smart products and the Internet of Things for the control of material flows on the shop floor (Zhang *et al*., 2010; Qu *et al*., 2012; Zhong *et al*., 2013; Qu *et al*., 2016; Lin *et al*., 2019; Guo *et al*., 2020) tends to focus on centralized solutions and traditional MFC mechanisms. But using item-centric data together with MFC mechanisms that use location-based labels for data aggregates, creates significant problems in terms of data structure and data access (Rönkkö, 2006), specifically in high variety contexts.

To the best of our knowledge, no item-centric MFC mechanism has been presented in the literature, i.e. an MFC mechanism that is completely executed at the job/item level using job information only, to enable smart products to autonomously navigate their way through the shop floor. Rather, existing production control solutions using smart products and the Internet of Things tend to have hierarchical multi-agent architectures (see e.g. Cavalieri *et al*., 2000; Lin *et al*., 2019) or focus on the interaction between job and capacity resource (see e.g. Bussmann & Schild, 2000). In response, the objective of this paper is twofold:

- To develop an item-centric MFC mechanism; and,
- To assess its performance against existing MFC mechanisms to proof the feasibility of item-centric control in high-variety make-to-order contexts.

This paper first outlines a new MFC mechanism in Section 2. This mechanism integrates insights from the POLCA literature with insights from the literature on look-ahead dispatching rules (e.g. Bertolini *et al*. 2020). The performance of the new mechanism is then assessed against a traditional POLCA system. POLCA was chosen for the design and as a baseline, since it has been argued to be an alternative to Kanban systems, specifically for companies that produce a high variety of products on a make-to-order basis (e.g. Suri, 1998). Krishnamurthy & Suri (2009) reported that by implementing POLCA in one company the total throughput time was reduced across different products by between 22% and 68% and in another company by an average of 25%. Meanwhile, Riezebos (2010) reported a total throughput time reduction of more than 70%. In all implementations POLCA improved production coordination. The simulation model used to assess performance will be introduced in Section 3, before simulation results will be presented and discussed in Section 4. Finally, Section 5 summarizes major conclusions, limitations and opportunities for future research.

2. Background

The POLCA system and look-ahead dispatching rules provide the building blocks for our new item-centric MFC mechanism. POLCA is introduced in Section 2.1 before look-ahead dispatching is introduced in Section 2.2. Finally, Section 2.3 outlines our new MFC mechanism.

2.1. POLCA

POLCA links the different stations in the routings of jobs using card loops between pairs of stations. Each pair of consecutive stations in the routing of a job has a specific POLCA card that identifies the two stations. The card is consequently job anonymous (Thürer *et al.*, 2019). The card-based element of POLCA is illustrated in Figure 1, following the framework proposed in Liberopoulos & Dallery (2000), for a shop that produces jobs that move from Station 1 to Station 2 and then to Station 3. Since there is no output queue, there are only three elements. Queue A_i^{POLCA} contains the POLCA cards for station i, with i = 1…. k (the number of stations in the system). Queue P_0 is the queue of newly created jobs that are to enter the system and have no POLCA cards attached to them. Finally, queue PA_i contains the jobs completed at the preceding station and to which a POLCA card from the preceding station is still attached. Queue P_0 is equivalent to queue PA_1, which consequently directly reflects the job arrival rate (or demand rate) λ.

When a customer places an order, a new job is created and enters queue PA_1 (P_0). The job waits in this queue until a POLCA 1-2 card is available in queue A_{12}. Once this card is available, the job is processed and moves to the queue PA_2 of the next station with the POLCA 1-2 card still attached. The job waits in queue PA_2 until a POLCA 2-3 card is available in queue A_{23}. Once this card is available, the job is processed. After processing, the POLCA 1-2 card is freed and moves back to queue A_1 while the job moves to the queue of the next station PA_3 with the

POLCA 2-3 card attached. Thus, card loops overlap since the POLCA 1-2 card is only released after the operation at Station 2 has been completed.

[Take in Figure 1]

While POLCA is typically implemented based on physical cards (e.g. Krishnamurthy & Suri 2009; Riezebos, 2010), it can also be implemented as an electronic version (Vandaele *et al.*, 2008). Regardless, how it is implemented it is resource-centric, focusing on the station load to realize MFC. For example, the electronic POLCA mechanism implemented in Vandaele *et al.* (2008) stores the data from the jobs in a centralized database, calculates the aggregates which are stored under machine labels, and then uses these aggregates for control.

2.2. Look-Ahead Dispatching

MFC mechanisms decide *whether* to release a job onto the shop floor and *whether* a job has authorization to be produced at a station. They do not indicate *which* job is released and *which* job is authorized if there is more than one job waiting for release or processing, as typical in high-variety make-to-order contexts. The former MFC decision is therefore supported by a backlog sequencing decision that decides the order in which jobs are considered once order release has been activated (e.g. Thürer *et al.*, 2015) while the latter is supported by a dispatching decision that decides which job should be authorized to be processed at a station (e.g. Blackstone *et al.*, 1982).

There are numerous dispatching rules presented in the literature (e.g. Panwalker & Iskaner, 1977). These rules can be local, i.e. rules that only use information at the current instance and location, or global, i.e. rules that take other time periods and locations into account. Global dispatching rules that focus on the workload at stations downstream in the routing of jobs, so called look-ahead dispatching rules, typically realize better load balancing across stations; however, global dispatching rules are also more complex to implement (Dabbas & Fowler, 2003). A simple and effective global dispatching rule is the Work-in-Next-Queue rule, as used, for example, by Bertolini *et al.* (2020). However, this rule may become dysfunctional if there are no further queues on which to base the dispatching decision, i.e. the operation is the last in the routing of a job and there is no 'next queue' to consider. While this can easily be solved when all jobs have the same final operation, as in Bertoloni *et al.* (2020), care has to be taken when there is routing variety, such as in the high-variety context considered in this study.

Surprisingly, and to best of our knowledge, there is no agent-based dispatching rule presented in the literature that allows different jobs at a location to autonomously decide *which* job to process next. While most agent-based production control systems have a job agent (e.g. Bussmann & Schild, 2000; Cavalieri *et al.*, 2000), job agents do not interact. Rather job agents are restricted to communicate with resource agents and the dispatching decision becomes resource-centric rather than item-centric.

2.3 The Design of a New Item-Centric MFC Mechanism

Both POLCA and Work-in-Next-Queue dispatching take advantage of information from the next station in the routing of a job. POLCA uses this information to authorize production at the current station and Work-in-Next-Queue uses this information to select which job to produce next at the current station. Since both use the same feedback POLCA's MFC decision can be integrated into the Work-in-Next-Queue dispatching rule. In other words, a job is only selected for processing at a station if the workload at the next station in its routing is below a certain limit or norm. Meanwhile the Work-in-Next-Queue dispatching decision is extended by an additional element to decide which job to choose if several jobs share the same destination. The resulting procedure, which is to be execute by the smart products at the job level, can be summarized as follows.

Take Rj to be the ordered set of operations in the routing of job j, and n_j to be the number of operations. Whenever a job arrives at the (physical) queue of station s at which operation i needs to be executed, the following process takes place:

(1) If $i < n_j$, the job communicates with jobs waiting at the next station in its routing performing operation (i+1). Otherwise there is no need for communication. It then enters one of three virtual queues: it enters the virtual queue Q_1 if the queue of the next station contains more than one job, or if $i = n_j$; it enters the virtual queue Q_2 if the queue of the next station is larger than the pre-determined workload limit N_s: else, it enters the virtual queue Q_3.

(2) If the job is an element of Q_1 or Q_3, the job communicates with all other jobs that are an element of Q_1 or Q_3. Jobs in Q_3 have priority over jobs in Q_1. Jobs pertaining to a virtual queue are prioritized according to a chosen dispatching rule.

(3) If the job j has the highest priority among jobs in the (physical) queue of station s it will start production at s once the station becomes idle. Otherwise, when a new job arrives at a station or an operation is completed go back to (1).

While our new item-centric system uses design principles from POLCA, there is a significant difference. In POLCA the workload limit used for production authorization is enforced on pairs of stations in the routing of a job. This leads to imprecise control of the workload in shops with high variety routings since, for example, in a general job shop with six stations, a station forms part of five station pairs. To realize more precise control, the workload limit should be enforced on the workload actually queuing at a station, as in our mechanism. In fact, this is what Bussmann & Schild (2000) did in one of the first applications of smart products at a Daimler Chrysler AG plant in Stuttgart-Untertürkheim (Germany). Bussmann & Schild (2000) used an input buffer and an output buffer at each station. To move from the output buffer of one station to the input buffer of the next station in the routing, a job has to auction off its operation whilst stations have to bid for this operation. MFC is realized by limiting the aggregate of the load in the input and output buffers. If this aggregate is violated, a station is not allowed to bid for an operation.

While the above appears to be similar to our system, again there is a significant difference. In Bussmann & Schild (2000), jobs do not communicate with each other, rather communication occurs between each job and the potential/bidding stations. In other words, a job sends an invitation to bid to all possible stations, collects the bids from the stations and then decides where to move. This ignores the potential for multiple jobs with different priorities to be competing for the same resource. While this assumption was reasonable in the manufacturing line for which the system was developed, it is typically violated in the context of high variety make-to-order shops, as considered in this study. In this context, several jobs typically compete for the same capacity resource.

In contrast, our system is specifically designed for high variety make-to-order contexts and completely job based. It provides production control self-organized by smart products communicating through the Internet of Things. Control is exercised by communication across the different jobs that autonomously navigate their way through the shop floor. Jobs at the current queue communicate with jobs at the next queue to determine which virtual queue they must enter. Then jobs within each virtual queue communicate with each other to determine which has the highest priority. This allows for trading off the different priorities of jobs requiring the same capacity resource. Discrete event simulation will be used next to explore the performance impact of our new item-centric MFC mechanism.

3. Simulation Model

A simulation model of a high-variety make-to-order shop has been implemented using ARENA simulation software. A general flow shop (Oosterman *et al.*, 2000) is considered in order to capture the manufacturing environment that defines high variety shops in practice, Hence, job routings, operation processing times, inter-arrival times to the system and due dates are modelled as stochastic (random) variables. The modelled shop and job characteristics are first detailed in Section 3.1. Section 3.2 then outlines the implemented MFC mechanisms before Section 3.3 defines the dispatching rules applied. Finally, the experimental setting and performance measures are detailed in Section 3.4.

3.1 Shop and Job Characteristics

The general flow shop contains six stations, where each station is a single constant capacity resource. The routing length varies uniformly from one to six operations. All stations have an equal probability of being visited and a station is required at most once in the routing of a job. The resulting routing vector (i.e. the sequence in which stations are visited) is sorted. Operation processing times follow a truncated 2-Erlang distribution with a maximum of 4 time units and a mean of 1 time unit before truncation. Set-up times are considered as part of the operation processing time. Meanwhile, the inter-arrival time of orders follows an exponential distribution with a mean of 0.642, which deliberately results in a utilization level of 90%. Due dates are set exogenously by adding a random allowance factor, uniformly distributed between 30 and 50 time units, to the job entry time. The upper and lower limits of the random allowance factor were set based on the realized throughput times in preliminary simulation experiments. Finally, Table 1 summarizes the main shop and job characteristics modelled in our study.

[Take in Table 1]

3.2 Material Flow Control

Four different MFC mechanisms will be considered: (i) item-centric MFC, measuring the load in terms of the number of jobs; (ii) item-centric MFC, measuring the load in terms of job processing times; (iii) the original POLCA system; and, (iv) POLCA with Starvation Avoidance (SA). Including POLCA SA reflects recent developments in the POLCA literature (Thürer *et al*. 2017) and ensures that arguably the best-performing version of POLCA is included in the experimental design. The development of POLCA SA was motivated by a shortcoming of the original system whereby a station may be starving although there is work in the queue, e.g. when all available POLCA cards that authorize production at that station are at the downstream stations in the loops. This form of premature idleness (Kanet, 1988; Land & Gaalman, 1998) can be resolved by attaching a starvation avoidance card to a job, thereby allowing it to be processed at the starving station (Thürer *et al*., 2017). Using a starvation avoidance card means that the work-in-process cap or limit is exceeded in the loop. To enable the limit to be restored, returning POLCA cards do not become available after being detached from jobs as long as starvation avoidance cards remain in use on the shop floor. Only after all starvation avoidance cards have been returned can standard POLCA cards be used again.

Finally, five levels for the work-in-process limit at a station are considered for the two version of item-centric MFC: 16, 18, 20, 22 and 24. Similarly, for the original POLCA system, five levels for the number of cards per loop are considered: 8, 10, 12, 14, and 16 cards. For POLCA SA, 3, 4, 5, 6 and 7 cards per loop are considered. The same number of cards is used within each loop in each experiment for both POLCA and POLCA SA. This is justified by the balanced shop considered in our study. As a baseline measure, experiments with an infinite number of POLCA cards have also been executed to represent immediate release, i.e. no control over the release of jobs through the shop.

3.3 Dispatching Rule

Two rules are considered for dispatching at stations: the Earliest Release Date (ERD) rule and the Modified Earliest Release Date (MERD) rule. ERD was chosen since it is the card allocation and dispatching rule advocated in POLCA (Suri, 1998; Riezebos, 2010). ERD's for each operation are calculated by backward scheduling from the job due date using planned operation throughput times for each operation in the routing of a job. As in previous POLCA literature (e.g. Riezebos, 2010), a constant allowance for the planned operation throughput time is used. This allowance was defined based on preliminary simulation experiments. The MERD rule is similar to the MODD rule (Baker & Kanet 1983) and combines the ERD with the SPT (Shortest Processing Time) rule. MERD divides the set of waiting jobs into two subsets: a subset of urgent jobs, for which the ERD has already passed, and a subset of non-urgent jobs. Urgent jobs always receive priority over non-urgent jobs, whereby urgent jobs are selected for processing according to the SPT rule and non-urgent jobs are selected according to the ERD rule. MERD was chosen as a more powerful dispatching rule alternative to ERD (Land *et al.*, 2015).

3.4 Experimental Design and Performance Measures

The experimental factors are: (i) the four MFC mechanisms; (ii) the five levels of Work-In-Process limit or number of cards per loop; and, (iii) the two different dispatching rules. A full factorial design was used with 40 (4x5x2) scenarios, where each scenario was replicated 100 times. All results were collected over 13,000 time units following a warm-up period of 3,000 time units to minimize initialization bias. These parameters allow us to obtain stable results while keeping the simulation run time to a reasonable level.

Since we focus on a make-to-order context, our main performance criterion is delivery performance. In this study, delivery performance will be measured by three main performance measures as follows: mean *total throughput time* – the mean of the completion date minus the entry date across jobs; *percentage tardy* – the percentage of jobs completed after the due date; and *mean tardiness* – that is, $T_j = \max(0, L_j)$, with L_j being the lateness of job j (i.e. the completion date minus the due date of job *j*). The percentage tardy provides the most general indication of delivery performance while the total throughput time indicates the mean lateness. Meanwhile, both the mean tardiness and the standard deviation of lateness can be used to measure the dispersion of lateness across jobs. We decided to measure the mean tardiness since the standard deviation of lateness is more sensitive to extreme values than the mean tardiness. Finally, in addition to the three main performance measures, we also measure the average shop floor throughput time. While the total throughput time includes the time that an order waits (in a backlog) before being authorized at the first station in its routing, the shop floor throughput time only measures the time after an order has been authorized. The average shop floor throughput time is a useful indicator of the work-in-process level on the shop floor as it is linked directly to the level of work-in-process.

4. Results

The results from the simulation experiments will first be presented in Section 4.1, before a discussion is provided in Section 4.2.

4.1 Performance Results

Detailed performance results are given in Table 2 together with the 95% confidence intervals to indicate statistical significance. In addition to the results for the four different MFC mechanisms, we also include the results for immediate release.

[Take in Table 2]

The following can be observed from the results:

- *MFC Mechanism*: As expected from the results in Thürer *et al*. (2017), POLCA SA improves performance over POLCA. POLCA SA also outperforms item-centric MFC in terms of mean tardiness when the number of jobs is used to calculate the work in the next queue, being the performance in terms of percentage tardy and total throughput time equivalent. Only if the actual processing time is used to calculate the work in the next queue can item-centric MFC realize a similar level of mean tardiness performance to POLCA SA, yet at a lower percentage tardy and shorter throughput times. In general, item centric MFC using job processing time information to calculate the work in the next queue leads to the best performance across all four MFC mechanisms considered in this study.

- *Dispatching Rule*: As expected, MERD dispatching improves performance over ERD dispatching. This can be observed from the results for immediate release. In fact, MERD dispatching realizes the best tardiness performance when combined with immediate release. However, using an MFC mechanism improves load balancing, as can be observed from the shorter total throughput times, whilst realizing much shorter shop floor throughput times.

The qualitative performance differences across MFC mechanisms are not affected by the dispatching rule, leading MERD dispatching in combination with item centric MFC that uses job processing time information to calculate the work in the next queue to the best performance. This proves the feasibility of the proposed item-centric MFC mechanism and highlights its potential compared to traditional MFC mechanisms, such as POLCA.

4.2 Discussion – Managerial Implications

Our MFC mechanism was designed for execution at the job level; jobs autonomously navigate their way through the shop floor. There is consequently no need for a specific information system infrastructure, centralized data storage systems or protocols for data access and retrieval as in previous literature on agent-based systems (e.g. Lin *et al*., 2019) and electronic POLCA (Vandaele *et al*., 2008) This provides a new way to reap the rewards of investments in smart products and Internet of Things applications, by avoiding the conversion of item-centric data into aggregates labelled according to location, as needed for the application of traditional MFC control. The heuristic is simple to implement and provides firms with performance results that are even better than traditional MFC mechanisms, such as POLCA. However, a main shortcoming is that the capacity resource is not controlled. In Bussmann & Schild (2000), capacity resources would communicate with jobs and indicate when there was a breakdown allowing flows to be diverted to other stations. This option does not exist in our system. Managers should therefore track the item-dwell time, which can be used to indicate problems (Holmström *et al*., 2011), including machine breakdowns.

5. Conclusions

MFC mechanisms decide: (i) *whether* to release a job onto the shop floor; and, (ii) *whether* a job is authorized to be produced at a station. Yet, while the focus of MFC is on the flow of individual jobs, MFC mechanisms typically use capacity resources to execute MFC. While this approach was historically justified by technological limitations on feasible information and feedback structures, advances in technology allow for different, item-centric designs of MFC mechanisms. For example, smart products are aware of their local context and can communicate with other smart products through the Internet of Things. While this enables decentralized monitoring and control, to date no item-centric MFC mechanism has been presented in the literature. In response, this study has outlined a simple item-centric MFC mechanism that combines insight from the POLCA literature with insight from the literature on look-ahead dispatching rules. Discrete event simulation has demonstrated that our new mechanism has the potential to not only match POLCA performance, but to outperform POLCA in terms of throughput times and the percentage of tardy jobs. This has important implications for practice and future research.

A main limitation of this study is our focus on only one MFC mechanism, POLCA. While this is justified by our focus on high-variety make-to-order shops, future research could, for example, explore how Kanban can be translated into an item-centric MFC mechanism for more repetitive contexts. Meanwhile, POLCA and Kanban both focus on product authorization. In contrast, ConWIP or Workload Control focus on order release control. Thürer *et al.* (2020) recently highlighted that combining both MFC decisions, i.e. order release and production authorization, leads to complementary performance effects. Future research could therefore also explore ways to integrate an explicit release function into our item-centric MFC mechanism. Finally, future research could also implement our system in practice. Our study provides a first proof of concept, but further empirical research is needed to assess the full benefits of smart product-driven self-organizing MFC mechanisms.

References

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

Bertolini, M., Braglia, M., Frosolini, M., & Marrazzini, L., 2020, Work In Next Queue CONWIP, *Computers & Industrial Engineering*, *143*, in print

Bussmann, S., & Schild, K., 2000, Self-organizing manufacturing control: An industrial application of agent technology, In *Proceedings Fourth International Conference on Multi Agent Systems* (pp. 87-94). IEEE.

Cavalieri, S., Garetti, M., Macchi, M., & Taisch, M., 2000, An experimental benchmarking of two multi-agent architectures for production scheduling and control, *Computers in Industry*, 43, 2, 139-152.

Goldratt, E.M., & Cox, J., 1984, *The Goal: Excellence in Manufacturing*, North River Press: New York.

Dabbas, R. M., & Fowler, J. W., 2003, A new scheduling approach using combined dispatching criteria in wafer fabs, *IEEE Transactions on Semiconductor Manufacturing, 16*, 3, 501-510.

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

Grundstein, S., Freitag, M., & Scholz-Reiter, B., 2017, A new method for autonomous control of complex job shops–Integrating order release, sequencing and capacity control to meet due dates, *Journal of Manufacturing Systems*, 42, 11-28.

Guo, D., Zhong, R. Y., Ling, S., Rong, Y., & Huang, G. Q., 2020, A roadmap for Assembly 4.0: self-configuration of fixed-position assembly islands under Graduation Intelligent Manufacturing System, *International Journal of Production Research*, (in print)

Holmström, J., Tenhiälä, A., & Kärkkäinen, M., 2011, Item dwell time in project inventories: a field experiment, *Computers in Industry*, *62*, 1, 99-106.

Hsu, H.P., 2015, The development of a RFID and agent-based lot management controller for PROMIS in a client/server structure for IC assembly firm, *Journal of Industrial & Production Engineering*, 32, 8, 538-550.

Jaegler, Y., Jaegler, A., Burlat, P., Lamouri, S., & Trentesaux, D., 2018, The ConWip production control system: a systematic review and classification, *International Journal of Production Research*, 56, 17, 5736-5757.

Kanet, J.J., 1988, Load-limited order release in job shop scheduling systems, *Journal of Operations Management*, 7, 3, 44 – 58.

Krishnamurthy, A. & Suri, R., 2009, Planning and Implementing POLCA: a card-based control system for high variety or custom engineered products, *Production Planning & Control*, 20, 7, 596-610.

Lage Junior, M. & Godinho Filho, M., 2010, Variations of the kanban system: Literature review and classification, *International Journal of Production Economics*, 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., & Gaalman, G.J.C., 1998, The performance of workload control concepts in job shops: Improving the release method, *International Journal of Production Economics*, 56-57, 347-364.

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

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

Liberopoulos, G., & Dallery, Y., 2000, A unified framework for pull control mechanisms in multi-stage manufacturing systems, *Annals of Operations Research*, 93, 1-4, 325-355.

Lin, P., Shen, L., Zhao, Z., & Huang, G.Q., 2019, Graduation manufacturing system: synchronization with IoT-enabled smart tickets, *Journal of Intelligent Manufacturing*, 30, 8, 2885-2900.

McFarlane, D., Sarma, S., Chirn, J. L., Wong, C., & Ashton, K., 2003, Auto ID systems and intelligent manufacturing control, *Engineering Applications of Artificial Intelligence*, *16*, 4, 365-376.

Meyer, G. G., Wortmann, J. C., & Szirbik, N. B., 2011, Production monitoring and control with intelligent products, *International Journal of Production Research*, *49*, 5, 1303-1317.

Meyer, G. G., Främling, K., & Holmström, J., 2009, Intelligent products: A survey, *Computers in Industry*, *60*, 3, 137-148.

Ohno, T., 1988, *Toyota Production System: Beyond Large-Scale Production*, 1st Ed., Productivity Press.

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

Panwalkar, S. S., & Iskander, W., 1977, A survey of scheduling rules. *Operations Research*, *25*, 1, 45-61.

Peng, Z., Guo, S., Wang, L., Guo, J. & Li, X., 2018, Study of order lifecycle tracking systems in building material equipment manufacturing enterprises, *Journal of Industrial & Production Engineering*, 35, 5, 328-337.

Qu, T., Lei, S. P., Wang, Z. Z., Nie, D. X., Chen, X., & Huang, G. Q., 2016, IoT-based real-time production logistics synchronization system under smart cloud manufacturing, *International Journal of Advanced Manufacturing Technology*, 84, 1-4, 147-164.

Qu, T., Yang, H. D., Huang, G. Q., Zhang, Y. F., Luo, H., & Qin, W., 2012, A case of implementing RFID-based real-time shop-floor material management for household electrical appliance manufacturers, *Journal of Intelligent Manufacturing*, *23*, 6, 2343-2356.

Riezebos, J., 2010, Design of POLCA material control systems, *International Journal of Production Research*, 48, 5, 1455-1477.

Rönkkö, M., 2006, A model for item centric material control in manufacturing, *Master's thesis, Helsinki University of Technology, Department of Industrial Engineering and Management*.

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

Suri, R., 1998, *Quick Response Manufacturing: A companywide approach to reducing leadtimes*, Productivity Press.

Thürer, M., Fernandes, N.O., & Stevenson, M., 2020, Material Flow Control in High-Variety Make-to-Order Shops: Combining COBACABANA and POLCA, *Production & Operations Management*, (in print).

Thürer, M., Fernandes, N.O., Ziengs, N., & Stevenson, M., 2019, On the Meaning of ConWIP Cards: An Assessment by Simulation, *Journal of Industrial & Production Engineering*, 36, 1, 49-58.

Thürer, M., Fernandes, N.O., Carmo-Silva, S., & Stevenson, M., 2017, Improving Performance in POLCA Controlled High Variety Shops: An Assessment by Simulation, *Journal of Manufacturing Systems*, 44, 143-153.

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

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

Thürer, M., Stevenson, M., Silva, C., Land, M.J., & 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.

Vandaele, N., Van Nieuwenhuyse, I., Claerhout, D., & Cremmery, R., 2008, Load-based POLCA: an integrated material control system for multiproduct, multimachine job shops, *Manufacturing & Service Operations Management*, 10, 2, 181-197.

Watson, K.J., Blackstone, J.H., & Gardiner, S.C., 2007, The evolution of a management philosophy: The theory of constraints, *Journal of Operations Management*, 25, 387-402.

Zhang, Y., Huang, G. Q., Qu, T., & Ho, O., 2010, Agent-based workflow management for RFID-enabled real-time reconfigurable manufacturing, *International Journal of Computer Integrated Manufacturing*, *23*, 2, 101-112.

Zhong, R. Y., Dai, Q. Y., Qu, T., Hu, G. J., & Huang, G. Q., 2013, RFID-enabled real-time manufacturing execution system for mass-customization production, *Robotics and Computer-Integrated Manufacturing*, *29*, 2, 283-292.

Table 1: Summary of Simulated Shop and Job Characteristics

Shop Characteristics	Routing Variability	Random routing; directed, no re-entrant flows
	No. of Stations	6
	Station Capacities	All equal
	Station Utilization Rate	90%
Job Characteristics	No. of Operations per Job	Discrete Uniform [1, 6]
	Operation Processing Times	Truncated 2–Erlang; (mean ≈ 1; max = 4)
	Due Date Determination Procedure	Due Date = Entry Time + d; d U ~ [30, 50]
	Inter-Arrival Times	Exp. Distribution; mean = 0.642

Table 2: Simulation Results

	Cards/Limit	ERD Dispatching				MERD Dispatching			
		STT[1]	TTT[2]	T[3]	P[4] (%)	STT[1]	TTT[2]	T[3]	P[4] (%)
Immediate Release	None	23.62 ± 0.43	23.62 ± 0.43	1.05 ± 0.14	11.21 ± 0.93	22.35 ± 0.32	22.35 ± 0.32	0.50 ± 0.06	2.63 ± 0.20
POLCA	8	22.06 ± 0.36	24.77 ± 0.57	1.51 ± 0.23	14.22 ± 1.28	21.97 ± 0.44	28.23 ± 1.71	4.25 ± 1.21	14.22 ± 1.74
	10	22.57 ± 0.37	24.07 ± 0.49	1.23 ± 0.18	12.42 ± 1.10	22.23 ± 0.45	25.18 ± 1.16	2.34 ± 0.75	12.57 ± 1.63
	12	22.90 ± 0.38	23.80 ± 0.46	1.12 ± 0.16	11.74 ± 1.02	22.38 ± 0.45	23.80 ± 0.84	1.47 ± 0.51	11.14 ± 1.36
	14	23.12 ± 0.39	23.69 ± 0.44	1.08 ± 0.15	11.45 ± 0.97	22.55 ± 0.44	23.33 ± 0.65	1.21 ± 0.37	10.57 ± 1.16
	16	23.28 ± 0.40	23.65 ± 0.43	1.06 ± 0.15	11.31 ± 0.96	22.60 ± 0.41	22.99 ± 0.46	0.98 ± 0.16	10.14 ± 0.98
POLCA SA	3	17.17 ± 0.28	21.40 ± 0.38	1.04 ± 0.14	10.40 ± 0.84	16.24 ± 0.22	20.04 ± 0.29	0.71 ± 0.08	4.52 ± 0.42
	4	18.34 ± 0.28	21.71 ± 0.38	0.90 ± 0.13	9.63 ± 0.83	17.51 ± 0.23	20.54 ± 0.30	0.62 ± 0.08	4.74 ± 0.46
	5	19.46 ± 0.29	22.16 ± 0.38	0.89 ± 0.13	9.75 ± 0.84	18.62 ± 0.24	21.05 ± 0.32	0.60 ± 0.08	5.41 ± 0.51
	6	20.29 ± 0.30	22.49 ± 0.38	0.91 ± 0.13	9.95 ± 0.85	19.45 ± 0.26	21.42 ± 0.33	0.60 ± 0.08	6.16 ± 0.56
	7	20.93 ± 0.31	22.73 ± 0.39	0.93 ± 0.13	10.14 ± 0.86	20.09 ± 0.27	21.70 ± 0.34	0.63 ± 0.08	6.95 ± 0.63
Item-Centric MFC Number of Jobs	16	12.48 ± 0.17	22.15 ± 0.58	1.58 ± 0.28	11.81 ± 1.09	12.00 ± 0.16	20.35 ± 0.47	1.06 ± 0.23	3.86 ± 0.28
	18	12.66 ± 0.20	21.62 ± 0.56	1.35 ± 0.27	10.77 ± 0.93	12.11 ± 0.17	19.81± 0.33	0.76 ± 0.10	3.51 ± 0.22
	20	12.78 ± 0.21	21.21 ± 0.46	1.16 ± 0.18	10.02 ± 0.93	12.16 ± 0.17	19.59 ± 0.33	0.67 ± 0.07	3.33 ± 0.18
	22	12.86 ± 0.22	20.98 ± 0.41	1.06 ± 0.13	9.51 ± 0.80	12.21 ± 0.18	19.52 ± 0.29	0.64 ± 0.06	3.28 ± 0.17
	24	12.93 ± 0.23	20.87 ± 0.39	1.02 ± 0.12	9.21 ± 0.74	12.24 ± 0.18	19.48 ± 0.28	0.63 ± 0.06	3.25 ± 0.16
Item-Centric MFC Job Processing Times	16	12.73 ± 0.21	20.94 ± 0.38	0.99 ± 0.12	9.27 ± 0.74	11.97 ± 0.16	20.27 ± 0.42	0.98 ± 0.18	3.82 ± 0.25
	18	12.83 ± 0.22	20.85 ± 0.38	0.98 ± 0.12	9.12 ± 0.72	12.07 ± 0.17	19.93 ± 0.36	0.76 ± 0.13	3.61 ± 0.25
	20	12.90 ± 0.22	20.79 ± 0.37	0.96 ± 0.11	9.01 ± 0.70	12.13 ± 0.17	19.71 ± 0.31	0.67 ± 0.08	3.42 ± 0.21
	22	12.96 ± 0.23	20.76 ± 0.37	0.97 ± 0.11	8.99 ± 0.70	12.17 ± 0.17	19.65 ± 0.30	0.66 ± 0.08	3.36 ± 0.20
	24	12.98 ± 0.23	20.72 ± 0.36	0.97 ± 0.11	8.92 ± 0.69	12.20 ± 0.17	19.58 ± 0.28	0.63 ± 0.06	3.30 ± 0.18

95% confidence intervals on: [1] shop throughput time; [2] total throughput time; [3] tardiness; and [4] percent tardy

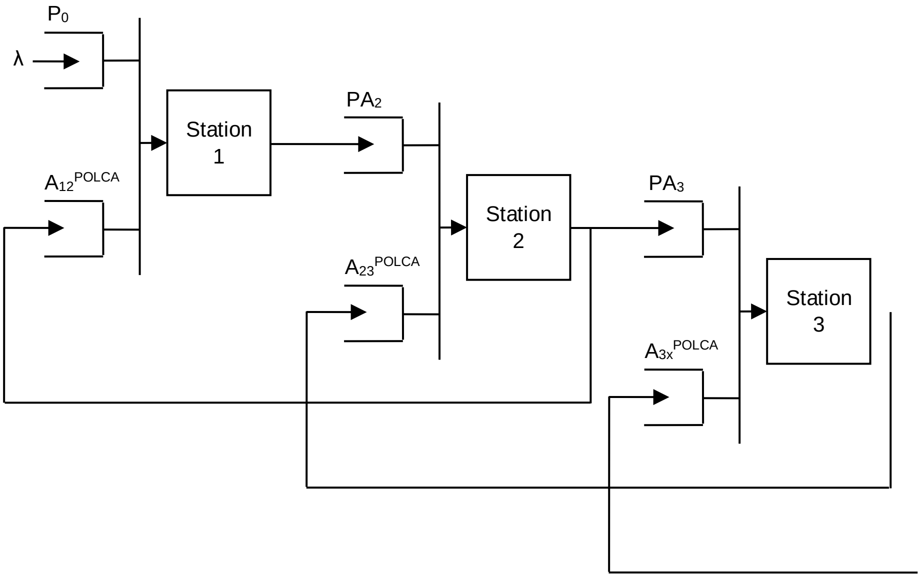

Figure 1: POLCA's Card-Based Control Loops